Wind Turbine Power output Prediction Using a New Hybrid Neuro-Evolutionary Method

Article *in* Energy · April 2021

DOI: 10.1016/j.energy.2021.120617

Mehdi Neshat
University of South Australia
75 PUBLICATIONS **997** CITATIONS
SEE PROFILE

Meysam Majidi Nezhad
Sapienza University of Rome
28 PUBLICATIONS **155** CITATIONS
SEE PROFILE

Ehsan Abbasnejad
University of Adelaide
69 PUBLICATIONS **427** CITATIONS
SEE PROFILE

Seyedali Mirjalili
Griffith University
265 PUBLICATIONS **26,799** CITATIONS
SEE PROFILE

Wind Turbine Power output Prediction Using a New Hybrid Neuro-Evolutionary Method

Mehdi Neshat[1,4], Meysam Majidi Nezhad[2], Ehsan Abbasnejad[3], Seyedali Mirjalili[4,5], Daniele Groppi[2], Azim Heydari[2], Lina Bertling Tjernberg[6], Davide Astiaso Garcia[7], Bradley Alexander[1], Qinfeng Shi[3], Markus Wagner[1],

[1] *Optimisation and Logistics Group, School of Computer Science, University of Adelaide, Australia,*
mehdi.neshat@adelaide.edu.au, bradley.alexander@adelaide.edu.au, markus.wagner@adelaide.edu.au

[2] *Department of Astronautics, Electrical and Energy Engineering (DIAEE), Sapienza University of Rome, Italy,*
meysam.majidinezhad@uniroma1.it daniele.groppi@uniroma1.it

[3] *The Australian Institute for Machine Learning, The University of Adelaide, Adelaide, Australia,*
ehsan.abbasnejad@adelaide.edu.au, javen.shi@adelaide.edu.au

[4] *Center for Artificial Intelligence Research and Optimization, Torrens University Australia, Brisbane, QLD 4006, Australia,*
ali.mirjalili@gmail.com

[5] *Yonsei Frontier Lab, Yonsei University, Seoul, Republic of Korea*

[6] *School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology Stockholm, Sweden,*
Linab@kth.se

[7] *Department of Planning, Design, and Technology of Architecture, Sapienza University of Rome, Italy,*
davide.astiasogarcia@uniroma1.it

Abstract

Short-term wind power prediction is challenging due to the chaotic characteristics of wind speed. Since, for wind power industries, designing an accurate and reliable wind power forecasting model is essential, we deployed a novel composite deep learning-based evolutionary approach for accurate forecasting of the power output in wind-turbine farms , which is developed in three stages. At the beginning stage (pre-processing), the k-means clustering method and an autoencoder are employed to detect and filter noise in the SCADA measurements. In the Next step (decomposition), in order to decompose the SCADA time-series data, we proposed a new hybrid variational mode decomposition (HVMD) method, that consists of VMD and two heuristics: greedy Nelder-Mead search algorithm (GNM) and adaptive randomised local search (ARLS). Both heuristics are applied to tune the hyper-parameters of VMD that results in improving the performance of the forecasting model. In the third phase, based on prior knowledge that the underlying wind patterns are highly non-linear and diverse, we proposed a novel alternating optimisation algorithm that consists of self-adaptive differential evolution (SaDE) algorithm and sine cosine optimisation method as a hyper-parameter optimizer and then combine with a recurrent neural network (RNN) called Long Short-term memory (LSTM). This framework allows us to model the power curve of a wind turbine on a farm. A historical dataset from supervisory control and data acquisition (SCADA) systems were applied as input to estimate the power output from an onshore wind farm in Sweden. Two short time forecasting horizons, including ten minutes ahead and one hour ahead, are considered in our experiments. The achieved prediction results supported the superiority of the proposed hybrid model in terms of accurate forecasting and computational runtime compared with earlier published hybrid models applied in this paper.

Keywords: Neuro-Evolutionary Algorithms, Alternating optimisation algorithm, Recurrent Deep Learning, Long short-term memory neural network, Adaptive Variational mode decomposition, power prediction model, Wind Turbine, Power curve

Nomenclature

Table 1: All used abbreviations in this paper are listed in alphabetical order:

abbreviation	**full name**
ADMM	Alternate Direction Method of Multipliers
AOA	Alternating optimisation algorithm
ARIMA	Auto-regressive integrated moving average models
AI	artificial intelligence
ANFIS	Adaptive neuro-fuzzy inference system
ANN	Artificial Neural networks
ARLS	Adaptive randomised local search
ARMA	Auto-regressive moving average models
Bi-LSTM	Bidirectional Long short-term memory network
BP	Back-propagation
BPNN	Back Propagation Neural Network
CBOD	Clustering Based outlier Detection
CMA-ES	Covariance matrix adaptation evolution strategy
CMAES-LSTM	Covariance matrix adaptation evolution strategy long short-term memory network
CSO	Cuckoo Search Optimization method
DA	Dragonfly algorithm
DE	Differential evolution
DE-LSTM	Differential evolution long short-term memory network
DNN	Deep neural networks
EAs	Evolutionary algorithms
EEMD	Ensemble empirical mode decomposition
EFADE	Enhanced fitness-adaptive differential evolution algorithm
EFADE-LSTM	Enhanced fitness-adaptive differential evolution algorithm long short-term memory network
ELM	Extreme learning machines
EMD	Empirical mode decomposition
ENN	Elman neural networks
EO	Equilibrium Optimiser
ESN	Echo State Network
FFNN	Feed-forward neural networks
FS	Feature selection
f-ARIMA	fractional version of Auto-regressive integrated moving average models
GA	Genetic algorithm
GNM	Greedy Nelder-Mead search algorithm
HVMD	Hybrid variational mode decomposition method
IMF	Intrinsic mode functions
LSTM	Long short-term memory network
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MPA	Marine predators algorithm
MSE	Mean square error
NAR	Nonlinear autoregressive neural networks
NNCT	No negative constraint theory
NM	Nelder-Mead simplex direct search method
NWP	Numerical weather prediction method
PCA	principal component analysis
PEO	Population extremal optimization
PE	Permutation entropy
PNN	Polynomial neural networks
PSO	Particle Swarm Optimisation
RLS	Randomised local search
RMSE	Root mean square error
RNN	Recurrent neural networks
SaDE	self-adaptive differential evolution algorithm
SCA	Sine cosine algorithm
SCA-LSTM	Sine cosine algorithm long short-term memory network
SCADA	Supervisory control and data acquisition
SSA	Singular spectrum analysis
SVM	Support vector machines
SVR	Support vector regression
WT	Wavelet transforms

Table 2: Symbols applied in the equations of this paper

Symbols	Description	Symbols	Description
P	Power output of wind turbine	ρ	Density of air
R_r	Rotor radius	w	Wind speed
$\hat{Z}$	Normalized wind speed	f_p	Predicted SCADA values
f_o	Observed SCADA values	$\overline{f}_p$	Means of the predicted power measures
$\overline{f}_o$	Means of the perceived power measures	$f(t)$	Original signal
t	Time script	K	Entire number of the modes
u_k	signifies the k_{th} mode	$\delta(t)$	The Dirac distribution
w_k	The center frequency	$\otimes$	The convolution operator
α	Parameter of balancing	n_d	Total number of decomposition iterations
$\hat{f}(w)$	Fourier transforms of $f(t)$	$\hat{u}_i(w)$	Fourier transforms of $u(t)$
$\hat{\lambda}(w)$	Fourier transforms of $\lambda(t)$	X	Candidate of decomposition parameters
Y	New candidate after mutation	$flag_s$	Number of successful mutation
τ	Threshold of the average performance	σ	Mutation step size
UB	Upper bound of the decision variables	LB	Lower bound of the decision variables
μ	Mean of normal distribution	$ac-per$	performance accumulation
$\tilde{c}_t$	Transmission centre of LSTM cell state	f_t	Output value of LSTM forgetting gate
b	Bias value	σ_l	Activation function
$\odot$	Hadamard product	F	Mutation rate
P_{cr}	Probability crossover rate	λ	Population size
Max_{iter}	Maximum number of iteration	f_b	best-found values before using optimiser
$\hat{f}_b$	best-found fitness values after using optimizer	a_i	i^{th} Optimiser
$\hat{U_{a_i}}$	Accumulated contribution of optimizer a_i	I_{a_i}	Fitness improvement
B_S	LSTM batch size	L_R	LSTM learning rate
Pop^t	Initial generated population of solutions	T_{GMax}	Maximum generation number
N_s	Number of successful mutation	N_f	Number of failed mutation
N_h	Number of layers	N_n	Number of neurons in each layer
Op	LSTM optimiser	PF	Penalty factor
L_m	Lagrange Multiplier	IMF	Intrinsic mode functions

1. Introduction

Considering the accelerated expansion of the global energy demand and the boundary of the traditional power resources, wind power technologies, as one of the popular renewable energies, have been advanced considerably and gained global attention. [64]. And also it is carefully considered and planned as a promising sustainable power resource for electric production, especially in terms of minimising the global warming effect all around the world. As renewable energy systems comprised of several sub-components and each element has considerable control parameters to enhance its performance, the application of techno-economic analysis [25], optimisation [3] and measurement methods play a pivotal role to develop these systems such as in wave energy systems [78], wind energy [46], hybrid energy system [73] and heating and cooling system [17]. Among various kinds of renewable energy, wind energy is a promising and popular sustainable source of power that has a much smaller impact on the environment in comparison with burning fossil fuels. With falling costs and large-scale production of generators, the deployment of wind energy is accelerating. For example, gross installations of an onshore and offshore wind farm in the EU were 0.3 GW in 2008 and increased to 3.2 GW in 2017 [36]. With such large increases in the deployment of wind energy forecasting the power output of installed wind turbines is becoming vitally important. However, as local wind environments in wind farms are complex, and as the responses of wind turbines are non-linear and dependent on the condition of the turbine, wind power forecasting is a challenging problem [45].

Wind power forecasting is fundamental to the effective integration of wind farms into the power grid. For a single turbine, the following equation describes power output: P.

$$P = \frac{1}{2}\rho\pi R_r^2 C_p w^3 \tag{1}$$

The terms ρ, R_r and w refer to the density of air, the rotor radius and the wind speed respectively. Meanwhile, C_p is the power coefficient of the proportion of available power , that the turbine is able to extract. The theoretical estimation of wind turbine power is depicted by Equation 1. This equation describes a smooth s-shaped power curve that resembles a logistic function with wind on the x-axis.

However, because of the variable nature of wind and complex dynamics within and between turbines, the real power output of individual wind turbines is not precisely described by this curve [69, 53]. A more realistic alternative model for each wind turbine in a farm can be derived by fitting observations to field data [45].

On top of the task of modelling wind turbine power in response to *current* wind conditions, managers of wind farms also need to forecast *future* power output based on current conditions. Recent works used complex data-driven models such as artificial neural networks (ANNs) to forecast turbine output with some degree of accuracy [41, 68, 55].

To analyse the primary challenges in improving the wind power forecasting models, the main research gaps can be listed as follows:

1. First and foremost, The efficiency of deep neural networks heavily based on the hyper-parameter tuning optimisation plan. Nevertheless, the meta-heuristics applied to improve the hyper-parameters are, commonly incompetent, as initialising the control parameters is challenging and time-consuming. Furthermore, most of them have been refined based on numerical benchmarks, rather than using the hyper-parameter tuning on the target problem.
2. To obtain the high achievement of the decomposition techniques, the decomposition parameters initialisation operates an essential function. This initialisation process faced with some challenges, because inadequate setting options can have a negative effect on the effectiveness of the forecasting models. Another weakness of filter-based models is that the computational runtime needed grows significantly as the number of parameters has progressed [23].

In this paper, we proposed an integrated approach that couples self-adaptive differential evolution with ANNs for accurate short term wind power forecasting. The input features used in our modelling are current wind speed, current wind direction and (in some models) current power output.

The main contributions of this paper are as follows:

1. A new hybrid deep learning-based evolutionary framework (AVMD-AOA-LSTM) is proposed for short-term wind turbine power output forecasting.
2. An advanced data filtering technique is implemented on the training observations (from SCADA data) using K-means clustering [2] and autoencoder neural-networks [79] to detect outliers;
3. To deal with the noise and nonlinearity of wind data characteristics, we developed a novel decomposition algorithm consists of Variational mode decomposition (VMD) and two heuristic algorithms: Adaptive Variational mode decomposition (AVMD) and Greedy Nelder-mead (GNM) search method. In order to measure the complexity of the decomposed sub-signals, permutation entropy [6] (PM) is used.
4. A comparison of the performance of the models (using raw and clean SCADA datasets) was completed, in terms of the models' ability to assess the impact of the outlier detection method.
5. A comparison of the performance of four forecasting models trained to act on different subsets of SCADA inputs was performed. Figure 1 illustrates the models and their inputs; These sub-sets are wind speed for model one; wind speed and wind direction for model two; wind speed and current power output for model three; and, finally, wind speed, wind direction and current power output for model four.
6. Finally, as there is no straightforward theory with regards to design and tune the hyper-parameters of an LSTM network [28], we tuned the model structure and hyper-parameters using a novel hybrid optimisation method that combines self-adaptive differential evolution (SaDE) [66] and Sine Cosine Algorithm (SCA) [48] with two forecasting horizons of ten-minute and one-hour ahead; .Furthermore, The obtained results were compared with 13 state-of-the-art hybrid models, and grid search.

The remainder of this article is organized as follows: Section 2 reviews current approaches to building wind power forecasting models. Section 3 describes the features of the SCADA datasets employed in this

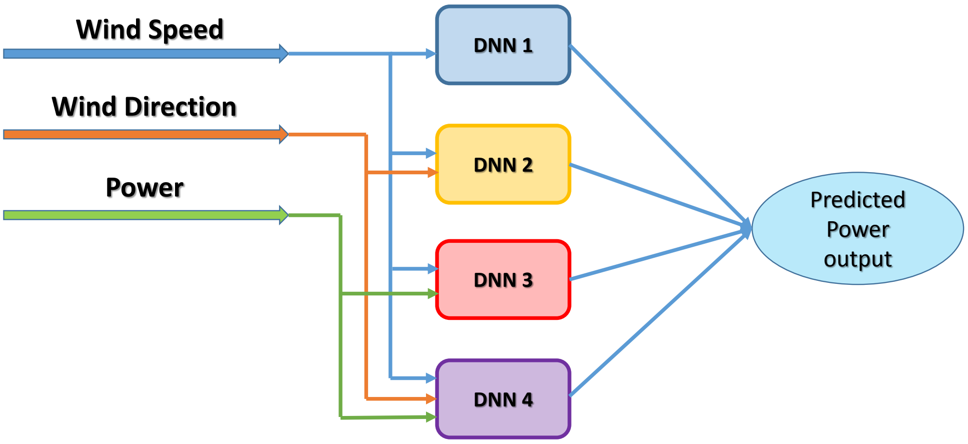


Figure 1: The proposed four different independent forecasting models.The applied power as an input is the current generated power by the wind turbine.

research, collected from 42-months' high-frequency monitoring of onshore wind turbines. Section 4 describes the new outlier detection method used in this work. The details of proposed forecasting models and optimisation algorithms are presented in Section 5. Section 6 describes the experimental methodology. Next, the power prediction results trained by datasets of both raw and clean SCADA datasets are demonstrated. Finally, Section 7 summarises the contributions of this work. Table 1 lists whole abbreviations applied in this work which are sorted in alphabetical order.

