Artificial Intelligence 277 (2019) 103176

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Representation learning with extreme learning machines and empirical mode decomposition for wind speed forecasting methods

Hao-Fan Yang, Yi-Ping Phoebe Chen [*]

Department of Computer Science and Information Technology, La Trobe University, Melbourne Bundoora, Victoria, Australia

ARTICLE INFO

Article history:
Received 6 November 2017
Received in revised form 14 September 2019
Accepted 20 September 2019
Available online 25 September 2019

Keywords:
Representation learning
Deep learning
Extreme learning machines
Empirical mode decomposition

ABSTRACT

Time series analysis has become more accurate with the emergence of powerful modelling methods based on machine learning development. Prediction models use historical time series to predict future conditions that occur over periods of time. However, most of these models are shallow models, only containing a small number of non-linear operations and without the ability or the capacity to extract underlying features from complex time series accurately. Moreover, deep learning approaches outperform statistical and computational approaches if a large amount of data and/or hidden layers are involved in the development of a forecasting model, but they are criticized for their relatively slow learning speeds. Therefore, this research proposes a hybrid model, which is hybridized by empirical mode decomposition, stacked auto-encoders, and extreme learning machines, aiming to forecast wind speed accurately and efficiently. The evaluation is undertaken by conducting extensive experiments using real-world data. The results show that the proposed E-S-ELM can accurately and efficiently forecast wind speed, and the effectiveness of the shared-hidden-layer approach for deep networks is also demonstrated.

1. Introduction

Maintaining the balance between power generation and consumption at any moment in the power grid can reduce energy fluctuations and improve power stability. However, it is hard to dispatch renewable energy sources when they penetrate the power grid because they are all intermittent energy sources. Therefore, fluctuations in the power generated by renewable energy sources has received a huge amount of attention and developing an accurate forecasting model for renewable energy sources is essential and beneficial for power grid integration and management [1–3]. Wind generation is one of the common renewable energy sources in the power grid, and it has been used all over the world since the 19th century [4]. Nevertheless, it is a significant challenge to integrate wind generation into the power grid since the wind speed time series has nonlinear, nonstationary, and intermittent characteristics. Implementing large-scale wind generation requires answering a lot of problems such as the design of the competitive electricity market, responses of real-time grid operations, the quality of power generation, and the stability and reliability of the power system, etc. [5]. Accurately forecasting wind speed is considered to be an effective tool to overcome the aforementioned problems. For instance, the notion of accurate wind forecasting is enticing in competitive electricity markets since it can provide appropriate market price incentives

[*] Corresponding author.
E-mail addresses: yangper1981@hotmail.com (H.-F. Yang), phoebe.chen@latrobe.edu.au (Y.-P.P. Chen).

https://doi.org/10.1016/j.artint.2019.103176
0004-3702/© 2019 Elsevier B.V. All rights reserved.

for energy imbalances and also can help to develop a well-functioning market [6]. Improving wind speed forecasting has significant economic and technical advantages, however, the biggest challenge of wind speed forecasting is the intermittency of wind.

In the literature, several time series methods have been developed for wind speed forecasting, and three main categories of forecasting methods are frequently adopted, these being statistical, computational intelligence, and hybrid approaches [7]. Statistical methods used in wind speed forecasting assume and constrain the developed models to be linear [8]. However, the collected wind speed data are always nonlinear which limits the capability of statistical methods to forecast wind speed. Computational intelligence methods, such as support vector machines (SVMs) [9] and neural networks (NNs) [10] have been widely used in wind speed forecasting because of their flexible nonlinear modelling capability. Nonetheless, most of these methods are time consuming, suffer from over-fitting, and require the expertise of domain knowledge. To overcome these difficulties, hybrid approaches, based on a combination of statistical and computational intelligence methods, have attracted increasing attention in recent years [11]. Although the literature has shown that hybrid approaches are able to generate more accurate forecasting results than statistical and computational intelligence methods, most of them may not be able to handle a large amount of data and the computed results are not as accurate as expected. Moreover, the dawning of the big data era brings opportunities to greatly promote the improvement of prediction accuracy [12], and a forecasting model with deep structures has proven to be a noteworthy advancement over previous methods in predictive performance [13,14]. This motivates us to develop a hybrid deep learning model for wind speed forecasting.

This paper proposes a wind speed forecasting model, namely E-S-ELM, which is hybridized by empirical mode decomposition (EMD), stacked auto-encoders (SAE), and an extreme learning machine (ELM), aiming to forecast wind speed accurately and efficiently and improve the performance of deep learning used in time series forecasting. EMD is adopted in the proposed model because it can decompose complex wind speed time series in the time domain into a collection of simpler ones, and it also allows varying frequencies in time to be preserved. With these advantages, the performance of the forecasting model will be potentially improved [15]. Nonetheless, the end effect (the decomposed time series will be gradually distorted by directly interpolating the cubic spline between the endpoints and local extreme points when the endpoints of the time series are not the extreme points) is an important problem related to EMD, and it may misrepresent the decomposed time series which can cause significant error. The literature [16,17] shows that the prediction error caused by the end effect can be solved by deep learning approaches. Also, deep learning approaches in prediction have shown impressive results in several areas [16,18]. Therefore, we applied one of the most popular deep learning approaches, SAE, to develop a deep architecture of the forecasting model. Without domain knowledge and human ingenuity to define the features hidden in the data, the SAE can represent the time series' salient structure by unsupervised feature learning. Deep learning approaches outperform statistical and computational approaches if a large amount of data and/or hidden layers are involved in the development of the forecasting model, but they are criticized for their relatively slow learning speeds. In order to overcome this disadvantage, the ELM learning algorithm is applied to replace the commonly used back-propagation learning algorithm to provide a fast learning speed and decent generalization capability [19]. The wind speed time series collected from five different airports in the United Kingdom is used to evaluate the developed E-S-ELM model. Moreover, a dataset combining the data collected from the aforementioned location is adopted to prove the effectiveness of the shared-hidden-layer approach which has been applied in [20].

The content of this paper is organized as follows: Section 2 reviews the current deep learning models for time series forecasting. Section 3 explains the methodologies used in this research. Section 4 evaluates the performance of the E-S-ELM wind speed forecasting model, and the conclusion is given in Section 5.

2. Current deep learning models for time series forecasting

The basic concept of deep learning can be defined as using deep architectures, such as multiple layers of nonlinear processing units, to extract and transform the inherent features in the data from the lowest level to the highest level, and every continuous layer uses the output from the previous layer as input. A variety of deep learning approaches has been developed to model complex time series. In this section, we review some current deep learning models for time series forecasting.

A hybrid deep learning model was introduced in [16], which is hybridized by SAE and the Levenberg-Marquardt (LM) algorithm. The proposed model, namely SAE-LM, was developed using the Taguchi method to replace the traditional trial-and-error method, and it was applied to traffic data collected in the United Kingdom. To evaluate the performance of the SAE-LM, three hybrid models, namely exponential smoothing and LM, the particle swarm optimization algorithm with NNs, and radial basis function NNs, were used as benchmark methods. The experiment results show that the SAE-LM has superior performance in traffic flow forecasting and it is the only model which can handle lumpy data compared with the benchmark methods. Since the amount of collected data may not always be sufficient, the authors in [20] proposed a shared-hidden-layer deep learning model where the hidden layers are shared across the data collected from different locations but the output layers are matched to the corresponding input layers. One of the most popular deep learning approaches, the SAE, was adopted in the proposed model to develop a deep network for unsupervised pre-training and supervised fine-tuning. The model was evaluated using data collected from four wind farms located in northern China, and each has different characteristics of wind time series. The forecasting results indicated that the SAE performed worse than the shallow models when training data is insufficient. However, the proposed shared-hidden-layer deep learning model can solve this disadvantage, and it outperforms the SAE, SVM, and ELM in wind speed forecasting, no matter whether the data is sufficient or insufficient.

An ensemble deep learning method applied in regression and time series forecasting was first proposed in [18]. The authors proposed an ensemble deep learning method which consists of a deep belief network (DBN) and support vector regression (SVR). DBN was applied to train the data using a different number of epochs, and the outputs from DBN were trained as inputs for an SVR model for forecasting. SVR, feedforward NNs, DBN, and ensemble feedforward NNs were used as benchmark methods and root mean square error (RMSE) and mean absolute percentage error (MAPE) were used to evaluate the performance of the proposed ensemble DBN-SVR model. The results show that the proposed model outperformed the benchmark methods for all targeted datasets. Another DBN forecasting model was proposed in [21], which is combined with multiple layers of a restricted Boltzmann machine (RBM) in order to represent the high-level features of high dimension data and further improve the accuracy of prediction. A 3-layer DBN of RBMs was used to capture the feature of input space of time series data, the data were pre-trained by the descent of probabilities of energy functions and then the connection weights between the layers of RBMs were fine-tuned using a back-propagation learning algorithm. Particle swarm optimization (PSO) was adopted to find the optimal numbers of input and hidden neurons and the learning rate of RBM. The authors also used the seasonal exponential smoothing approach to pre-process the original time series data (remove the seasonal trends in the original time series data) which has proven to be more appropriate for NN-based models. The CATS benchmark [22] was used in two experiments, long-term and one-ahead prediction, to verify the effectiveness of the proposed model. The experiment results showed that the proposed DBN-RBM model achieved superior performance compared to conventional NN-based models and mathematical linear models, especially in short-term prediction.

To sum up, a huge number of deep learning approaches have been developed to solve regression problems in time series forecasting, however, most of them only perform well if sufficient data is supplied for training. Also, the extremely slow learning speeds of deep learning are still an unavoidable challenge. These disadvantages motivate us to propose the wind speed forecasting model, E-S-ELM, aiming to improve the performance of deep learning used in time series forecasting.

3. Methodologies

The methodologies of the proposed E-S-ELM wind speed forecasting model are explained in this section.

3.1. Empirical mode decomposition (EMD)

In the proposed model, the empirical mode decomposition (EMD) method is used to decompose the collected wind time series into several intrinsic mode functions (IMFs) and residues. An IMF is defined as a function which has only one extrema between zero-crossings and a mean value of zero [23]. With this definition of IMF, the EMD can decompose the time series $y(t)$ using the following steps:

Step 1. Identify all local extremes of the $y(t)$ and create lower envelope $l(t)$ and upper envelope $u(t)$ by interpolating the cubic spline with local minima and maxima, respectively.
Step 2. Calculate the mean of lower and upper envelopes $m(t)$, and the first component $d(t)$ can be computed by subtracting $m(t)$ from $x(t)$. That is, $d(t) = y(t) - m(t)$.
Step 3. Check whether $d(t)$ and $m(t)$ satisfy the requirement of an IMF. If so, the first IMF component $x_1(t)$ is set as equal to $d(t)$ and the first residue $r_1(t) = y(t) - d(t)$. Otherwise, replace $y(t)$ with $d(t)$ and repeat Step 1 to Step 3 until $x_1(t)$ and $r_1(t)$ can be obtained.
Step 4. Take the $r_1(t)$ as the new time series and repeat Step 1 to Step 4 to get the second IMF component $x_2(t)$ and the second residue $r_2(t)$. Repeat the aforementioned steps n times until $r_n(t)$ becomes either a monotonic function or over-distorted which cannot be decomposed into an IMF.

Finally, the original time series can be decomposed as:

$$y(t) = \sum_{i=1}^{n} x_i(t) + r_n(t) \tag{1}$$

where $x_1(t)$ and $x_n(t)$ indicate the IMF components with the highest and the lowest frequency band, respectively. In this research, the collected wind flow time series can then be decomposed into several IMFs and residues which are easier and more efficient than the original time series for the forecasting model to learn and predict. An example of the decomposition process using EMD is shown in Fig. 1.