2. Related work

Previous work in this field combined physical models with numerical weather prediction (NWP) to forecast wind turbine output [37]. While such methods are fast, they lack accuracy. More accurate predictions can be obtained using statistical methods to model the relationship between the inputs of the system and the corresponding outputs. Commonly applied statistical methods include time-series methods [8, 34], machine learning methods [9, 27], the persistence method [40] and Kalman filtering [89].

One set of well-known time series forecasting methods for wind power prediction is known as autoregressive modelling. Such modelling includes autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and a fractional version of ARIMA (f-ARIMA) [82]. Both ARMA and ARIMA models can capture short-range correlations between inputs and outputs, and the f-ARIMA method is well adapted for representing the time series data with long memory characteristics [34]. In [70] Shamshirband et al. applied a Support vector regression (SVR) model in forecasting the optimal conditions of the wind turbine reaction torque which leads to achieving the maximum level of energy from the wind.

One of the adaptive neural models which have been widely considered to deal with the changeability and fluctuation of wind speed is the adaptive neuro-fuzzy inference system (ANFIS) [63]. Petkovic et al. [60] proposed the ANFIS with a back-propagation learning algorithm to predict the optimal coefficient values of the wind turbines power output. In another investigation, with regard to optimising the average power output of a wind turbine, a smart controller proposed [61]. This controller consists of an ANFIS, a wind generator implemented with the continuously variable transmission (CVT), and showed robust performance. A wind farm project management which is a combination of the function approximation and regression (ANFIS) was adopted to maximise the produced power of a wind turbine using forecasting the wake effects such as waking wind speed [59]. Wang et al. [43] suggested a hybrid neural forecast engine model which consists of an improved Elman neural network and empirical mode decomposition (EMD) with a sliding window, in order to maximise the pertinency and reduce the redundancy measure.

In order to predict the wind speed series, Petkovic et al. [62] proposed the application of fractal interpolation idea and artificial intelligence method with various training algorithms including Back-propagation (BP) and extreme learning machine (ELM). Petkovic et al. represented that the ELM performs better than the BP training approach. Moreover, from [56], a hybridisation of the ANFIS and fractal interpolation prediction suggested improving the performance of the forecasting wind speed model. Recently, a short-term forecasting model for a wind-solar power system was introduced [50] which includes a feature selection filter plus a hybrid forecast engine. The hybrid model depends on a neural network (NN) combined with an evolutionary algorithm. Concerning to design of an accurate wind signal prediction model, Leng et al. [39] represented a mixture of a ridgelet transform (as a decomposition model), a feature selection technique, and finally, a neural system with a meta-heuristic algorithm.

With regard to handling the disadvantages of physical and statistical models, recently a considerable number of estimation techniques have been produced which are mainly depend on the machine learning models. These models are able to perform with high performance in terms of forecasting nonlinear and non-stationary wind speed time-series data.

Another effective technique for forecasting time series data is recurrent neural networks (RNNs). Olaofe et al. [57] applied RNNs to predict the wind turbine power output one-day ahead. However, the applied 'tanh' activation function used in this work leads to disappearing and exploding gradients, which lead to difficultly in training an accurate model. Long short-term memory networks (LSTMs) were introduced in [47], partly to help avoid these issues. LSTMs can learn the correlations that exist in time series data with some accuracy. In [88], LSTMs were employed for short-term predictions of wind power. That study showed that LSTMs could outperform traditional ANNs and support vector machines (SVMs) in terms of prediction accuracy. A combination of principal component analysis (PCA) and an LSTM forecasting model was proposed in [80], and compared with a back-propagation (BP) neural network and an SVM model. The study found that the PCA-LSTM framework results produced higher forecasting accuracy than other methods. Recently, Erick et al. [44] defined a new architecture for wind power forecasting composed of LSTM blocks that replaced the hidden units in the Echo State Network (ESN). The authors also used quantile regression to produce a robust estimation of the proposed forecast target. Moreover, Yu et al. [81] used an LSTM with an enhanced forget-gate network model (LSTM-EFG) combined with a Spectral Clustering method to forecasting wind power. This technique resulted in considerably increased accuracy.

Recently, multi-layer forecasting models have been absorbed a considerable number of the researcher to develop short-term wind speed prediction models. a two-layer forecasting model [12] proposed to predict the short-term wind speed consists of extreme learning machine (ELM), Elman neural network (ENN), LSTM, and ELM-based nonlinear aggregated technique to relieve the natural deficiency of the linear combination model.In other work [85], to control and predict a short-term traffic flow, a combination of LSTM, population extremal optimization (PEO) method, and no negative constraint theory (NNCT) weight integration was introduced (EnLSTM-WPEO). Furthermore, Chen et al. [11] suggested a nonlinear-learning ensemble of hybrid time-series estimation for wind speed forecasting which is composed of the LSTMs, SVRM (support vector regression machine) and EO (extremal optimization algorithm). However, none of the above studies used an automated method to tune the hyper-parameters of their ANN models. Such automatic tuning helps with porting models to a new setting and makes it possible to more rigorously compare modelling approaches.

One recent work that has used hyper-parameter tuning is that of Qin et al. [71] who used the Cuckoo Search Optimization (CSO) method to improve the performance of a Back Propagation Neural Network (BPNN) by adjusting the connection weights. They reported that the accuracy of the proposed hybrid model was higher than that of other methods for predicting the wind speed time series. Shi et al. [72] used the dragonfly algorithm (DA) to tune RNN hyper-parameters for wind power forecasting.

In another recent work, Peng et al. [58] used Differential Evolution (DE) to optimise LSTM parameters, and the reported results indicated that the hybrid DE-LSTM model is able to outperform traditional forecasting models in terms of prediction accuracy. More recently, Neshat et al. [54] forecasted time series through online hyper-parameter tuning of an LSTM model using CMA-ES. This work improved on earlier work by systematically comparing the impact of tuning strategies, model input sets, and data pre-processing on prediction performance. These comparisons define some of the search landscape in the parameter space

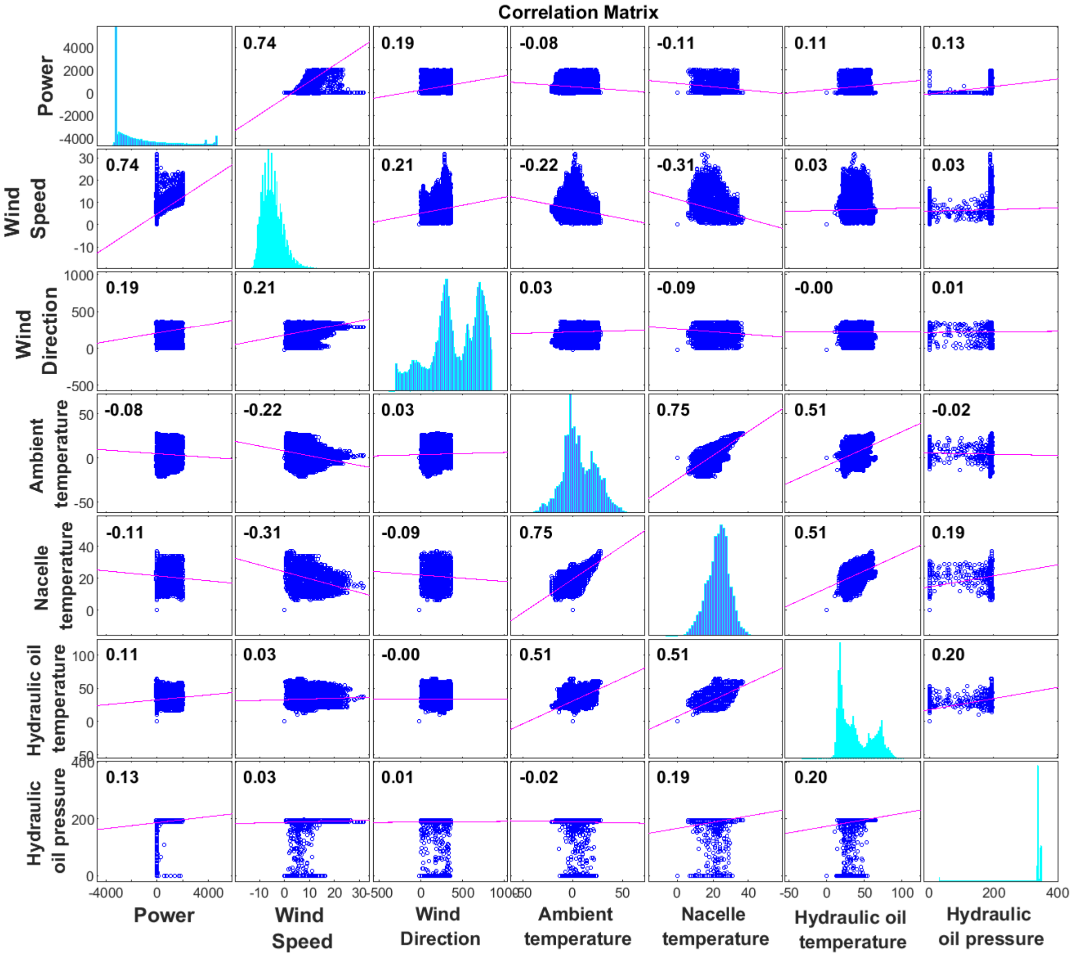


Figure 2: The Pearson's linear correlation coefficients between all pairs of the wind turbine data (SCADA). The correlation plot shows that wind speed, wind direction and Power are highly correlated.

of these algorithms.

3. SCADA data description and analysis

The data analysed in this research comes from six turbines of one onshore wind farm in north-western Europe (Sweden) [30]. For each turbine, 42 months of data are available from January 2013 to June 2016, including 10-minute interval operation data and a log file. Data on faults and maintenance were also stored. In this paper, we select and investigate the SCADA data from the sixth turbine in the wind farm. In order to evaluate and analyse the correlation between power output and other SCADA features, seven features are chosen including wind speed, wind direction, ambient temperature, Nacelle temperature, Hydraulic oil temperature and Hydraulic oil pressure. These are the most recommended SCADA features for power prediction from [41]. Pearson's linear correlation coefficients between all pairs of the wind turbine data features can be seen in Figure 2. The highest correlations are those between Power output and wind speed, as well as between power output and wind direction. Therefore, we select the wind speed and wind

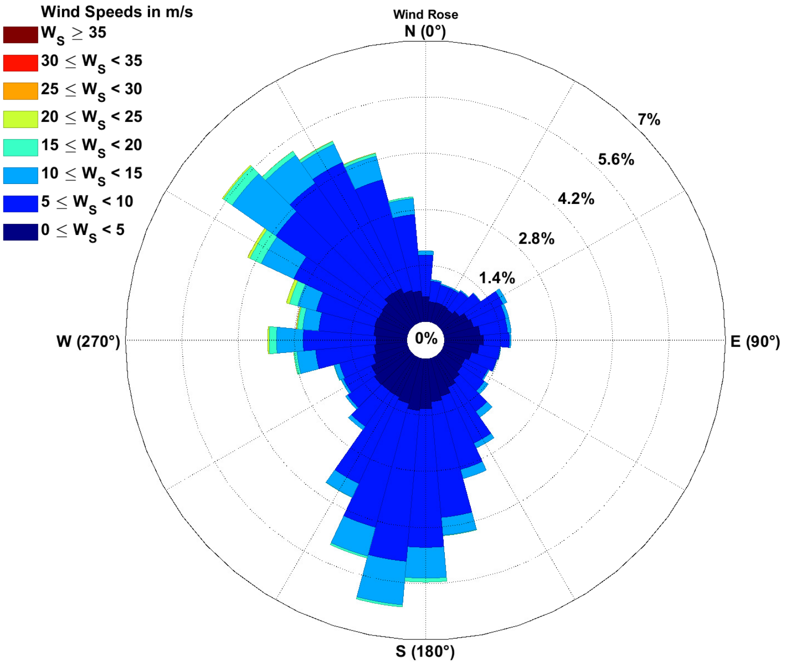


Figure 3: a large view of how the wind speed and wind direction are distributed at the wind farm (Sweden) from 2013 to 2016 (June).

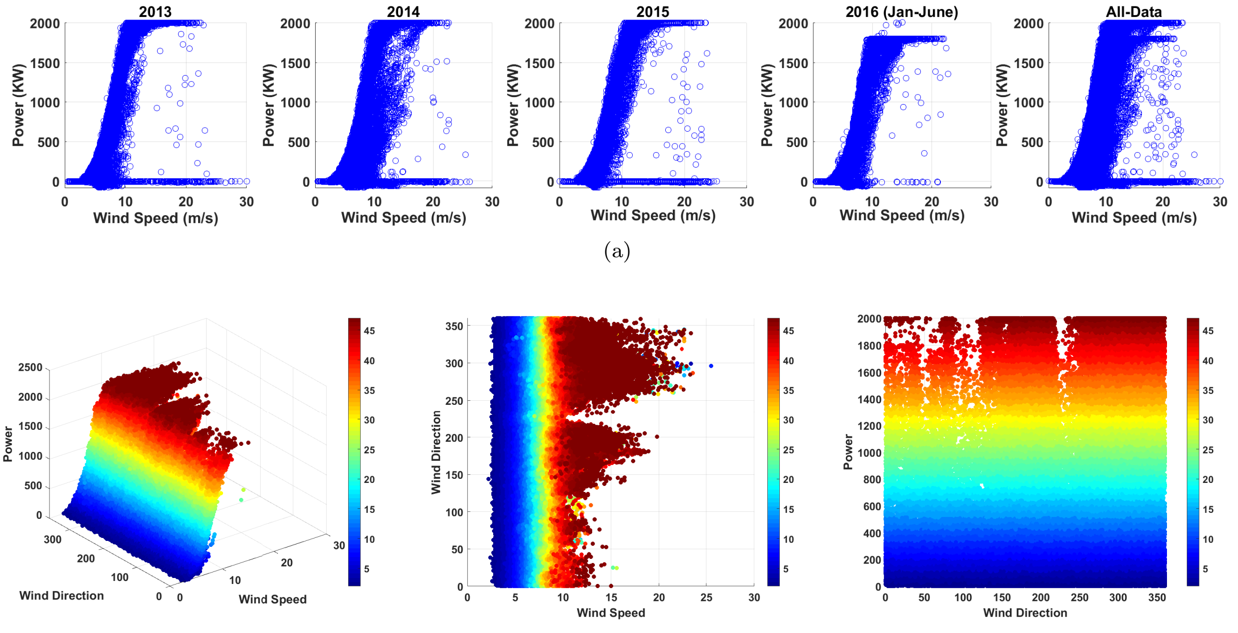
direction as ANN inputs designate generated power as the network output. The diagonal in Figure 2 shows the distributions of each variable, including power. These distributions show some outliers, which might pose stchallenging challenges for modelling. It is also of note that there are some negative values for produced power; these values are caused by stationary turbines spinning up.