3.2. Auto-encoders

Analyzing time series is a significant challenge, especially when the time series is high-dimensional and complex with unique characteristics. Many techniques, such as dimensionality reduction techniques [24] and wavelet transform methods, are applied to remove the noise contained in the time series and reduce dimensionality [25]. Nonetheless, it should be

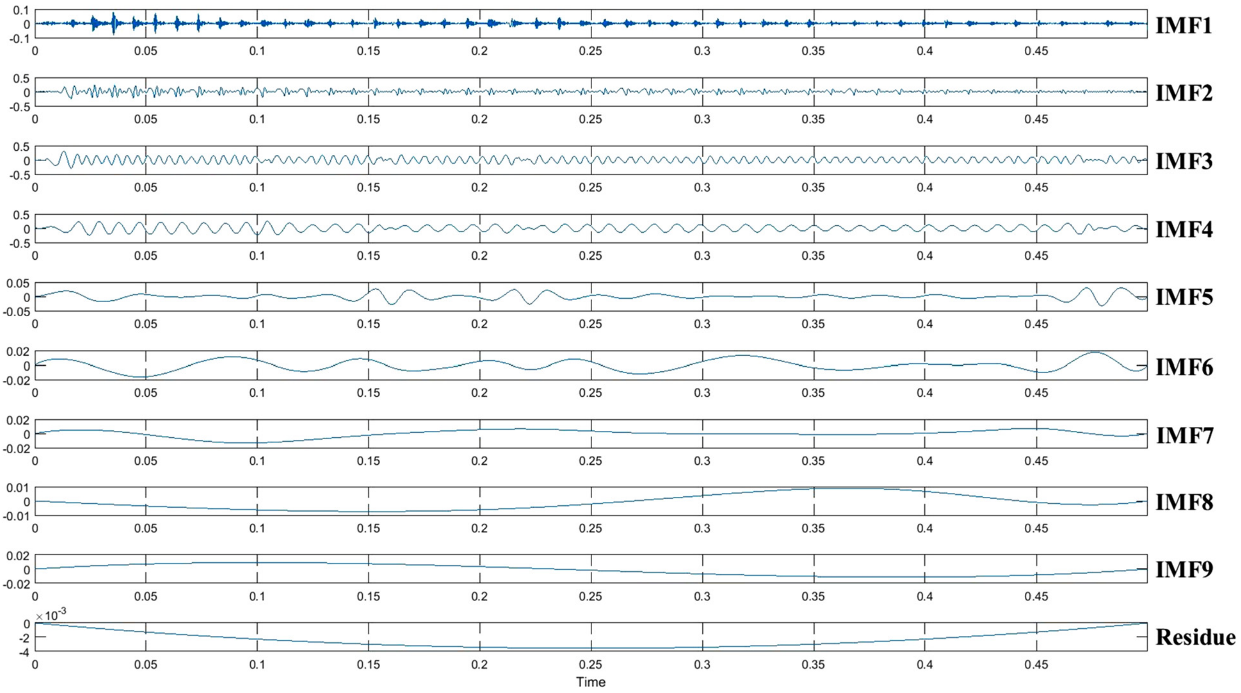

Fig. 1. Example of the process of EMD.

noted that valuable information could be lost and the selection of the preserved features and adopted techniques always requires expertise with the targeted data [26]. Unsupervised feature learning has been successfully applied and combined with deep networks to create more accurate prediction models [27]. Therefore, in this research, we represent the decomposed wind flow data using one of the most common unsupervised feature learning approaches, i.e. an auto-encoder. In general, an auto-encoder first encodes a vector input x to a hidden layer h, and then decodes it to a reconstruction z. For example, the decomposed time series data $\{x_1(t), x_2(t), \ldots, x_n(t)\}$, where $x_i(t) \in RD$, an input $x_i(t)$ is first mapped to a latent representation $h^1(x_i(t))$ and then it is mapped back to a reconstruction $z^1(x_i(t))$ according to equation (2) and equation (3), respectively.

$$h(x) = f(w_1 * x + b_1) \tag{2}$$

$$z(x) = g\big(w_2 * h(x) + b_2\big) \tag{3}$$

where $f(.)$ and $g(.)$ are activation functions, w_1 is the encoding matrix, w_2 is the decoding matrix, b_1 is the encoding bias vector, and b_2 is the decoding bias vector. In this research, as the model is developed to forecast the wind speed (regression problem), we follow the squared loss function $L(x, z)$ proposed in [28] to measure the reconstruction error. By minimizing the average difference between input x and output z, the model parameters, denoted as λ, can be obtained as in equation (4).

$$\lambda = \operatorname{avg}\min_{\lambda} \frac{1}{n} \sum_{i=1}^{n} \big\| x_i(t) - z\big(x_i(t)\big) \big\|^2 \tag{4}$$

The promise of unsupervised feature learning is that potentially better features than hand-labelled feature representation can be learned. With limited or even no expertise with the data, we can take the learned feature representation and whatever labelled data we have to solve the prediction tasks by applying supervised learning algorithms. Fig. 2 shows the schematic of the AE model that is used for unsupervised feature learning in this research.

3.3. Deep learning and extreme learning machine

Deep learning is a type of machine learning method based on the learning representations of data, and through multiple layers of nonlinear processing units, the inherent features in the data can be extracted and transformed from the bottom layer to the top layer, and every continuous layer uses the output from a previous layer as input. This nonlinearity enables a better prediction model that can learn more representations when multiple layers are stacked one by one from a deep network to further disentangle the factors of variations in the time series data. The literature has proven that extending

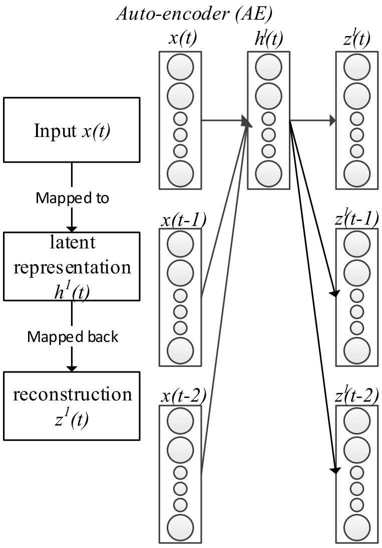

Fig. 2. Schematic of the AE model with one hidden layer.

the modelling structures and applying the concept of deep learning for prediction models can greatly improve prediction accuracy [13]. However, although the deep network has a more compact representation, enabling it to perform better than shallow models, the difficulty of training multiple hidden layers using back-propagation algorithms may result in vanishing gradients and an exceedingly long training time [29]. This can be solved by a greedy layer-wise unsupervised learning algorithm to pre-train each hidden layer and a faster supervised learning algorithm at the last layer of the network to obtain the prediction result.

Stacked auto-encoders (SAE) can represent data features from the obtained data without prior knowledge, which may greatly improve the performance of the prediction models since poor local minima can be avoided and a better initialization of the modelling parameters can be acquired. Moreover, with the ability to randomly assign and never update the input weights, a powerful feedforward neural network, an extreme learning machine (ELM), has been proven to generate highly accurate prediction results and perform thousands of times faster than back-propagation algorithms to train deep networks [30]. Therefore, we adopt the SAE as an unsupervised learning approach, and combine this with ELM as a supervised learning algorithm to structure a deep network for time series prediction. The proposed deep network is divided into two parts (unsupervised and supervised), which is illustrated in Fig. 3.

As can be seen in Fig. 3, each auto-encoder is used as a structured hidden layer and several hidden layers can be stacked one by one from the bottom layer to create a deep network, i.e. SAE. In this deep network, each hidden layer (excluding the last hidden layer) is trained to minimize the reconstruction error of its inputs, and the outputs of a hidden layer are used as inputs of its next layer. In this research, the amount of inputs and outputs of a hidden layer is set as equal. By randomly generating the input weight of the first hidden layer, its output weight β^1 can then be calculated. All the following input weights of the hidden layers will be chosen to be orthogonal which are $(\beta^1)^T, \ldots, (\beta^{i+1})^T, \ldots,$ and $(\beta^{j-1})^T$. The outputs of the last hidden layer in unsupervised learning are trained by ELM in the supervised learning layer to complete the proposed deep network.

ELM is a training approach for single hidden layer feedforward neural networks in which the input weights and biases of the hidden layer are randomly generated and the output weights are systematically computed [31]. It has been successfully applied in many areas and its extension for multi-layers has been proven to have better performance than many deep learning approaches [30]. Suppose that ELM with $\tilde{A}$ hidden nodes is trained to learn A arbitrary distinct samples (x_j, c_j), where $x_{ji} \in R$, $c_{ji} \in R$, $x_j = [x_{j1}, x_{j2}, \ldots, x_{jp}]^T$ and $c_j = [c_{j1}, c_{j2}, \ldots, c_{jq}]^T$. The model developed by ELM can be mathematically formulated as:

$$I(x_k; w, b, \mu) = \sum_{j=1}^{\tilde{A}} \mu_j v(w_j * x_k + b_j), \quad k = 1, 2, \ldots, A \tag{5}$$

where $w_j = [w_{j1}, w_{j2}, \ldots, w_{jp}]^T$ is the weight vector connecting the input nodes and the jth hidden node; $\mu_j = [\mu_{j1}, \mu_{j2}, \ldots, \mu_{jq}]^T$ is the weight vector connecting the output nodes and the jth hidden node; b_j is the bias of the jth hidden node; and $v(.)$ is the activation function. The developed model can approximate these A samples with zero error [31], meaning that