Figure 3 depicts the wind rose for the wind farm. It also shows that the dominant wind direction is North-west, and a secondary prevailing direction is South-east. However, there are also occasional West winds.

4. Information preprocessing

In data science, outliers are values that differ from regular observations in a dataset. Figure 4 shows the correlation between wind turbine power output and wind speed; moreover, it represents the correlation between wind turbine power output and wind direction during the 42 months data collection period. The outliers can be seen clearly in the scatter plots; they are distributed on the right side of the plot. In this study, we applied a combination of a K-means method which is one of the well-known Clustering Based Outlier Detection (CBOD) [33] methods and an autoencoder neural network to detect and remove the outliers from the SCADA dataset. As previous studies listed in Section 2, wind speed is the primary factor that determines wind power among the SCADA features. In the data, wind speed is widely distributed; we used the K-Means clustering algorithm to classify the wind power data into K sub-classes. Before the clustering, due to the significant differences in numerical values of each type of data which has a great impact on the training of the autoencoder's latent model, the data is normalised between zero and one [29]. The normalisation used in this paper is described in Equation 2.

$$\hat{Z} = \frac{Z - Z_{min}}{Z_{max} - Z_{min}} \tag{2}$$



(a)

(b)

(c)

(d)

Figure 4: The correlation between wind turbine power output and wind speed over the 42 months of data collection.(a) Outliers can be seen clearly in the data. (b) 3D figure of power curves, wind speed and wind direction. (c) the correlation between wind speed and direction. (d) power curves and wind direction.

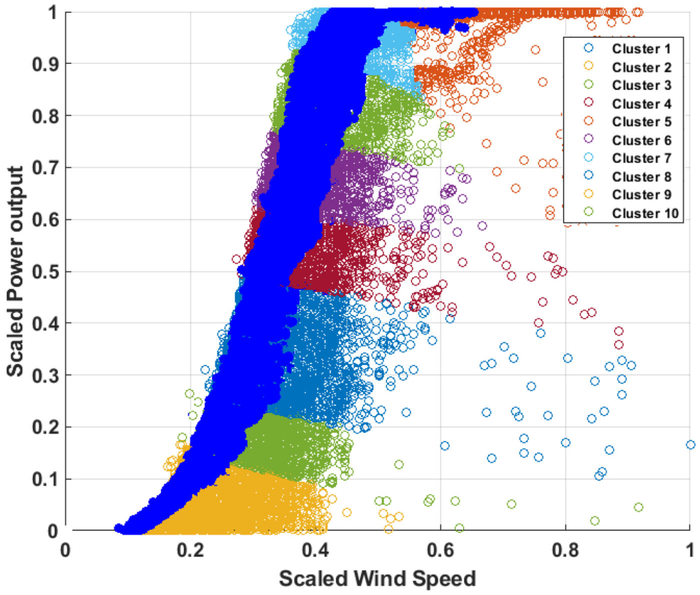


Figure 5: Clustering the data into 10 groups by K-means and then detecting and removing the outliers using an autodecoder NN. The purified data (after removing outliers) is shown in the dark blue region.

For this work, the number of the clusters is set to 10 [29] for wind speed and power output of SCADA data. These clusters indicate the different operation states of the main subsystems, such as the drive train and the control system [15]. Figure 5 shows ten clusters of data from Turbine 6. It can be observed that the distribution in each cluster is a horizontal band. Within these bands, outliers are more easily discerned as being relatively far from the main body of the cluster. In order to remove the outliers in each cluster, an autoencoder neural network is used that shows better performance compared with other traditional outliers detection methods [4, 51]. An autoencoder is a particular type of unsupervised feedforward neural network. This network is trained to reconstruct output in such a way that it becomes similar to each input. In this work, the autoencoder consists of an input layer and one hidden layer which are fully connected [7]. For training an autoencoder, the input data are mapped to the hidden layer where the encoding of input data takes place, which typically comprises fewer nodes than the input layer and consequently compresses the data. Next, from the hidden layer, the reconstructed data flows through the output layer, which is re-transformed in a process called 'decoding', and the squared restoration error between the network's output and its input is calculated. For detecting the outliers, It is noticed that outliers have higher reconstruction error than the norm of the dataset. Therefore we remove the observations which have higher RMSE than the average of all data RMSE. Figure 5 presents the outliers detection and removal process, with the dark blue sections showing the clean data.

4.1. Performance criteria of forecasting models

To evaluate and compare the performance of the applied forecasting models, four broad performance indices are used: the mean square error (MSE), the root mean square error (RMSE), mean absolute error (MAE), and the Pearson correlation coefficient (R) [84]. The equations for MAE, RMSE and R are described as follows :

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|f_p(i) - f_o(i)| \tag{3}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_p(i) - f_o(i))^2} \tag{4}$$

$$R = \frac{\frac{1}{N}\sum_{i=1}^{N}(f_p(i) - \overline{f}_p)(f_o(i) - \overline{f}_o)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_p(i) - \overline{f}_p)^2} \times \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_o(i) - \overline{f}_o)^2}} \tag{5}$$

where $f_p(i)$ and $f_o(i)$ denote the predicted and observed SCADA values at the i^{th} data point. The total number of observed data points is N. The variables $\overline{f}_p$ and $\overline{f}_o$ are the means of the predicted and perceived power measures, respectively. In order to develop the effectiveness of the predicted model, MSE, RMSE and MAE should be minimised, while R should be maximised.

5. Methodology

In this section, we introduce the proposed methodologies and related concepts for short-term wind turbine power output forecasting, including LSTM network details, self-adaptive differential evolution (SaDE) and the hybrid LSTM network and the SaDE algorithm.

5.1. Wind time-series data decomposition

5.1.1. Variational mode decomposition (VMD)

VMD is a modern, fully-intrinsic, quasi-orthogonal decomposition signal processing algorithm, where it can decompose a signal into a series of modes with particular bandwidth in spectral-domain non-recursively [19]. Each mode is packed down a core pulsation defined through the decomposition process. In order to achieve the bandwidth of each mode, three steps are performed [42]: (1) Hibert transform is adopted to arrange a one-sided frequency spectrum for each mode. (2) The transfer frequency spectrum of mode to base-band through incorporating the exponential tune to the corresponding estimated centre frequency. (3) For each mode, determine the bandwidth through using the Gaussian smoothness of the demodulated signal. Next, the constrained variational problem can be addressed as follows [14]

$$\min_{u_k, w_k} \left\{ \sum_{k=1}^{K} \quad \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) \otimes u_k(t) \right] e^{-jw_k t} \right\|_2^2 \right\} \tag{6}$$

$$s.t. \sum_{k=1}^{K} u_k = f(t) \tag{7}$$

where $f(t)$ involves the original signal, t denotes the time script, K is the entire number of the modes, u_k signifies the k_{th} mode, $\delta(t)$ depicts the Dirac distribution, w_k is the center frequency and $\otimes$ is the convolution operator. Furthermore, the mode with high-order represents the low-frequency sub-layers. To transform the mentioned optimization problem to an unconstrained one, both the penalty function and Lagrangian multipliers are operated, which can be indicated in Equation 8.

$$\begin{aligned} L(u_k, w_k, \lambda) =& \alpha \sum_{k=1}^{K} \quad \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) \otimes u_k(t) \right] e^{-jw_k t} \right\|_2^2 \\ &+ \left\| f(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t).f(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \end{aligned} \tag{8}$$

where α denotes the parameter of balancing that is related to the needed data fidelity constraint. The identical unconstrained problem in Equation refeq-LVMD can be figured out using the ADMM (Alternate Direction Method of Multipliers). ADMM is able to obtain the saddle point of the augmented Lagrangian. Both u_k and w_k can be updated in two inclinations to perform the analysis of the VMD based on the ADMM.

In order to solve the optimization problem for u_k, Equation 9 can be defined as:

$$\hat{u}_k^{n_d+1} = \frac{\hat{f}(w) - \sum_{i \neq k} \hat{u}_i(w) + \frac{\hat{\lambda}(w)}{2}}{1 + 2\alpha(w - w_k)^2} \tag{9}$$

where n_d is the total number of iterations, $\hat{f}(w)$, $\hat{u}_i(w)$, $\hat{\lambda}(w)$ and $\hat{u}_k^{n+1}$ are the Fourier transforms of $f(t)$, $u(t)$, $\lambda(t)$, and $u_k^{n_d+1}(t)$, respectively.

5.1.2. Permutation Entropy (PE)

Measuring the complexity of perceived time series data enables us to have a better comprehension of the characteristics of the dynamical system [83]. However, there is a shortage of consent on the complexity representation [20]. Entropy can be one of the best metrics to estimate the signal complexity [5]. Therefore, complexity expresses the level of randomness and unpredictability. Many entropy methods are recently introduced, and some of the popular ones are including permutation entropy (PE) [6], fuzzy entropy [13] and sample entropy [67]. In comparison with other entropy methods, PE is recognised for its benefits of being computationally cheap and conceptually uncomplicated. Additionally, it is identified to apply for all types of signals, such as deterministic, stationary, noisy, chaotic, and stochastic [31].

PE was firstly proposed by Bandt and Pompe [6] in order to measure the complexity rate of time series based on a correlation of neighbouring states. According to the experimental results, PE performance can overcome other similar methods like Lyapunov exponents especially, in several well-known chaotic dynamical systems [6]. Moreover, PE is particularly beneficial in the behaviour of dynamical or observational noise. Some of the PE advantages reported including simplicity, remarkably fast computation, robustness, and invariance concerning nonlinear monotonous transformations.

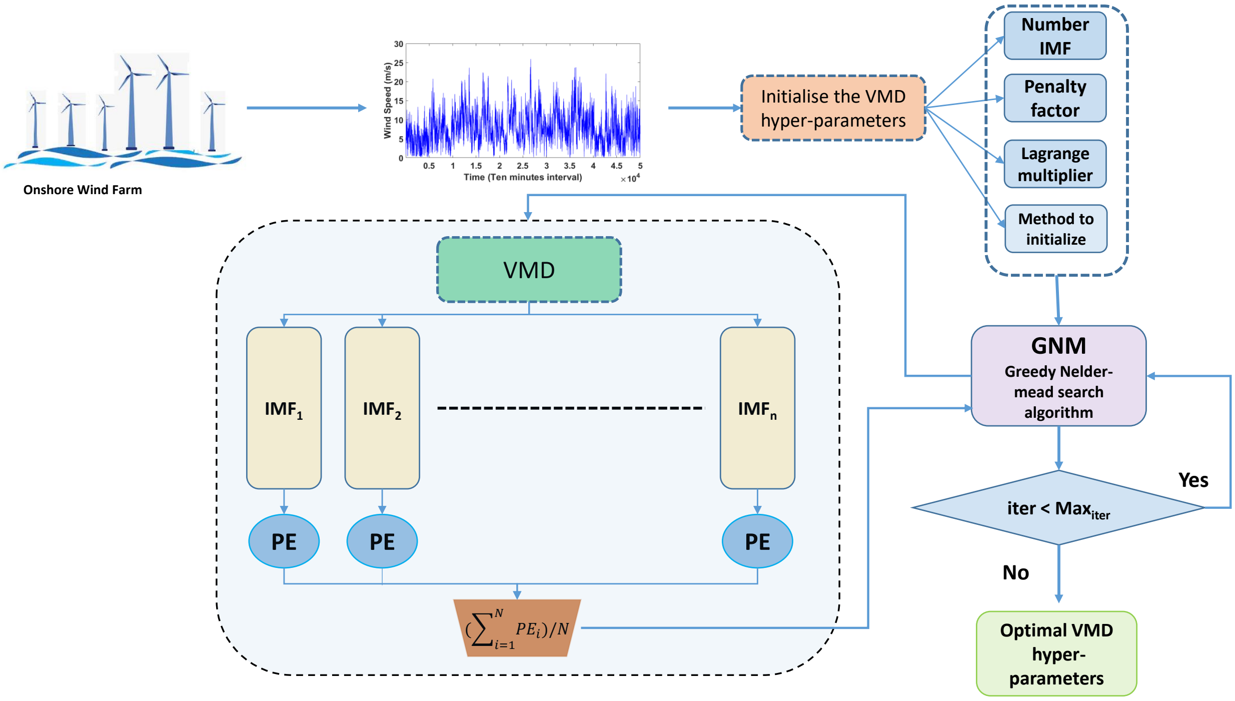


Figure 6: The structure of the proposed Adaptive Variational mode decomposition (AVMD), N is the number of IMF.

5.1.3. Adaptive Variational mode decomposition (AVMD)

Although Variational Mode Decomposition (VMD) has advantages such as suppression noise, fast computation, strong mathematical background and non-recursive sifting process, the decomposition performance of VMD is strongly depends on the number of intrinsic mode functions (IMFs), penalty factor, the method to initialise the central frequencies and the update rate for the Lagrange multiplier which need to be determined in advance. To deal with the issue of VMD hyper-parameter tuning, we propose two fast heuristic algorithms: Greedy Nelder-mead (GNM) search and a randomised local search (RLS) with an adaptive mutation step size. Furthermore, to evaluate the generated solutions using the proposed method, we use the permutation entropy (PE). Indeed, PE indicates the degree of signal randomness, that should be minimised. As illustrated in Figure 6 detailing the applied strategy in the Adaptive Variational mode decomposition (AVMD).

Greedy Nelder-mead (GNM) search method. GNM is a combination of a Nelder-Mead simplex direct search algorithm (NM) and a greedy algorithm. NM is a fast downhill search algorithm especially in the low dimensions and derivative-free method [10] To speed up the convergence rate of the NM, we propose a greedy search algorithm to concentrate the NM exploitation ability on a variable instead of optimising all-in-one.

Adaptive randomised local search (ARLS) . ARLS is a modified version of RLS that is a simple single-based solution evolutionary algorithm (EA) [32]. According to the practical optimisation results, RLS can be more efficient than complicated EA approaches. RLS begins with a solution X and produces a new candidate Y in each iteration by mutating one determined variable of X randomly. In the RLS, the mutation is performed using a uniform distribution which results in a non-curved and noisy local search. Therefore, to handle this issue, we apply a normally distributed mutation. Furthermore, to get a great balance between the exploration and exploitation ability of RLS, we propose an adaptive mutation step size. In order to update the mutation step size, we keep a historical achievement of the last 10 iterations. if more than one-fifth of the mutations in this search phase has been thriving ($flag_s$) and the average performance satisfies the threshold (τ), the mutation step size σ is extended, otherwise, σ has dwindled.

In the following, the pseudo-code of ARLS can be seen by the Algorithm 1 where UB and LB are the upper and lower bound of the decision variables, and also N is the number of variables.

Algorithm 1 Adaptive Randomized Local Search

```
procedure ARLS
Initialization
    $Var = VMD_{hyper-parameters}$                              ▷ decision variables
    $f$ = Permutation Entropy                                   ▷ fitness function
    LB=Min($Var$);UB= Max($Var$)                                ▷ Upper and Lower bounds
    $X_{iter} \in \{LB_i, UB_i\}^{i \in N}$ uniformly at random ▷ Generate first feasible VMD configuration
    $\sigma_i = 0.5 * (UB_i - LB_i)$                            ▷ initial mutation step
    $\rho = 10$                                                 ▷ size of pool to keep the ARLS performance
    while Stopping Criteria do
Mutation
        Create $Y^i_{iter} = X^i_{iter}$ independently for each $i \in \{1, 2, ..., N\}$
        $Y_{iter} = N(\mu, \sigma)$        ▷ Mutate one random variable of $Y_{iter}$by normally distributed random
        $per_{iter}$= $\Delta f(Y_{iter})$                      ▷ Record the performance of ARLS
        $ac\text{-}per = \frac{\sum_{iter}^{iter-10} per_{iter}}{\rho}$   ▷ performance accumulation
Selection
        if $(f(Y_{iter}) \leq f(X_{iter}))$ then
            $X_{iter+1} = Y_{iter}$, $flag_s = flag_s + 1$
        else
            $X_{iter+1} = X_{iter}$, $flag_s = flag_s - 1$
        end if
Update mutation step size
        if (ac−per $> \tau$ & $\frac{flag_s}{\rho} > 0.2$)) then
            $\sigma = \sigma \times 1.5$                        ▷ Extend the step area of mutation
        else
            $\sigma = \sigma \times 0.5$                        ▷ Shorten the mutation step size
        end if
    end while
end procedure
```

5.2. Long short-term memory deep neural network (LSTM)

The LSTM network [47] is a special kind of recurrent neural network (RNN) with three thresholds: the input gate, the output gate and the forgetting gate. The unit structure of the LSTM network can be seen in Figure 7. The forgetting gate defines the permissible rise or drop of the data flow [86] by setting the threshold, which indicates reservation and forgetting. Considering that an RNN hidden layer has only one state, there are severe difficulties with gradient fading and gradient explosion. Augmenting the RNN, the LSTM adds the structure of the cell state, which can recognise the long-term preservation of the state and emphasises the active memory function of the LSTM network. In the case of massive wind power time series data, the network can significantly enhance the accuracy of wind power prediction. In the forward propagation method of the LSTM network, the output value of the forgetting gate f_t can prepare the information trade-off of the unit state and the functional relationship encoded by Equation 10.

$$f_t = \sigma_l(w_f h_{t-1} + u_f x_t + b_f) \tag{10}$$

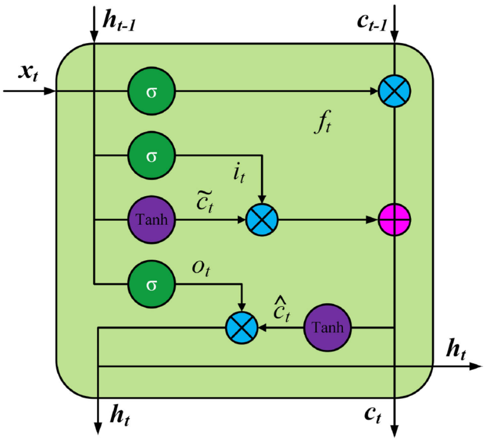


Figure 7: The internal structure of LSTM network from [86].

Both i_t and $\tilde{c}_t$ are generated by the input gate, which are related to the previous moment. The expressions are as shown as Equation 11 and 12.

$$i_t = \sigma_l(w_i h_{t-1} + u_i x_t + b_i) \tag{11}$$

$$\tilde{c}_t = tanh(w_c h_{t-1} + u_c x_t + b_c) \tag{12}$$

Cell state c_t is the transmission centre of the cell state before and later LSTM, which has the following functional relationship:

$$c_t = c_{t-1} \odot f_t + i_t \odot \tilde{c}_t \tag{13}$$

The output h_t of the output gate derives from two components. The first part is the output of the previous moment that is the input of the current moment, and the second part is the information of the current cell state and the particular expression model is delivered as Equation 14 and 15.

$$o_t = \sigma(w_o h_{t-1} + u_o x_t + b_o) \tag{14}$$

$$h_t = o_t \odot tanh(c_t) \tag{15}$$

where both u and w are the weight values; b and σ_l are the bias values and activation function respectively, and $\odot$ is the Hadamard product. For the LSTM network training settings, the Adam algorithm [35] is employed to optimise the loss function, and Dropout [24] is used to prevent model overfitting.

5.3. Optimisation algorithms

In this research, we developed a novel alternating optimisation algorithm (AOA) and propose also a comprehensive optimisation framework to improve LSTM architecture and hyper-parameters tuning. In order to compare the performance of the proposed method with the state-of-the-art optimisation algorithms, eight meta-heuristics are performed including Sine Cosine Algorithm (SCA) [49] , Equilibrium optimizer (EO) [1], Marine predators algorithm (MPA) [21], covariance matrix adaptation evolution strategy [26] (CMA-ES), Differential Evolution (DE) [28], Self-adaptive Differential Evolution (SaDE) [66], and Enhanced Fitness-adaptive Differential Evolution Algorithm (EFADE) [52]. The details of the algorithms evaluated for each strategy are summarised in Table 3.

Table 3: The details of the optimisation methods parameters. All methods are restricted to the same evaluation number.

Methods	Settings
CMA-ES [26]	with the default settings and $\lambda = 12$;
DE [74]	with $\lambda = 24$, $F = 0.5$, $P_{cr} = 0.8$;
EFADE [52]	with $\lambda = 24$, Enhanced Fitness-adaptive Differential Evolution Algorithm: F and P_{cr};
SaDE [66]	with $\lambda = 24$, two adaptation schemes are applied to update the control parameters: F and P_{cr}, $\mu = 0.5$, $\sigma = 0.3$, $C_{Rm} = 25$;
SCA [49]	with $\lambda = 24$, $\alpha = 2$; $r_1 = \alpha$ decreases linearly from α to 0
EO [1]	$\lambda = 24$, and applied the default settings;
MPA [21]	$\lambda = 24$, and applied the default settings;
SCA-SaDE	The same population size, $\lambda = 24$, and applied the default settings;

5.3.1. Self-adaptive Differential Evolution (SaDE)

SaDE [66] is proposed by Qin et al. to concurrently perform two popular mutation strategies "DE/rand/1" and "DE/current-to-best/1". SaDE adjusts the probability that offspring solutions will be generated using each strategy, depending on the success rates (improved solutions) in the past N_f generations of the algorithm. This adaptation scheme aims to evolve the best mutation strategy as the search progresses. This methodology is similar to the ideas proposed in [77], where striving heuristics (including diverse DE variants, simplex methods and evolution strategies) are adopted simultaneously, and probabilities for offspring generation are adjusted dynamically.

In SaDE, the vectors of the mutation factors are generated independently at each iteration based on a normal distribution ($\mu = 0.5$, $\sigma = 0.3$), and trimmed to the interval (0, 2]. This scheme can retain both local (with small F_i values) and global search capability, so as to create potentially suitable mutation vectors during the evolution process. In addition, the crossover probabilities are randomly generated based on independent normal distribution, with $\mu = C_{Rm}$ and $\sigma = 0.1$. This is in contrast to the Fi and C_{Ri} values, which remain fixed for the last five generations before the next regeneration. The C_{Rm} is initially set to 0.5. in order to tune C_R to suitable values, the authors renew C_{Rm} every 25 generations using the best C_R values from the last C_{Rm} update.

To speed up the SaDE convergence rate, a further local search procedure (quasi-Newton method) is used on some competent solutions after N_s generations. The benefits of Self-adaptive parameter control make the SaDE as one of the most successful evolutionary algorithms, especially in the context of real engineering optimisation problems that have multi-modal search spaces with many local optima [87].

5.3.2. Sine Cosine Algorithm (SCA)

Sine-cosine algorithm (SCA) is a modern meta-heuristic algorithm proposed by Mirjalili in 2016 [49]. This algorithm has been applied and studied by many researchers because of simple structure, a few control parameter, fast convergence rate and strong exploration ability [48]. Whole benefits can be accomplished through a creative combine idea of the simple variation of sine and cosine functions performance. It has been successfully applied to solving the parameter optimization of neural networks, multiple hydropower reservoirs operation optimization [22], hydrothermal scheduling (HTS) problem in order to optimise fuel cost [16] and different real-world engineering problems at present [48].

SCA optimisation algorithm is population-based, in which a collection of candidates is generated and then tried to improve the average fitness of the population utilising two principal mathematical formulas: sine and cosine functions. The equations are as follows:

$$X_i^{t+1} = X_i^t + r_1 \times \sin(r_2)|r3 \times P_i^t - X_i^t| \quad if\ r_4 < 0.5 \tag{16}$$

$$X_i^{t+1} = X_i^t + r_1 \times \cos(r_2)|r3 \times P_i^t - X_i^t| \quad if\ r_4 \geq 0.5 \tag{17}$$

where X_i^t is a vector that outlines the current solution, r_1 and r_2 are a random vector representing the magnitude and domain of sin and cos functions. r_3 shows the destination contribution magnitude of the new state of the solution. Finally, r_4 is a random number in [0, 1] to provide the same probability for both operators. The mentioned search operators are designed to fit a balance between the exploratory and exploitative behaviours for the SCA algorithm.

In order to converge to an optimum at the end of the optimisation process, exploration ability should be decreased in population-based algorithms. Therefore, In SCA, the random variable r_1 is linearly decreased as follows:

$$r_1 = \alpha - iter \times \frac{\alpha}{Max_{iter}} \tag{18}$$

where $iter$ is the current iteration, α is a constant, and Max_{iter} determines the maximum number of iteration.

5.3.3. Alternating optimisation algorithm (AOA= SCA+SaDE)

In this work, we develop an alternating optimisation (AOA) framework in order to optimise the hyper-parameters of the applied deep learning model using two popular meta-heuristics Self-adaptive Differential Evolution (SaDE) and Sine Cosine Algorithm (SCA). In this framework, there is a competition between both optimisation methods to achieve the computational budget based on their performance. In the first cycle, both candidate optimisers are assigned with an equal computational budget to solve the problem, and their performances are compared. In the remaining iterations, the candidate optimiser that delivers the greatest contribution to the fitness development will be chosen (allocates computational resources) for optimising. To provide a fair balance between both optimisers, an accumulated contribution to fitness improvement is proposed [75] as following:

$$I_{a_i} = \frac{(\hat{f_b} - f_b)}{\hat{f_b}} \tag{19}$$

where f_b and $\hat{f_b}$ are the best-found fitness values before and after using optimizer a_i in one iteration. Equation 20 shows the overall contribution of optimizer a_i during the optimisation process.

$$U_{a_i} = \frac{(\hat{U_{a_i}} + I_{a_i})}{2} \tag{20}$$

where $\hat{U_{a_i}}$ is the most advanced accumulated contribution of optimizer a_i before starting this iteration. The initial value of $\hat{U_{a_i}}$ is 0.

Moreover, to evaluate the AOA algorithm (SCA+SaDE) thoroughly, we compared AOA's performance against the performances of each optimiser: SCA and SaDE. This hybridisation technique has been suggested a great biased algorithm selection to enjoy the high exploration capability of SaDE, and considerable SCA exploitation ability to generate promising results. Another benefit of this adaptive alternating strategy is prohibiting the premature convergence and the stagnation issue. The pseudo-code of the proposed AOA algorithm for solving the hyper-parameter optimisation problem is explained in Algorithm 2.

5.4. Hybrid Neuro-Evolutionary Deep Learning method

Multiple parameters can influence the precision and performance of LSTM networks. The selected hyper-parameters include the maximum training number of LSTM (Epoch), hidden layer size, batch size, initial learning rate and the optimiser type. If the maximum training number is too small, then it will be difficult for the training data to converge; if we set the number to a large value, then the training process might overfit. The hidden layer size can influence the impact of the fitting [58]. Batch size is also an important hyper-parameter. If the batch size is set too low, then the training data will struggle to converge, resulting in under-fitting. If the batch size is too large, then the necessary memory will rise significantly. There are also complex interactions between hyper-parameters. Therefore, a reliable optimisation technique should be utilised to tune the optimal combination of hyper-parameters and to balance forecasting performance and computational efficiency.

Algorithm 2 *Alternating optimisation algorithm* $(SCA + SaDE)$

```
procedure SCA+SaDE ($hyper - parameters$ )
Initialization
    A = 2                                          ▷ Number of optimiser
    $U_{a_i} = 0$                                  ▷ Initialise the contribution of each optimiser
    $D = size(\mathbb{H})$                         ▷ D is the dimension of the Deep learning model hyper-parameters
     N= population size; UB, LB= upper and lower bounds
     t= current iteration, $T_{GMax}$ = maximum generation number
    $Pop^t$= $LB_j + (UB_j - LB_j) * rand$         ▷ Initialise population randomly
    $fitness$=Eval($x$)                            ▷ Evaluate the population
    for $t$ in $[1, .., T_{GMax}]$ do
        if i==1 then                               ▷ Run both optimiser, In the first iteration
            $f_t^{SCA}$=$SCA(Pop^t)$
            $f_t^{SaDE}$=$SaDE(Pop^t)$
            Compute $I_{a_i}$ and $U_{a_i}$ for both Optimizer
        else
            $< a_i >= arg\ max\{U_{a_i} \rightarrow 1 \leq j \leq |A|\}$
            Applying optimiser $a_i \rightarrow if\ i = 1\ SCA\ else\ SaDE$
            $I_{a_i} = \frac{(\hat{f}_b - f_b)}{\hat{f}_b}$              ▷ Compute contribution
            $U_{a_i} = \frac{(U_{a_i} + I_{a_i})}{2}$                   ▷ Accumulate historical contribution
        end if
    end for
    return $hyper - parameters$
end procedure
```

There are three main methods for tuning hyper-parameters, including 1) manual trial and error, 2) systematic grid search, and 3) meta-heuristic approaches. In this paper, we propose nine meta-heuristic approaches and grid search that can be used to adjust the optimal configuration of settings for the LSTM. The performance of this hybrid forecasting model (AVMD-AOA-LSTM) is compared with that of grid search; eight hybrid neuro-evolutionary methods: CMAES-LSTM [54], and DE-LSTM [58], and also six new composed models including SaDE-LSTM, EFADE-LSTM, SCA-LSTM, EO-LSTM, MPA-LSTM, AOA-LSTM. and ANFIS [65].