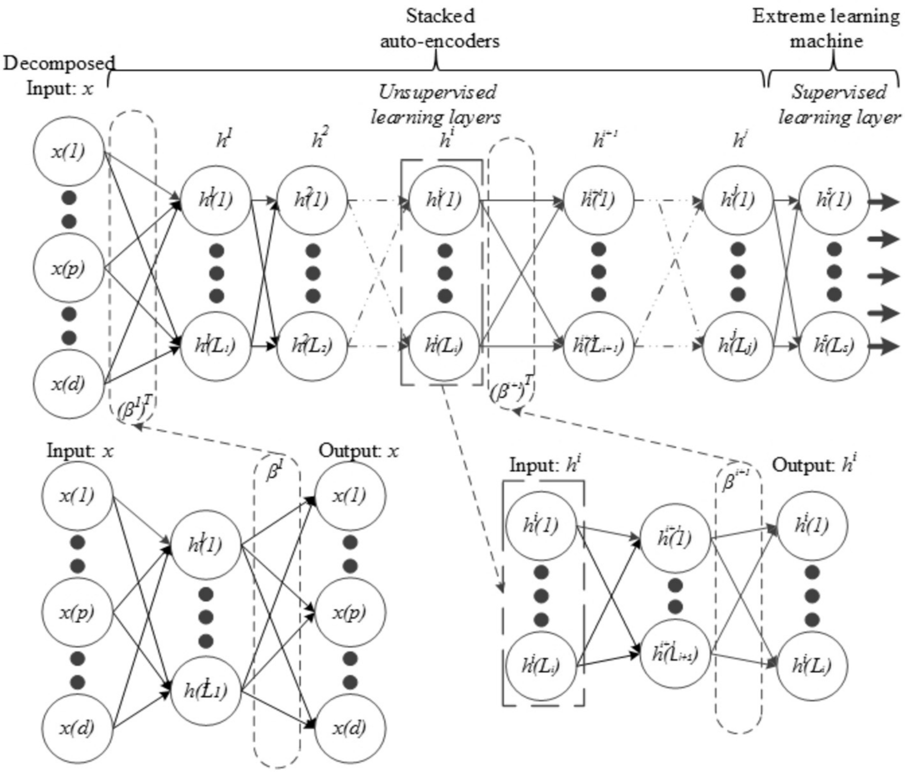

Fig. 3. Proposed deep network for wind speed forecasting. The decomposed input is the time series data decomposed by EMD. The decomposed input is encoded to the hidden layer h^1, and then decoded back to a reconstruction. Each hidden layer in unsupervised learning is trained to minimize the reconstruction error of its input, and the outputs of a hidden layer are used as inputs of its next layer. The above approach is repeated until the desired number of hidden layers (h^j) is reached. The outputs of the last hidden layer in unsupervised learning are trained by ELM in the supervised learning layer to complete the proposed deep network.

$$\sum_{j=1}^{\tilde{A}} \mu_j v(w_j * x_k + b_j) = c_k, \quad k = 1, 2, \ldots, A \tag{6}$$

The above A equations can be rewritten as

$$\mathrm{H}\mu = \mathrm{T} \tag{7}$$

where H is the hidden layer output matrix of the developed ELM model which can be formulated as

$$\mathrm{H} = \begin{bmatrix} v(w_1 * x_1 + b_1) & \ldots & v(w_{\tilde{A}} * x_1 + b_{\tilde{A}}) \\ \vdots & \ldots & \vdots \\ v(w_1 * x_A + b_1) & \ldots & v(w_{\tilde{A}} * x_A + b_{\tilde{A}}) \end{bmatrix}_{A \times \tilde{A}} \tag{8}$$

$\mu = [\mu_1, \mu_2, \ldots, \mu_{\tilde{A}}]^T$ and $T = [c_1, c_2, \ldots, c_q]^T$ indicate the matrix of output weights and the matrix of targets.

The input weights w_j and the biases of the hidden layer b_j in ELM are randomly generated and not necessarily tuned. Also, the output matrix of the hidden layer H can remain unchanged once w_j and b_j are randomly assigned. Therefore, the output weights can be systematically determined by computing the least-squares solution of equation (7). The smallest norm least-squares solution is unique and has the smallest norm of all the solutions, which can be formulated as

$$\hat{\mu} = \mathrm{H}^{\dagger}\mathrm{T} \tag{9}$$

where $\mathrm{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the matrix H. In this research, we use the singular value decomposition method to obtain $\mathrm{H}^{\dagger}$.

4. Evaluation

Extensive experiments are conducted to 1) estimate the effectiveness of EMD, SAE, and ELM compared to the proposed E-S-ELM; and 2) compare them with current deep learning models to evaluate the accuracy and efficiency of the proposed

E-S-ELM model. For each of the deep learning models, we follow the recommendation from [16,32,33] to set the number of hidden layers as 50 and the number of hidden nodes as $log_2(N_i)$, where N_i refers to the number of input nodes. All the models are evaluated with five wind speed datasets obtained from Southampton airport (D-1), London city airport (D-2), Lydd airport (D-3), Cranfield airport (D-4), and Sheffield airport (D-5) in the United Kingdom. Moreover, since the shared-hidden-layer approach in [20] can perform well in wind speed forecasting, a combined dataset with D-1 to D-5 is also used to evaluate the effectiveness of the shared-hidden-layer approach. The dataset with "C" at the end of its name (i.e. D-1C, D-2C, D-3C, D-4C, and D-5C) denotes that the forecasting result is obtained from the combined dataset. The wind speed data of each dataset was collected from 1st January 2013 to 31st December 2014 at 15-minute intervals, and we used two weeks' data to forecast the wind speed for the next 12 and 24 hours. Each dataset is divided into two parts: 75% (the first 274 days in 2013 and in 2014) for training and 25% (the last 91 days in 2013 and in 2014) for testing. The wind speed data has two dimensions, which are the wind direction marked by a 16-wind compass rose and the wind speed measured in meters per second. According to the physical significance of the wind direction and speed, we transform the collected wind speed data as:

$$v^0 = v * \cos\theta \tag{10}$$

where v^0, v, and θ indicate the wind speed in a 0-degree angle direction, the wind speed, and the degree angle of wind direction, respectively. Therefore, the wind speed time series used in this research is simulated as the speed component of the air motion in a 0-degree angle direction.

The experiments using each model for every dataset were conducted 50 times, and the results were averaged for evaluation. All the evaluation was performed on a Windows 7 PC with 3.40 GHz Intel i7 CPU, 16.0 GM RAM, and 64-bit operating system. To evaluate the effectiveness of EMD, SAE, and ELM compared to the proposed E-S-ELM, we adopted three error measures, mean absolute percentage error (MAPE), variance absolute percentage error (VAPE), and coefficient of determination (R^2). MAPE, VAPE, and R^2 denote the difference in the mean, the difference in the variance, and the related degree between the observed data and predicted value, which can be formulated as in equations (11), (12), and (13), respectively.

$$MAPE = \frac{1}{A}\sum_{j=1}^{A}\left|\frac{Y_j - F_j}{Y_j}\right| \tag{11}$$

$$VAPE = Var\left(\left|\frac{Y_j - F_j}{Y_j}\right|\right) \tag{12}$$

$$R^2 = 1 - \frac{\sum_j (Y_j - F_j)^2}{\sum_j (Y_j - F_j)^2 + \sum_j (\tilde{Y}_j - F_j)^2} \tag{13}$$

where Y_j is the observed data, $\tilde{Y}_j$ is the mean of the observed data, and F_j is the predicted value. A lower value of MAPE and VAPE indicates the higher accuracy of the prediction, and a higher value of R^2 denotes better forecasting performance.

4.1. Evaluation of the effectiveness of EMD, SAE, and ELM compared to the proposed E-S-ELM

The data collected from 1st January 2013 to 31st December 2013 were used to examine the effectiveness of EMD, SAE, and ELM. The evaluation results of the effectiveness of EMD, SAE, and ELM compared to the proposed E-S-ELM are shown in Table 1. From Table 1, we can observe that all the deep structure models, i.e. E-S-ELM, EMD-SAE, and SAE-ELM, can simulate the real data relatively well by providing a large amount of training data. The predicted results demonstrate that the features of the big data learned and represented by the SAE approach can achieve higher accuracy. The effectiveness of the EMD approach is revealed by comparing the results obtained from E-S-ELM and SAE-ELM, which shows a decent improvement in forecasting performance. Moreover, compared with the results of E-S-ELM and EMD-SAE, the effectiveness of ELM does not have a significant effect on the accuracy improvement of wind speed forecasting. However, as can be seen in Fig. 4, E-S-ELM outperforms EMD-SAE on training time saving, which demonstrates the effectiveness of ELM in wind speed forecasting.

The proposed E-S-ELM performs the best on almost all the single and combined datasets in a 12-hour forecasting horizon. However, it should be mentioned that in a 24-hour forecasting horizon, the EMD-SAE performs better than E-S-ELM on the combined datasets. Moreover, except for EMD-ELM, all the models perform better on the combined datasets (D-1C to D-5C) compared to the single datasets (D-1 to D-5) for both the 12-hour and 24-hour forecasting horizon using the shared-hidden-layer structure, which shares all the input and the hidden layers to train the data in combined datasets. This result shows that such a parallel training structure can transfer universal features to each output layer and perform better than a sequential training structure if the amount of training data is sufficient. To further evaluate the performance of the proposed E-S-ELM and the effectiveness of the shared-hidden-layer structure, we use three deep learning models for comparison in the following section.

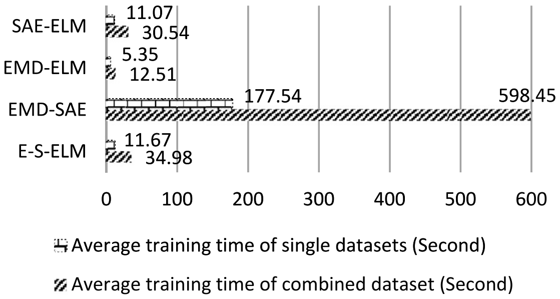

Fig. 4. Comparison of average training time in single and combined datasets.

Table 1
Performance evaluation of E-S-ELM, EMD-SAE, EMD-ELM, and SAE-ELM forecasting.

Error standard	Dataset	12 hours ahead				24 hours ahead			
		E-S-ELM	EMD-SAE	EMD-ELM	SAE-ELM	E-S-ELM	EMD-SAE	EMD-ELM	SAE-ELM
MAPE	D-1	0.0899	**0.0892**	0.1897	0.1056	**0.0961**	0.0963	0.2082	0.1124
	D-2	**0.0913**	0.0965	0.1892	0.1027	**0.0966**	0.1000	0.2080	0.1107
	D-3	**0.0914**	0.0925	0.1909	0.0954	**0.0998**	0.1002	0.2082	0.1078
	D-4	**0.0889**	0.0910	0.1912	0.1073	**0.0937**	0.0968	0.1951	0.1120
	D-5	**0.0906**	0.0924	0.1922	0.1066	**0.0927**	0.1020	0.2014	0.1127
	D-1C	**0.0785**	0.0839	0.1858	0.0949	0.0915	**0.0868**	0.2033	0.1066
	D-2C	**0.0824**	0.0868	0.1867	0.0937	0.0939	**0.0880**	0.2077	0.1061
	D-3C	0.0790	**0.0785**	0.1885	0.0922	**0.0900**	0.0907	0.2069	0.1005
	D-4C	**0.0759**	0.0768	0.1910	0.0917	0.0922	**0.0863**	0.1997	0.1027
	D-5C	0.0849	**0.0822**	0.1901	0.0905	0.0907	**0.0905**	0.2050	0.1041
Error standard	Dataset	12 hours ahead				24 hours ahead			
		E-S-ELM	EMD-SAE	EMD-ELM	SAE-ELM	E-S-ELM	EMD-SAE	EMD-ELM	SAE-ELM
VAPE	D-1	0.0020	**0.0019**	0.0164	0.0023	0.0024	**0.0024**	0.0186	0.0026
	D-2	**0.0018**	0.0021	0.0172	0.0023	**0.0023**	0.0026	0.0190	0.0026
	D-3	**0.0020**	0.0020	0.0165	0.0023	**0.0025**	0.0025	0.0186	0.0025
	D-4	**0.0019**	0.0020	0.0168	0.0023	**0.0024**	0.0025	0.0182	0.0025
	D-5	**0.0020**	0.0020	0.0170	0.0024	**0.0024**	0.0025	0.0186	0.0026
	D-1C	**0.0017**	0.0018	0.0170	0.0021	0.0022	**0.0022**	0.0187	0.0024
	D-2C	**0.0016**	0.0019	0.0166	0.0020	0.0024	**0.0021**	0.0187	0.0023
	D-3C	**0.0017**	0.0017	0.0164	0.0020	0.0022	**0.0022**	0.0186	0.0022
	D-4C	**0.0016**	0.0017	0.0166	0.0019	0.0022	**0.0021**	0.0183	0.0022
	D-5C	**0.0017**	0.0017	0.0173	0.0020	**0.0021**	0.0022	0.0189	0.0022
Error standard	Dataset	12 hours ahead				24 hours ahead			
		E-S-ELM	EMD-SAE	EMD-ELM	SAE-ELM	E-S-ELM	EMD-SAE	EMD-ELM	SAE-ELM
R^2	D-1	0.9863	**0.9872**	0.9463	0.9733	0.9778	**0.9788**	0.9388	0.9703
	D-2	**0.9867**	0.9851	0.9466	0.9751	**0.9776**	0.9771	0.9389	0.9713
	D-3	**0.9859**	0.9848	0.9454	0.9782	0.9762	**0.9769**	0.9387	0.9729
	D-4	**0.9860**	0.9845	0.9454	0.9724	**0.9789**	0.9784	0.9455	0.9705
	D-5	**0.9864**	0.9841	0.9408	0.9731	**0.9793**	0.9761	0.9418	0.9702
	D-1C	**0.9885**	0.9875	0.9501	0.9785	0.9801	**0.9826**	0.9418	0.9735
	D-2C	**0.9884**	0.9870	0.9479	0.9790	0.9790	**0.9821**	0.9392	0.9738
	D-3C	0.9886	**0.9889**	0.9464	0.9788	0.9807	**0.9812**	0.9395	0.9763
	D-4C	**0.9892**	0.9891	0.9455	0.9801	0.9799	**0.9829**	0.9431	0.9754
	D-5C	0.9871	**0.9873**	0.9411	0.9802	0.9802	**0.9812**	0.9400	0.9748