In the grid search method, we evaluate and tune only two hyper-parameters of the LSTM: the batch size and the learning rate. Other settings assign a fixed value for the optimizer type, the number of LSTM hidden layers, the hidden layer size, maximum number of epochs by ('adam' [35]) one, 100 and 100 respectively. These values are chosen in order to provide a baseline for the LSTM model evaluation. The ranges of batch size and learning rate are, respectively, selected from $128 \leq BS \leq 2048$ and $10^{-5} \leq LR \leq 10^{-1}$.

The procedures of the proposed hybrid forecasting model are as follows:

- **Step 1**. Data preprocessing. Detecting and removing the outliers using a combination of K-means and Autoencoder. Next, the dataset is divided into three subsets: training, validation, and test sets.

- **Step 2**. Decomposition: a new Adaptive Variational mode decomposition (AVMD) is developed that consists of a Variational mode decomposition, a permutation entropy method, and two optimisation algorithms. These optimisers are applied to explore the search space of the VMD's hyper-parameters and converge to the best VMD configuration.

- **Step 3**. Initialization. The following parameters are set: maximum iteration number of AOA (SCA-SaDE), population size (NP), minimum and maximum crossover rate (C_R), mutation rate (F), and the upper and lower bounds of decision variables, the iteration numbers for updating the control parameters (N_f and N_s) are set.

- **Step 4**. Hyper-parameters tuning of parallel LSTMs: The new hybrid evolutionary algorithm (AOA) is applied to optimise the performance of LSTMs. The fitness values of the offspring population are computed by applying the proposed hyper-parameters in the LSTM. The fitness is the root of the mean square error (RMSE) of the validation set; other performance indices are also computed and recorded. The RMSE should be minimized, and the corresponding individual is the current best solution that achieves this.
- **Step 6**. Training deep learning-based model: using the new generated hyper-parameters, and next, evaluating the trained forecasting models.
- **Step 7**. Stopping criteria: if the maximum iteration is achieved, then SaDE is terminated and the optimum configuration is taken; otherwise, the procedure returns to Step 3.

The combined procedures of the proposed multi-step hybrid short-term power output forecasting strategy can be depicted in Figure 8.

The fitness function of the optimisation process is defined as follows:

$$\begin{aligned} &Argmin \to f = fitness(N_{h_1}, N_{h_2}, ..., N_{h_D}, N_{n_1 h_1}, N_{n_2 h_2}, ...N_{n_D h_D}, L_R, B_S, Op),\\ &Subject-to:\\ &\qquad LN_h \leq N_h \leq UN_h,\\ &\qquad LN_n \leq N_n \leq UN_n,\\ &\qquad 10^{-5} \leq L_R \leq 10^{-1},\\ &\qquad 128 \leq B_S \leq 2048\\ &\qquad 1 \leq Op \leq 3. \end{aligned} \tag{21}$$

where $N_{h_i}, \{i = 1, \dots, D\}$ is the number of hidden layers for the $i-^{th}$ LSTM network and $N_{n_i,h_j}, \{j = 1, \dots, D_l\}$ is the number of neurons in the i^{th} hidden layer of this network. The lower and upper bounds of N_h are presented by LN_h and UN_h , while LN_n and UN_n are the lower and upper bounds of neuron number. The Op is the selected optimizer for optimising the LSTM weights ('sqdm' [38], 'adam' [35], 'rmsprop' [76]).

6. Experimentation designs

6.1. Performance comparison of four proposed models

In the first step of the forecasting power output of the wind turbine, we proposed four recurrent DNN models with different inputs and the same output. The main aim of proposing these forecasting models is to analyse the impact of three SCADA features, wind speed, wind direction and the currently generated power on the predicting accuracy of the power output both ten-minute and one-hour ahead.

To evaluate the performance of four models using raw SCADA data which are randomly categorised into three training (80%), testing (10%) and validating (10%) sets, the LSTM deep network is used. This network is composed of one sequence input layer, one LSTM layer, a fully-connected layer and a regression layer. A grid search method is used to tune the hyper-parameters, batch size and learning rate. Figure 9 presents the performance of the LSTM framework with tuned batch size and learning rate parameters for three forecasting models in the interval of ten-minute. The best performance of model 1 (one input) is obtained where the values of batch size are greater than 512 and the learning rate placed between the range of 10^{-2} and 10^{-4}. The forecasting behaviours of both models 2 and 3 are similar, and the best accuracy occurs for batch sizes greater than 256 and a learning rate between 10^{-3} and 10^{-5}.

Figure 10 shows an average performance (MSE and R) comparison between two forecasting models (1 and 2 and) shows that using the currently generated power as an input plays a significant role in producing an accurate prediction.

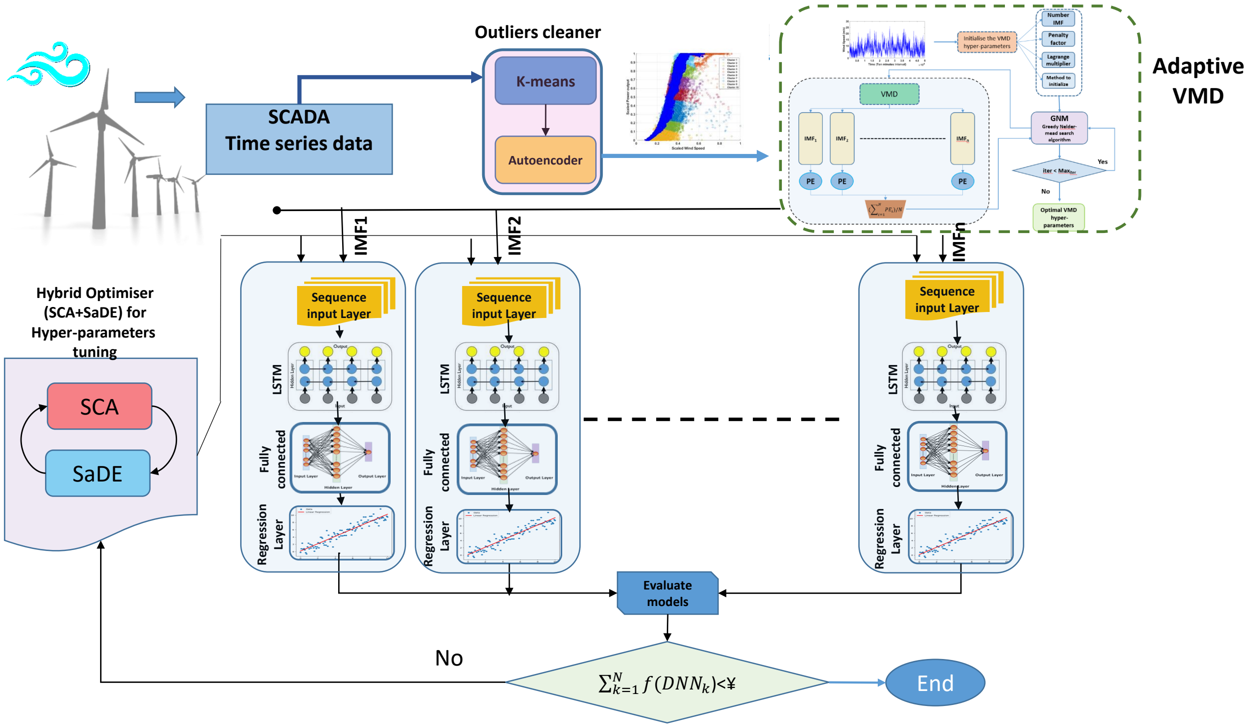


Figure 8: The landscape of the proposed hybrid deep learning-based evolutionary model (AVMD-AOA-LSTM) with adaptive decomposition method.

6.2. *The results of outliers detection techniques*

In the second step, we compare the performance of the proposed models before and after removing the outliers from the SCADA dataset to illustrate the effectiveness of the outlier detection technique (K-means + Autoencoder).

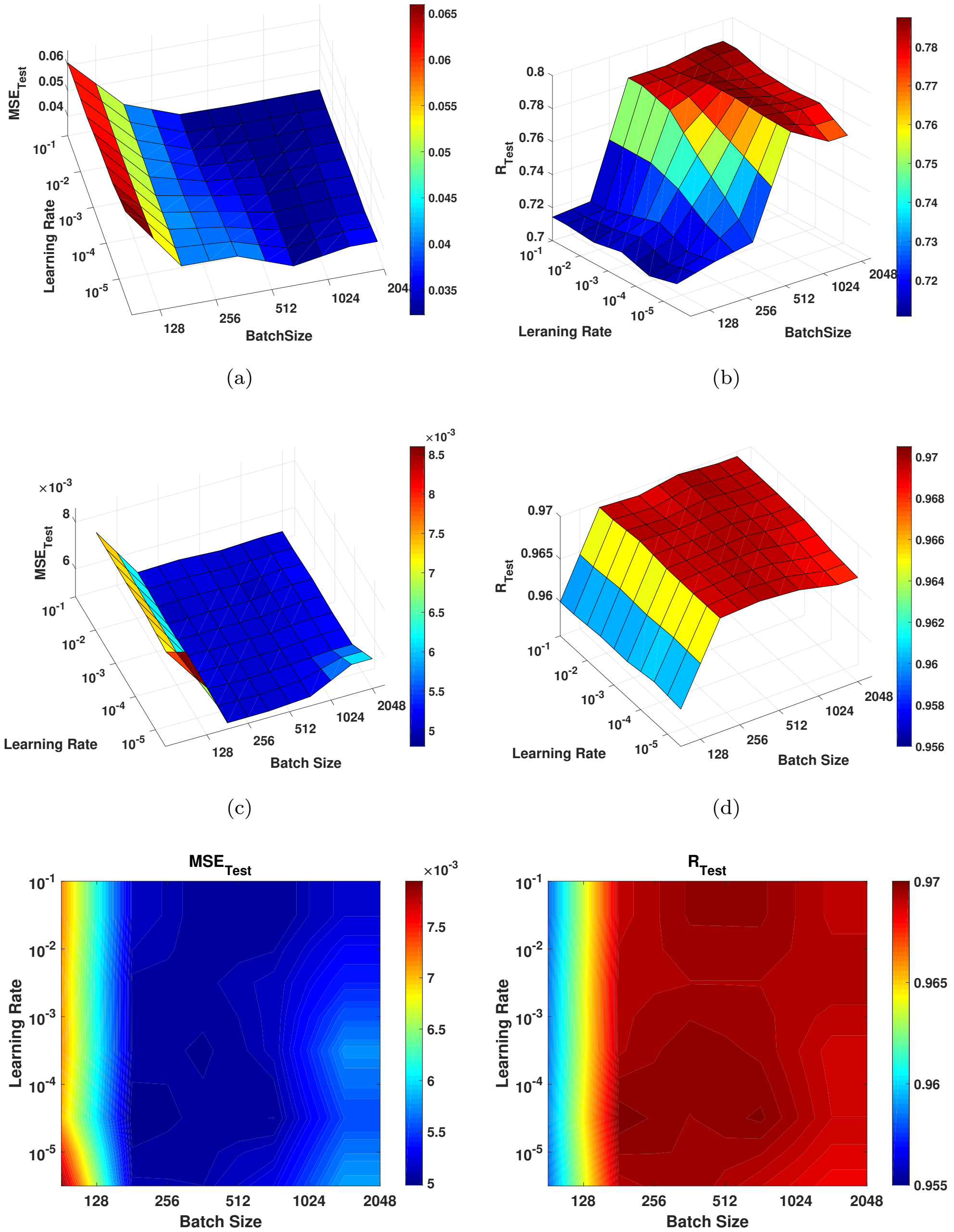
After removing the outliers, we compare the performance of the LSTM framework as a predictor of the power produced by the 6^{th} wind turbine in two models (model 1 with one input and model 3 with two inputs including wind speed and direction) with different ranges of the batch size and learning rate. Figure 11 illustrates the 3D forecasting landscape of the correlation between the batch size, learning rates and forecasting accuracy for model 1 (within ten-minute) and model 3 (one-hour interval) .

The applied method for tuning the hyper-parameters is the grid search in this experiment. In model 1, the best prediction results happen where the batch size is more than 1024 and the learning rate value is between 10^{-3} and 10^{-4}.

In order to highlight the benefit of using the outlier detection & removal technique (K-means + Autoencoder), a comprehensive comparison for four prediction models is applied (the time interval considered within these models is ten minutes), and the results of this experiment can be seen in Figure 12. Meanwhile, it is noticed that removing the noise from the SCADA data results in a significant enhancement in the accuracy of the prediction except two inappropriate configurations (Batch size=128 and Learning rate=10^{-3}, 10^{-1}).

The same experiment was completed in order to compare the performance of four models in forecasting the wind turbine power output in the one-hour ahead. The comparative outcomes are presented in Figure 13. In the first and second models, the highest accuracy was observed when batch sizes were large; however, the best-found configurations of the hyper-parameters in model 3 and 4 occurred where the batch size and learning rate of 256 and 10^{-4} respectively.