4.2. Evaluation of E-S-ELM compared to current deep learning models

In this section, we used the data collected from 1st January 2014 to 31st December 2014 to evaluate the performance of E-S-ELM over stacked auto-encoders Levenberg-Marquardt (SAE-LM) [16], deep belief networks with support vector regression (DBN-SVR) [18], and deep belief networks with a restricted Boltzmann machine (DBN-RBM) [21] in wind speed forecasting. All the selected deep learning models are back-propagation-based and fully connected multilayer perceptron training schemes, and the initial learning rate of each model is set as 0.1 with a 95% decay rate for each learning epoch. To reduce the chance of model overfitting, the sparsity constraint is commonly used in deep learning approaches. However, in order to avoid the pre-training step which is used to determine the sparsity parameters, we embedded the dropout function [34] in the training process of each deep learning model to save computational time.

Table 2
Comparison of testing accuracy and training time in wind speed forecasting.

Forecasting horizon		12 hours ahead			
Model		E-S-ELM	SAE-LM	DBN-SVR	DBN-RBM
Average accuracy (%)	Single datasets (D-1∼D-5)	**93.73**	92.51	93.12	93.04
	Combined datasets (D-1C∼D-5C)	**94.04**	92.84	93.92	93.89
Forecasting horizon		24 hours ahead			
Model		E-S-ELM	SAE-LM	DBN-SVR	DBN-RBM
Average accuracy (%)	Single datasets (D-1∼D-5)	92.79	91.98	92.81	**92.83**
	Combined datasets (D-1C∼D-5C)	93.00	92.21	**93.73**	93.69

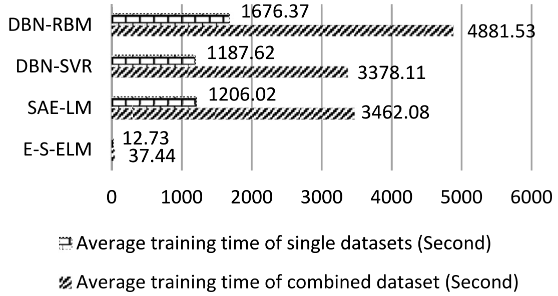

Fig. 5. Comparison of average training time on single and combined datasets.

Table 2 shows the testing accuracies of each model in wind speed forecasting. It can be seen that compared with other deep learning models, the proposed E-S-ELM performs the best in the 12-hour forecasting horizon and achieves 93.73% and 94.04% accuracy on the single datasets and combined datasets, respectively. However, in the 24-hour forecasting horizon, the DBN-RBM achieves 92.83% accuracy on the single datasets and the DBN-SVR achieves 93.73% accuracy on the combined datasets, which are the best results of all the models.

Interestingly, compared to the results obtained from the single datasets, the accuracy of E-S-ELM only improves 0.31% in the 12-hour forecasting horizon and improves even less (0.21%) in the 24-hour forecasting horizon on the combined datasets. Similarly, the accuracy of the SAE-LM model on the combined datasets only improves 0.33% and 0.23% in the 12- and 24-hour forecasting horizon, respectively. In contrast, the accuracy of DBN-SVR and DBN-RBM improves at least 0.8% in the 12-hour forecasting horizon and improves slightly more (at least 0.86%) in the 24-hour forecasting horizon on the combined datasets compared to the single datasets. Using the shared-hidden-layer structure (combined datasets) in the deep structure models can improve the accuracy. Nonetheless, the E-S-ELM and the SAE-LM do not improve as much as the other models, especially when the forecasting horizon is increased from 12 to 24 hours. From the above experiment results, we can infer that the shared-hidden-layer structure is effective in deep learning models, but it may not greatly improve the accuracy of SAE-related deep learning models in a longer forecasting horizon. Fig. 5 shows the average training time of each model on the single and combined datasets. Although the proposed E-S-ELM performs slightly worse in the 24-hour forecasting horizon, it can still achieve at least 92.79% accuracy with an approximately ninety times faster training time compared to DBN-SVR and DBN-RBM.

5. Conclusion

In this paper, we proposed a hybrid model, E-S-ELM, comprising empirical mode decomposition, stacked auto-encoders, and an extreme learning machine. Since the shared-hidden-layer architecture is considered to be a universal feature transformation approach which can transfer knowledge from data-rich to data-poor sources, we also evaluate its capability in this research. The proposed E-S-ELM is applied to real-world wind speed data collected from five airports in the U.K. and is compared with current deep learning models, SAE-ELM, DBN-SVR, and DBN-RBM. The evaluation results show that the E-S-ELM model has superior performance (average of 93.73% on single datasets and 94.04% on combined datasets) in the 12-hour forecasting horizon. Although the E-S-ELM does not generate the best forecasting results in the 24-hour forecasting horizon compared to other deep learning models, it can still achieve an average of 92.79% on single datasets and 93.00% on combined datasets with at least a ninety times faster training time. Moreover, the experiment results also indicate that the shared-hidden-layer is effective when combined with deep learning approaches. The most important contribution of this paper is that it shows the potential of the proposed model and the shared-hidden-layer structure for the ordinary computational intelligence forecasting approach.