We performed a statistical analysis of the forecasting results above using a Friedman test (non-parametric and multiple comparisons). The statistical analysis results by implementing the Friedman test are shown in



(a)

(b)

(c)

(d)

(e)

(f)

Figure 9: Hyper-parameter tuning of the applied LSTM network for forecasting the short-term power output of the wind turbine without removing the outliers(Layer number=1, neuron number=100, Optimizer='Adam'). (a) the average of MSE test-set (ten-minute ahead) with one input (wind speed) (b) the average of R-value test-set (ten-minutes ahead) with one input (wind speed). (c) and (d) the average of MSE and R-value test-set with two inputs (wind speed, Power), respectively. (e) and (f)the average of MSE and R-value test-set with two inputs (wind speed, wind direction and Power), respectively

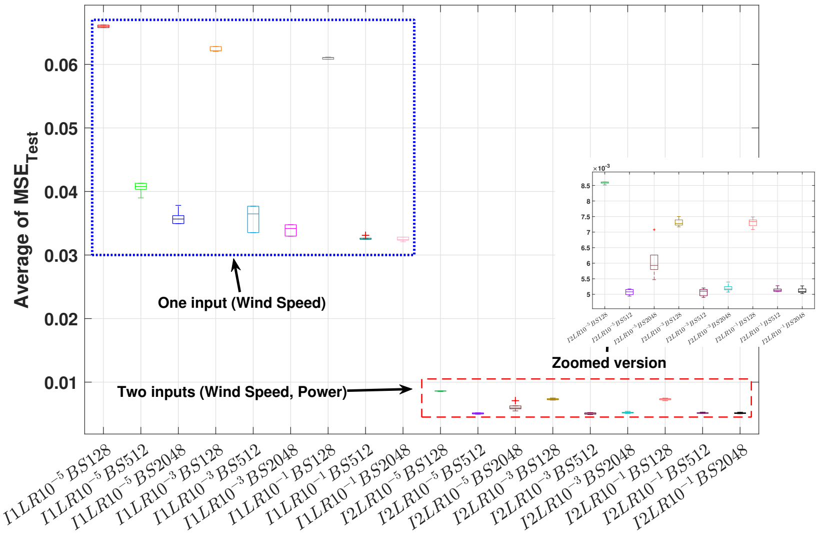


Figure 10: Comparison of the LSTM performance with one (I_1) and two inputs (I_2) without removing outliers

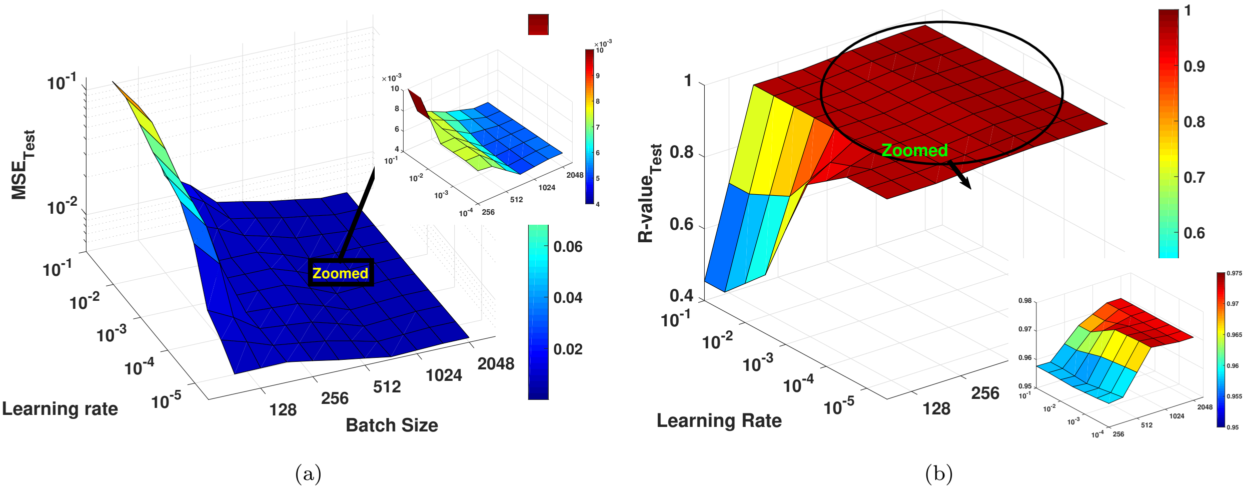
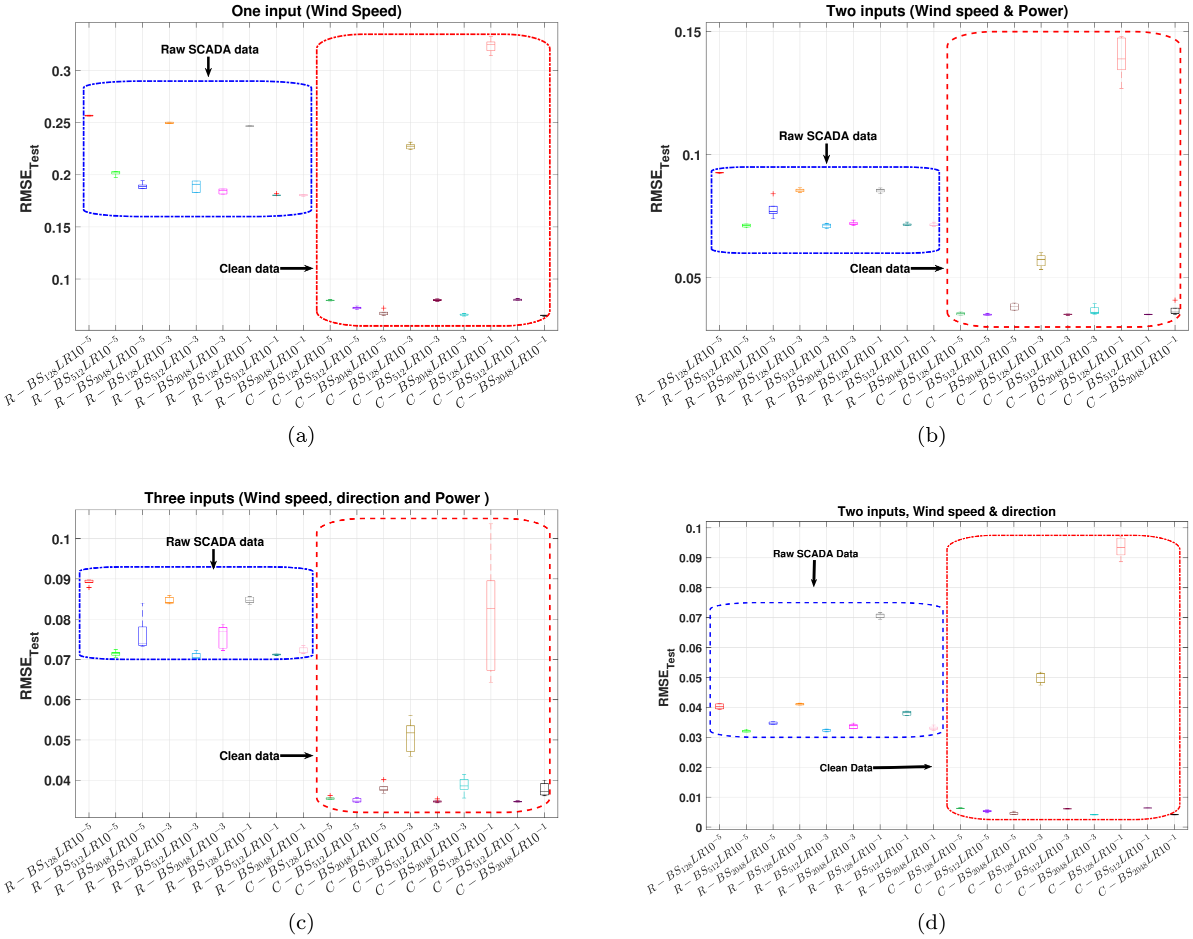


Figure 11: Hyper-parameter tuning of the applied LSTM network applied to the forecasting of the wind turbine's short-term power output after removing outliers(Layer number=1, neuron number=100, Optimizer='Adam'). (a) the average of MSE test-set (ten-minute ahead) with one input (wind speed) (b) the average of R-value test-set (one-hour ahead) with two inputs (wind speed and direction).



(a)

(b)

(c)

(d)

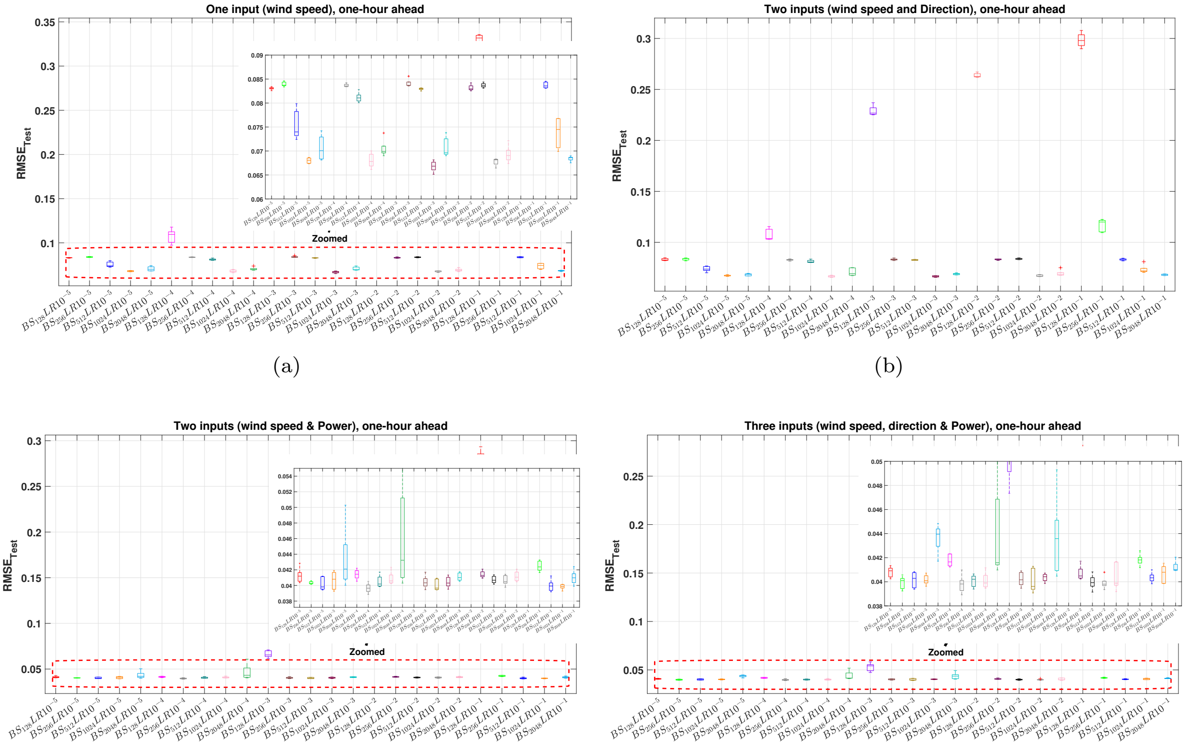
Figure 12: A comparison of the LSTM network Hyper-parameter tuning performance training on the raw SCADA data (R) and training after removing the outliers (C) for ten-minute ahead forecasting (Layer number=1, neuron number=100, Optimizer='Adam'). (a) the RMSE test-set with one input (wind speed). (b) the RMSE test-set with two inputs (wind speed and current power).

Figure 14. We ranked the four forecasting LSTM models corresponding to their mean value. As shown in Figure 14, Model 3 and 4 obtained the first and second ranking (i.e., the lowest value goes the first rank). This was in a comparison featuring all models over the 25 configurations of the hyper-parameters, under the ten-minute ahead forecasting conditions.

The overall comparison of four LSTM models at intervals of ten-minute and one-hour ahead can be seen in Figure 15. Each box represents the average RMSE forecasting of all 25 configurations per model. These statistical results show that the 4th model with three inputs, including wind speed, wind direction and the current power output of the wind turbine found a configuration with the minimum validation error. However, model 3 outperforms other models on average. According to the statistical results, model 3 with two inputs (wind speed and the current produced power) perform better than other models. Therefore, we then applied this model to developing hybrid neuro-evolutionary methods.

6.3. Experimental results of hybrid decomposition algorithm

According to the technical details of the adaptive VMD (AVMD) method proposed in Section 5.1.1, the standard VMD requires the initial number of decomposition mode (K) through the decomposition process.



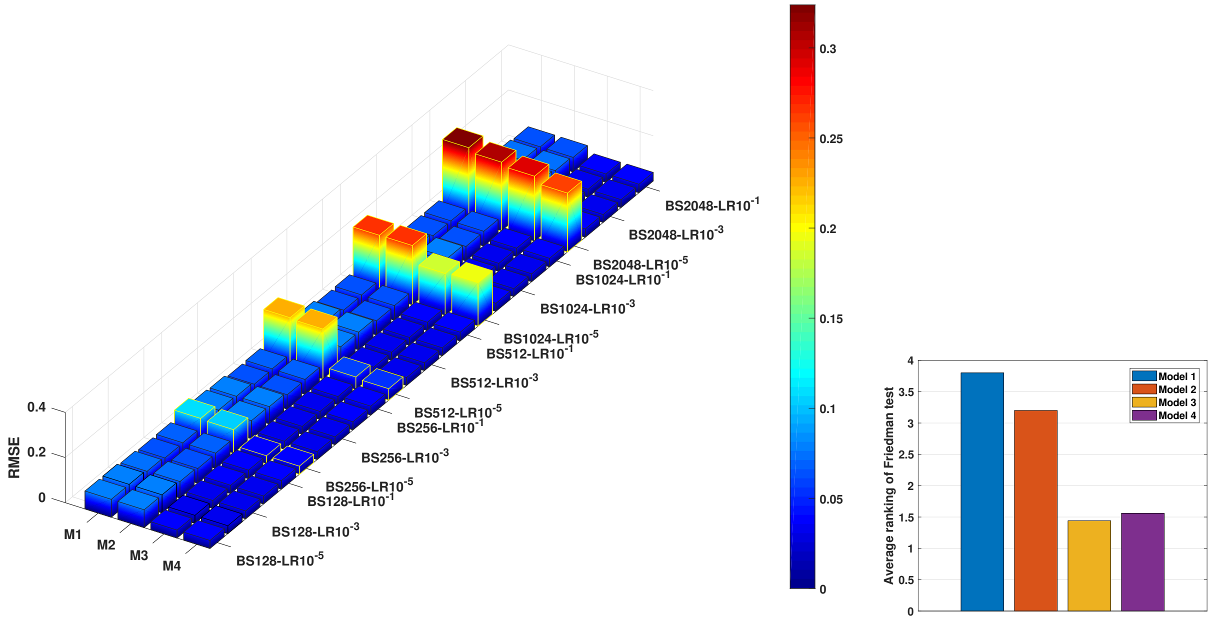
(a) (b)

(c) (d)

Figure 13: A comparison of four forecasting LSTM network models performance with various Hyper-parameters for forecasting the power output in one-hour ahead (Layer number=1, neuron number=100, Optimizer='Adam'). (a) the RMSE test-set with one input (wind speed). (b) the RMSE test-set with two inputs (wind speed and direction), (c) the RMSE test-set with two inputs (wind speed and current power), (d) the RMSE test-set with three inputs (wind speed, direction and current power).

This parameter has a great impact on decomposition performance. Furthermore, the value of the penalty factor (PF) plays a significant role in determining the reconstruction fidelity. A smaller PF value leads to achieving stricter data fidelity. The third important VMD argument is the update rate for the Lagrange multiplier (L_m). A higher degree conducts in faster convergence speed; however, it raises the probability of the optimization process converging to a local optimum. The last VMD hyper-parameter considered in this paper is the type of Method (M_I) applied in initialising the central frequencies. We consider three different methods including uniform random, grid and peaks [18]. In order to adjust the VMD hyper-parameters, we propose two fast and effective algorithms: Greedy Nelder-mead (GNM) search method and Adaptive randomised local search (ARLS). The Permutation Entropy is applied to evaluate the performance of both optimisers. Figure 18 shows the convergence rate of optimisers to find the best configuration of the VMD method, and we can see that GNM can overcome another method in terms of convergence rate and quality of proposed configuration on average. The best-found configuration values are $K = 5$, $PF = 1900$, $L_m = 1.34 \times 10^{-6}$, and $M_I = grid$.

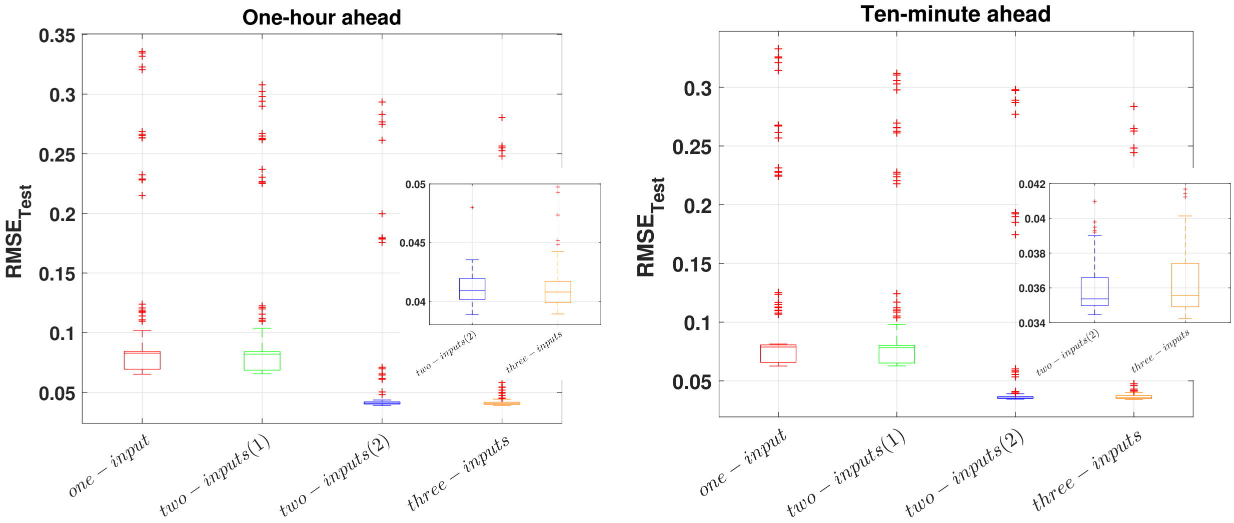
As shown in Figure 18, the developing IMFs decomposed using adaptive VMD of the case study in the field of wind speed from IMF_1 to IMF_5. It can be remarked that the number of IMF has been significantly diminished compared with the primary decomposed IMFs, due to significantly reducing the computational expenditure of the whole model. Furthermore, we can observe that the first IMF consists of the highest frequency components, and the last IMF holds more considerable magnitudes and characteristic biases of the original SCADA time-series data compared with the other $IMFs$.



(a)

(b)

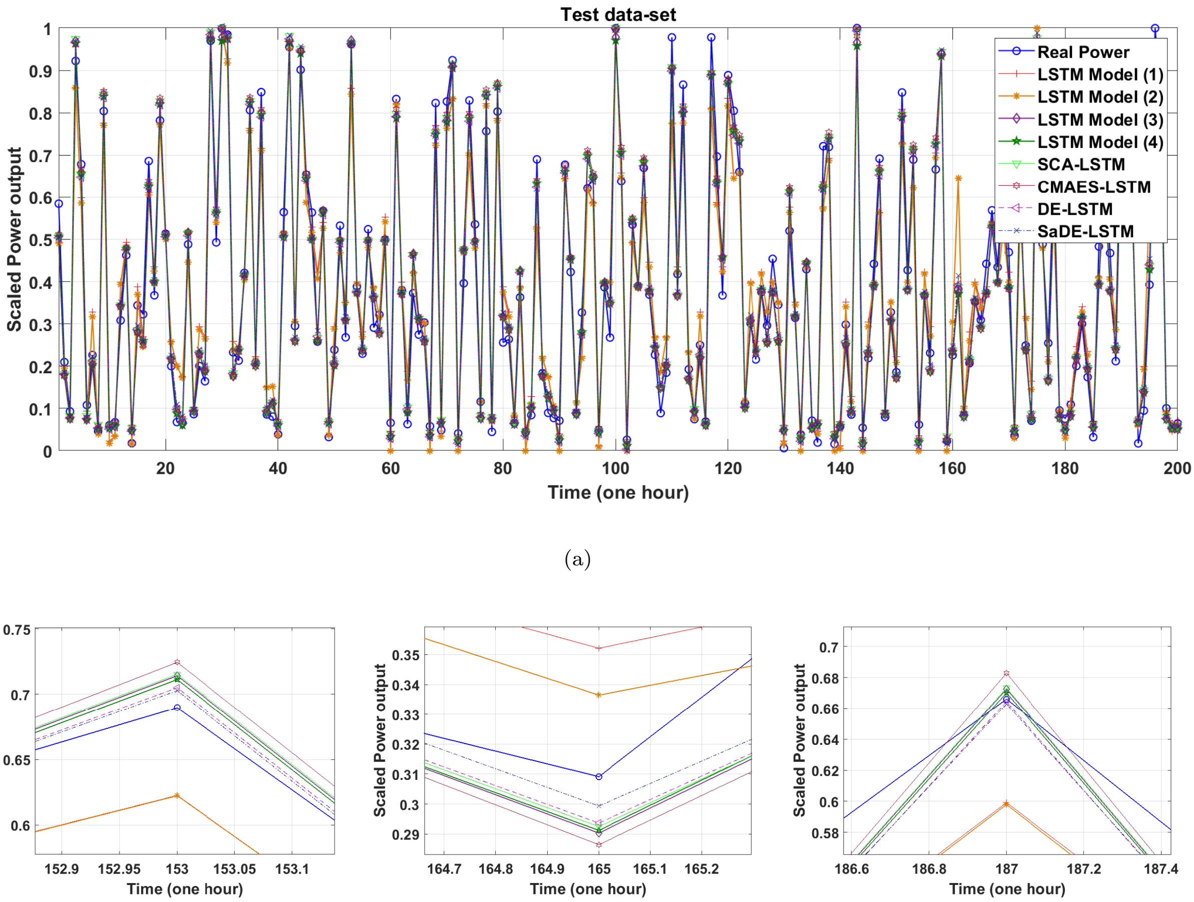
Figure 14: A comparison of four proposed forecasting models with 25 different configurations of hyper-parameters. (a) comparison of various LSTM settings based on RMSE. (b) Average ranking of the Friedman test for four applied models.



(a)

(b)

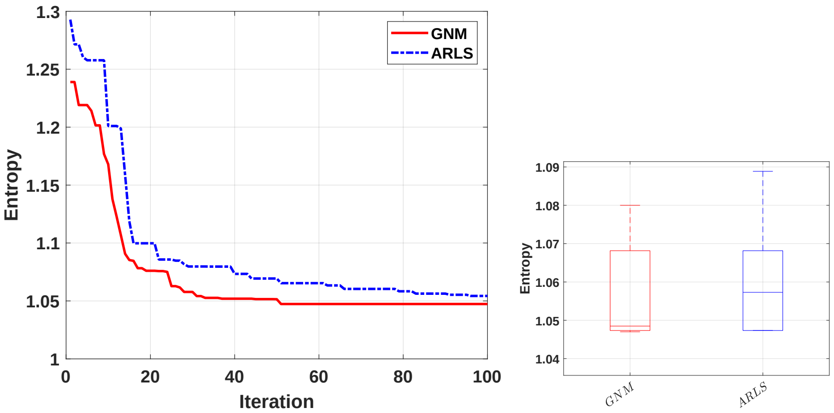
Figure 15: The total performance comparison of four LSTM forecasting models with ten-minute and one-hour ahead prediction. two-input(1) is the wind speed and direction, two-input(2) mentions the wind speed and current power of wind turbine.



(a)

(b) (c) (d)

Figure 16: The best estimated power output values from the proposed hybrid models and the corresponding measured values in SCADA system. The initial values of the weights are kept the same.



(a) (b)

Figure 17: A comparison of both optimisers GNM and ARLS applied to improve the performance of VMD by hyper-parameters tuning. a) convergence rate b) a summary of statistical optimisation results with box plot.

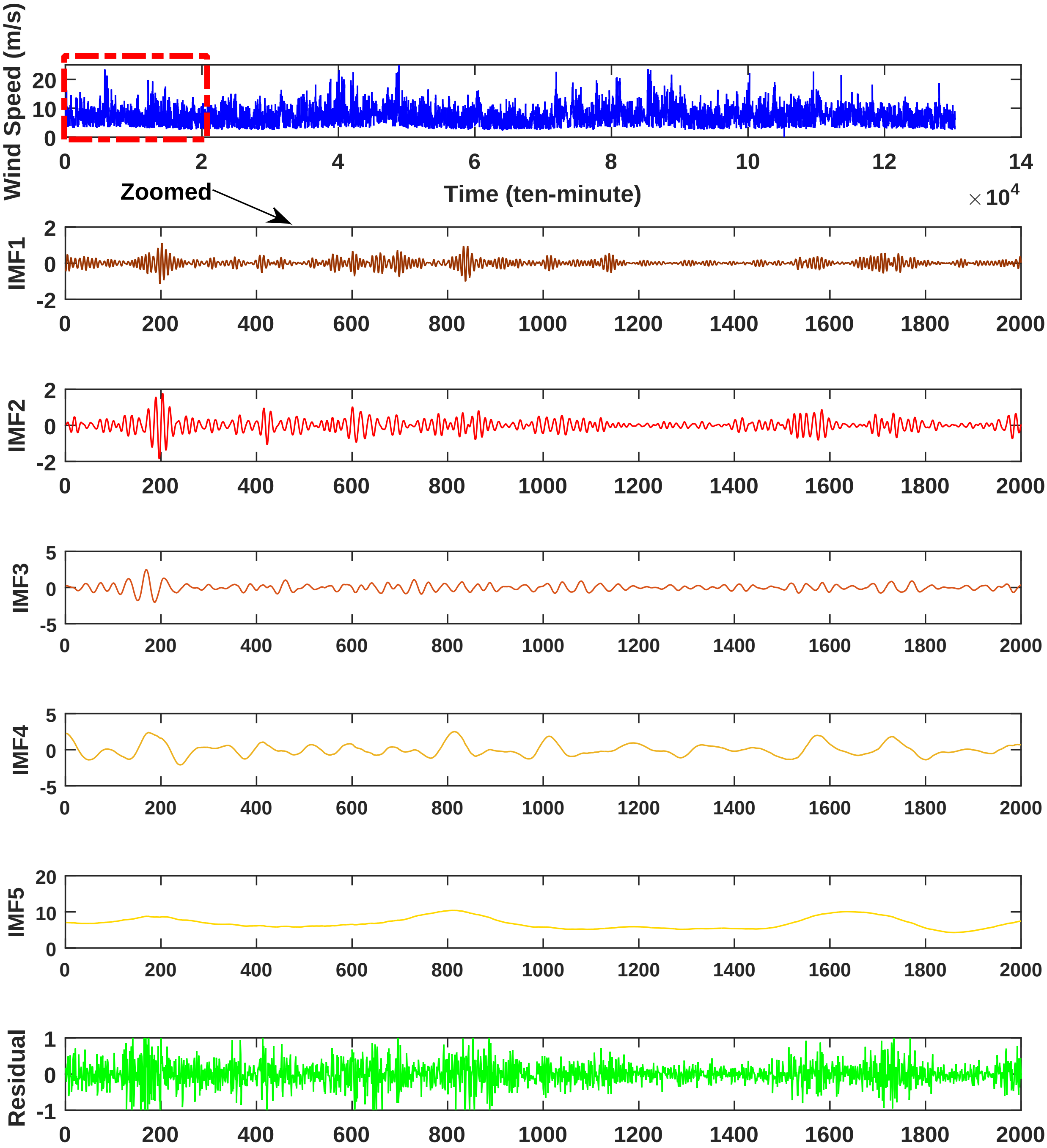


Figure 18: The decomposition results of wind speed obtained by proposed adaptive variational mode decomposition method.

6.4. Overall hybrid neuro-evolutionary model performance

Finally, the proposed hybrid model (AVMD-AOA-LSTM) is compared with some of the state-of-the-art forecasting frameworks. To evaluate the performance of the proposed hybrid model, we compare 13 different forecasting methods, including the best LSTM model which is tuned by the grid search, an adaptive neuro-fuzzy inference system (ANFIS) (its hyper-parameters are assigned based on the study in [65]), Feed-forward neural networks (FFNN), Polynomial neural networks (PNN), a developed LSTM called Bi-directional LSTM (Bi-LSTM), two existing hybrid neuro-evolutionary methods (CMAES-LSTM [54], and DE-LSTM [58], and also six new composed models including SaDE-LSTM, EFADE-LSTM, SCA-LSTM, EO-LSTM, MPA-LSTM, and AOA-LSTM. Table 4 and 5 summarise the outcomes of the performance indices produced to determine the optimal structure and hyper-parameters of the applied forecasters. It is evident that the AVMD-AOA-LSTM hybrid model outperforms other hybrid models and provides more accurate forecasting results.

In addition, The actual and forecasting values generated by the hybrid models and the LSTM networks for the one-hour ahead forecasting are shown in Figure 16. In the zoomed versions of Figure 16(b, c and d), it can be seen that a self-adaptive DE with LSTM estimates the power output with a considerable accuracy compared with other models.

Figure 19 and 20 show the statistical results of the RMSE performance indices applied in this work for short-term wind turbine power forecasting (ten-minute and one-hour ahead) received by 13 other models and the proposed AVMD-AOA-LSTM hybrid model. Concerning this experiment, the AVMD-AOA-LSTM hybrid evolutionary model can outperform its 13 competitors for short-term wind speed forecasting with the minimum RMSE value as well as the highest rate of R-value.

6.5. Training runtime analysis

In order to evaluate the computational complexity of the applied model in this paper, we trained the LSTM as the core of the proposed hybrid model for different ranges of learning rate and batch size for one epoch. Figure 21 shows (a) the LSTM runtime landscape with various values of both batch sizes and learning rates. It is noted that the size of the applied batch is meaningful and plays a significant role in the computational cost of training. Furthermore, this figure represents (b) the performance landscape of the LSTM when the number of epoch for training is just one. According to the observations, small batch sizes with a learning rate between 10^{-1} and 10^{-4} can achieve better forecasting results.

To compare the computational complexity of the proposed hybrid frameworks, the average runtime of one iteration for nine hybrid models are compared. It is noted that the population size of evolutionary algorithms is the same. Table 6 shows the average runtime of one iteration that consists of LSTM training runtime, the computational time for applying evolutionary operators like mutation and crossover and hyper-parameters tuning, and finally the required decomposition process runtime. In terms of runtime, the best performance is related to AVMD-AOA-LSTM due to having a decomposition technique and applying the parallel training setup. This parallel training setup is able to save considerably the computational runtime; however, it is involving some overheads of computation mainly because of data transfers, synchronisation, thread creation, and removal.

7. Conclusions

Due to multiple systems and meteorological factors, wind power time series data exhibit chaotic behaviours which are hard to predict. In this paper, a combination of autoencoder and clustering was adopted to reduce the stochastic noise inherent in raw data series. In the second phase, we propose a novel hybrid decomposition method consists of Variational Mode Decomposition (VMD) and two fast and effective search methods to optimise the hyper-parameters of VMD including the Greedy Nelder-mead (GNM) search method and Adaptive randomised local search (ARLS). Subsequently, a hybrid neuro-evolutionary approach (AVMD-AOA-LSTM) consisting of the self-adaptive differential evolution (SaDE), Sine Cosine Algorithm (SCA) and LSTM network was used for modelling the wind behavior. We then conducted extensive experiments and compared our proposed approach with ten alternative hybrid models. As our experiments

Table 4: Performance indices of forecasting wind turbine power output achieved by different models for ten-minutes ahead.