As future work, we will employ the E-S-ELM and the shared-hidden-layer architecture on other time series data; and we will continue to apply deep structure networks to deal with complex time series problems.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgement

We acknowledge the support of the Broadspectrum Asset Management Division, Australia. We would also like to thank the referees and the editors for their helpful comments and suggestions.

References

[1] J.M. Carrasco, et al., Power-electronic systems for the grid integration of renewable energy sources: a survey, IEEE Trans. Ind. Electron. 53 (4) (2006) 1002–1016.
[2] S. Weitemeyer, D. Kleinhans, T. Vogt, C. Agert, Integration of renewable energy sources in future power systems: the role of storage, Renew. Energy 75 (2015) 14–20.
[3] V. Quaschning, Understanding Renewable Energy Systems, Routledge, 2016.
[4] D.Y. Leung, Y. Yang, Wind energy development and its environmental impact: a review, Renew. Sustain. Energy Rev. 16 (1) (2012) 1031–1039.
[5] S.S. Soman, H. Zareipour, O. Malik, P. Mandal, A review of wind power and wind speed forecasting methods with different time horizons, in: North American Power Symposium 2010, IEEE, 2010, pp. 1–8.
[6] Y.-K. Wu, J.-S. Hong, A literature review of wind forecasting technology in the world, in: 2007 IEEE Lausanne Power Tech., IEEE, 2007, pp. 504–509.
[7] A.M. Foley, P.G. Leahy, A. Marvuglia, E.J. McKeogh, Current methods and advances in forecasting of wind power generation, Renew. Energy 37 (1) (2012) 1–8.
[8] G. Sideratos, N.D. Hatziargyriou, An advanced statistical method for wind power forecasting, IEEE Trans. Power Syst. 22 (1) (2007) 258–265.
[9] D. Liu, D. Niu, H. Wang, L. Fan, Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm, Renew. Energy 62 (2014) 592–597.
[10] Z. Guo, W. Zhao, H. Lu, J. Wang, Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model, Renew. Energy 37 (1) (2012) 241–249.
[11] A.A. Abdoos, A new intelligent method based on combination of VMD and ELM for short term wind power forecasting, Neurocomputing 203 (2016) 111–120.
[12] J.N. Liu, Y. Hu, Y. He, P.W. Chan, L. Lai, Deep neural network modeling for big data weather forecasting, in: Information Granularity, Big Data, and Computational Intelligence, Springer, 2015, pp. 389–408.
[13] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[14] M. Jordan, T. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.
[15] H. Liu, C. Chen, H.-q. Tian, Y.-f. Li, A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks, Renew. Energy 48 (2012) 545–556.
[16] H.-F. Yang, T.S. Dillon, Y.-P.P. Chen, Optimized structure of the traffic flow forecasting model with a deep learning approach, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2371–2381.
[17] H.-F. Yang, Y.-P.P. Chen, Hybrid deep learning and empirical mode decomposition model for time series applications, Expert Syst. Appl. 120 (2019) 128–138.
[18] X. Qiu, L. Zhang, Y. Ren, P.N. Suganthan, G. Amaratunga, Ensemble deep learning for regression and time series forecasting, in: 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning, CIEL, IEEE, 2014, pp. 1–6.
[19] L.L.C. Kasun, H. Zhou, G.-B. Huang, C.M. Vong, Representational learning with ELMs for big data, IEEE Intell. Syst. 28 (6) (2013) 31–34.
[20] Q. Hu, R. Zhang, Y. Zhou, Transfer learning for short-term wind speed prediction with deep neural networks, Renew. Energy 85 (2016) 83–95.
[21] T. Kuremoto, S. Kimura, K. Kobayashi, M. Obayashi, Time series forecasting using a deep belief network with restricted Boltzmann machines, Neurocomputing 137 (2014) 47–56.
[22] A. Lendasse, E. Oja, O. Simula, M. Verleysen, Time series prediction competition: the CATS benchmark, Neurocomputing 70 (13–15) (2007) 2325–2329.
[23] N.E. Huang, M.L. Wu, W. Qu, S.R. Long, S.S. Shen, Applications of Hilbert–Huang transform to non-stationary financial time series analysis, Appl. Stoch. Models Bus. Ind. 19 (3) (2003) 245–268.
[24] J. An, J.X. Yu, C.A. Ratanamahatana, Y.-P.P. Chen, A dimensionality reduction algorithm and its application for interactive visualization, J. Vis. Lang. Comput. 18 (1) (2007) 48–70.
[25] S. Yang, Z. He, Y.-P.P. Chen, Workload-based ordering of multi-dimensional data, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 831–844.
[26] J. Rong, G. Li, Y.-P.P. Chen, Acoustic feature selection for automatic emotion recognition from speech, Inf. Process. Manag. 45 (3) (2009) 315–328.
[27] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognit. Lett. 42 (2014) 11–24.
[28] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.
[29] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.
[30] J. Tang, C. Deng, G.-B. Huang, Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2016) 809–821.
[31] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[32] N. Wanas, G. Auda, M.S. Kamel, F. Karray, On the optimal number of hidden nodes in a neural network, in: 1998. IEEE Canadian Conference on Electrical and Computer Engineering, vol. 2, IEEE, 1998, pp. 918–921.
[33] H.-F. Yang, T.S. Dillon, E. Chang, Y.-P.P. Chen, Optimized configuration of exponential smoothing and extreme learning machine for traffic flow forecasting, IEEE Trans. Ind. Inform. 15 (1) (2019) 23–34.
[34] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.