		MSE		RMSE		MAE		R	
Model		**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
FFNN	Mean	1.740E-03	1.733E-03	4.172E-02	4.163E-02	2.558E-02	2.554E-02	8.929E-01	8.931E-01
	Min	1.722E-03	1.677E-03	4.150E-02	4.095E-02	2.548E-02	2.535E-02	8.913E-01	8.901E-01
	Max	1.764E-03	1.775E-03	4.201E-02	4.213E-02	2.570E-02	2.572E-02	8.942E-01	8.968E-01
	Std	1.679E-05	3.894E-05	2.011E-04	4.686E-04	7.451E-05	1.346E-04	9.998E-04	2.339E-03
PNN	Mean	2.853E-03	2.860E-03	5.255E-02	5.263E-02	4.156E-02	4.159E-02	9.467E-01	9.466E-01
	Min	1.794E-03	1.873E-03	4.236E-02	4.328E-02	3.239E-02	3.270E-02	9.330E-01	9.330E-01
	Max	6.075E-03	6.058E-03	7.794E-02	7.783E-02	6.489E-02	6.483E-02	9.558E-01	9.553E-01
	Std	1.230E-03	1.217E-03	1.013E-02	1.001E-02	8.925E-03	8.872E-03	8.377E-03	8.286E-03
ANFIS	Mean	6.981E-03	6.949E-03	8.324E-02	8.343E-02	5.287E-02	5.282E-02	9.615E-01	9.618E-01
	Min	5.657E-03	5.725E-03	7.566E-02	7.521E-02	4.795E-02	4.809E-02	9.593E-01	9.584E-01
	Max	7.648E-03	7.673E-03	8.760E-02	8.745E-02	5.546E-02	5.544E-02	9.665E-01	9.664E-01
	Std	7.860E-04	7.614E-04	4.662E-03	4.834E-03	2.999E-03	2.979E-03	2.825E-03	2.858E-03
LSTM-grid	Mean	1.427E-03	1.407E-03	3.778E-02	3.751E-02	2.834E-02	2.819E-02	9.826E-01	9.829E-01
	Min	1.411E-03	1.382E-03	3.756E-02	3.718E-02	2.822E-02	2.804E-02	9.826E-01	9.826E-01
	Max	1.452E-03	1.450E-03	3.810E-02	3.808E-02	2.867E-02	2.836E-02	9.827E-01	9.830E-01
	Std	1.520E-04	2.552E-04	1.233E-02	1.598E-02	1.908E-03	1.216E-04	4.968E-05	1.606E-04
Bi-LSTM	Mean	1.559E-03	1.565E-03	3.949E-02	3.956E-02	3.039E-02	3.034E-02	9.762E-01	9.761E-01
	Min	1.551E-03	1.509E-03	3.939E-02	3.885E-02	3.034E-02	3.008E-02	9.730E-01	9.726E-01
	Max	1.568E-03	1.604E-03	3.960E-02	4.006E-02	3.044E-02	3.056E-02	9.786E-01	9.788E-01
	Std	5.094E-06	3.039E-05	6.450E-05	3.848E-04	3.355E-05	1.712E-04	1.884E-03	2.043E-03
CMAES-LSTM	Mean	1.236E-03	1.193E-03	3.515E-02	3.454E-02	2.643E-02	2.615E-02	9.926E-01	9.928E-01
	Min	1.227E-03	1.171E-03	3.503E-02	3.422E-02	2.631E-02	2.594E-02	9.926E-01	9.927E-01
	Max	1.256E-03	1.206E-03	3.544E-02	3.473E-02	2.679E-02	2.638E-02	9.927E-01	9.930E-01
	Std	7.280E-06	1.161E-05	1.033E-04	1.682E-04	1.206E-04	1.393E-04	2.597E-05	9.001E-05
DE-LSTM	Mean	1.241E-03	1.163E-03	3.522E-02	3.411E-02	2.648E-02	2.583E-02	9.926E-01	9.930E-01
	Min	1.235E-03	1.159E-03	3.515E-02	3.404E-02	2.639E-02	2.569E-02	9.925E-01	9.929E-01
	Max	1.248E-03	1.180E-03	3.533E-02	3.436E-02	2.667E-02	2.609E-02	9.926E-01	9.932E-01
	Std	4.002E-06	5.727E-06	5.680E-05	8.376E-05	7.153E-05	1.087E-04	2.693E-05	6.738E-05
SaDE-LSTM	Mean	1.238E-03	1.164E-03	3.519E-02	3.412E-02	2.642E-02	2.580E-02	9.926E-01	9.930E-01
	Min	1.233E-03	1.155E-03	3.511E-02	3.398E-02	2.634E-02	2.571E-02	9.925E-01	9.929E-01
	Max	1.247E-03	1.171E-03	3.531E-02	3.422E-02	2.652E-02	2.592E-02	9.926E-01	9.931E-01
	Std	4.227E-06	5.470E-06	6.003E-05	8.020E-05	6.600E-05	7.349E-05	2.258E-05	6.997E-05
EFADE-LSTM	Mean	1.238E-03	1.202E-03	3.519E-02	3.467E-02	2.648E-02	2.623E-02	9.926E-01	9.928E-01
	Min	1.224E-03	1.169E-03	3.499E-02	3.419E-02	2.629E-02	2.586E-02	9.924E-01	9.927E-01
	Max	1.273E-03	1.222E-03	3.568E-02	3.496E-02	2.693E-02	2.651E-02	9.926E-01	9.929E-01
	Std	1.502E-05	1.796E-05	2.126E-04	2.595E-04	1.942E-04	2.132E-04	8.773E-05	1.069E-04
SCA-LSTM	Mean	1.238E-03	1.163E-03	3.519E-02	3.411E-02	2.643E-02	2.581E-02	9.926E-01	9.930E-01
	Min	1.233E-03	1.155E-03	3.511E-02	3.398E-02	2.637E-02	2.572E-02	9.925E-01	9.929E-01
	Max	1.244E-03	1.172E-03	3.527E-02	3.423E-02	2.652E-02	2.591E-02	9.926E-01	9.931E-01
	Std	3.458E-06	5.903E-06	4.913E-05	8.657E-05	5.261E-05	6.799E-05	2.895E-05	6.926E-05
EO-LSTM	Mean	1.235E-03	1.166E-03	3.515E-02	3.414E-02	2.639E-02	2.582E-02	9.926E-01	9.930E-01
	Min	1.229E-03	1.151E-03	3.506E-02	3.392E-02	2.630E-02	2.564E-02	9.926E-01	9.929E-01
	Max	1.243E-03	1.182E-03	3.525E-02	3.438E-02	2.661E-02	2.606E-02	9.926E-01	9.931E-01
	Std	4.989E-06	8.645E-06	7.096E-05	1.266E-04	1.055E-04	1.213E-04	2.380E-05	6.940E-05
MPA-LSTM	Mean	1.237E-03	1.171E-03	3.517E-02	3.422E-02	2.643E-02	2.589E-02	9.926E-01	9.930E-01
	Min	1.231E-03	1.158E-03	3.509E-02	3.403E-02	2.634E-02	2.572E-02	9.925E-01	9.929E-01
	Max	1.245E-03	1.180E-03	3.529E-02	3.435E-02	2.654E-02	2.604E-02	9.926E-01	9.931E-01
	Std	4.816E-06	6.158E-06	6.842E-05	9.008E-05	6.868E-05	1.124E-04	2.426E-05	6.394E-05
AOA-LSTM	Mean	1.238E-03	1.162E-03	3.519E-02	3.409E-02	2.643E-02	2.581E-02	9.926E-01	9.930E-01
	Min	1.232E-03	1.156E-03	3.510E-02	3.399E-02	2.634E-02	2.572E-02	9.925E-01	9.929E-01
	Max	1.243E-03	1.168E-03	3.526E-02	3.417E-02	2.653E-02	2.590E-02	9.926E-01	9.931E-01
	Std	3.355E-06	5.164E-06	4.768E-05	7.576E-05	5.154E-05	5.537E-05	2.966E-05	8.395E-05
AVMD-AOA-LSTM	Mean	1.097E-03	1.103E-03	3.312E-02	3.322E-02	2.330E-02	2.337E-02	9.935E-01	9.935E-01
	Min	1.078E-03	1.091E-03	3.283E-02	3.303E-02	2.309E-02	2.310E-02	9.934E-01	9.934E-01
	Max	1.138E-03	1.134E-03	3.373E-02	3.367E-02	2.370E-02	2.387E-02	9.935E-01	9.935E-01
	Std	2.559E-05	1.734E-05	3.846E-04	2.598E-04	2.721E-04	3.074E-04	7.234E-05	4.087E-05

Table 5: Performance indices of forecasting wind turbine power output achieved by different models for one-hour ahead.

		MSE		RMSE		MAE		R	
Model		**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
FFNN	Mean	1.973E-03	1.960E-03	4.442E-02	4.427E-02	2.723E-02	2.717E-02	8.777E-01	8.786E-01
	Min	1.959E-03	1.916E-03	4.426E-02	4.377E-02	2.710E-02	2.696E-02	8.766E-01	8.764E-01
	Max	1.992E-03	1.993E-03	4.463E-02	4.465E-02	2.731E-02	2.732E-02	8.786E-01	8.810E-01
	Std	1.192E-05	2.775E-05	1.341E-04	3.138E-04	6.541E-05	1.119E-04	6.441E-04	1.486E-03
PNN	Mean	3.551E-03	3.560E-03	5.854E-02	5.859E-02	4.667E-02	4.676E-02	9.379E-01	9.379E-01
	Min	1.879E-03	1.900E-03	4.335E-02	4.359E-02	3.316E-02	3.326E-02	9.294E-01	9.295E-01
	Max	7.702E-03	7.745E-03	8.776E-02	8.801E-02	7.642E-02	7.685E-02	9.589E-01	9.589E-01
	Std	1.586E-03	1.605E-03	1.176E-02	1.188E-02	1.132E-02	1.143E-02	8.900E-03	8.871E-03
ANFIS	Mean	6.875E-03	6.789E-03	8.284E-02	8.229E-02	5.319E-02	5.289E-02	9.608E-01	9.610E-01
	Min	6.208E-03	6.013E-03	7.879E-02	7.754E-02	5.071E-02	4.986E-02	9.585E-01	9.575E-01
	Max	7.617E-03	7.905E-03	8.728E-02	8.891E-02	5.620E-02	5.711E-02	9.634E-01	9.636E-01
	Std	6.226E-04	7.344E-04	3.758E-03	4.421E-03	2.348E-03	2.851E-03	2.318E-03	2.314E-03
LSTM-grid	Mean	1.906E-03	1.909E-03	4.366E-02	3.966E-02	3.046E-02	3.053E-02	9.903E-01	9.905E-01
	Min	1.880E-03	1.827E-03	4.336E-02	3.886E-02	3.036E-02	3.024E-02	9.902E-01	9.899E-01
	Max	1.923E-03	2.004E-03	4.386E-02	4.051E-02	3.055E-02	3.083E-02	9.905E-01	9.909E-01
	Std	1.968E-05	8.082E-05	2.458E-04	5.861E-04	6.857E-05	2.415E-04	1.238E-04	4.684E-04
Bi-LSTM	Mean	1.773E-03	1.788E-03	4.211E-02	4.228E-02	3.230E-02	3.222E-02	9.754E-01	9.752E-01
	Min	1.750E-03	1.720E-03	4.183E-02	4.147E-02	3.207E-02	3.212E-02	9.713E-01	9.710E-01
	Max	1.790E-03	1.860E-03	4.231E-02	4.313E-02	3.259E-02	3.231E-02	9.792E-01	9.790E-01
	Std	1.146E-05	4.012E-05	1.362E-04	4.744E-04	1.990E-04	7.218E-05	2.852E-03	2.923E-03
CMAES-LSTM	Mean	1.634E-03	1.547E-03	4.042E-02	3.933E-02	3.061E-02	3.033E-02	9.902E-01	9.907E-01
	Min	1.608E-03	1.520E-03	4.010E-02	3.898E-02	3.041E-02	3.000E-02	9.901E-01	9.905E-01
	Max	1.652E-03	1.589E-03	4.065E-02	3.987E-02	3.084E-02	3.066E-02	9.903E-01	9.910E-01
	Std	1.208E-05	2.141E-05	1.495E-04	2.717E-04	1.180E-04	1.854E-04	6.810E-05	1.496E-04
DE-LSTM	Mean	1.645E-03	1.483E-03	4.056E-02	3.851E-02	3.064E-02	3.005E-02	9.901E-01	9.911E-01
	Min	1.636E-03	1.467E-03	4.045E-02	3.830E-02	3.050E-02	2.963E-02	9.901E-01	9.909E-01
	Max	1.655E-03	1.519E-03	4.068E-02	3.897E-02	3.085E-02	3.026E-02	9.902E-01	9.913E-01
	Std	6.622E-06	1.504E-05	8.163E-05	1.948E-04	1.043E-04	1.679E-04	3.803E-05	1.328E-04
SaDE-LSTM	Mean	1.641E-03	1.488E-03	4.050E-02	3.858E-02	3.063E-02	2.998E-02	9.901E-01	9.911E-01
	Min	1.627E-03	1.469E-03	4.034E-02	3.833E-02	3.049E-02	2.980E-02	9.900E-01	9.909E-01
	Max	1.658E-03	1.498E-03	4.072E-02	3.870E-02	3.083E-02	3.014E-02	9.902E-01	9.913E-01
	Std	1.017E-05	9.844E-06	1.255E-04	1.277E-04	1.066E-04	1.217E-04	7.218E-05	1.095E-04
EFADE-LSTM	Mean	1.637E-03	1.544E-03	4.047E-02	3.929E-02	3.062E-02	3.036E-02	9.902E-01	9.907E-01
	Min	1.612E-03	1.482E-03	4.015E-02	3.850E-02	3.044E-02	2.998E-02	9.899E-01	9.903E-01
	Max	1.686E-03	1.603E-03	4.106E-02	4.003E-02	3.110E-02	3.072E-02	9.903E-01	9.911E-01
	Std	2.001E-05	4.337E-05	2.463E-04	5.527E-04	1.950E-04	2.380E-04	1.216E-04	2.669E-04
SCA-LSTM	Mean	1.641E-03	1.483E-03	4.051E-02	3.850E-02	3.061E-02	2.997E-02	9.902E-01	9.911E-01
	Min	1.630E-03	1.468E-03	4.037E-02	3.831E-02	3.052E-02	2.979E-02	9.901E-01	9.909E-01
	Max	1.661E-03	1.499E-03	4.075E-02	3.872E-02	3.073E-02	3.025E-02	9.902E-01	9.913E-01
	Std	8.274E-06	9.258E-06	1.020E-04	1.203E-04	6.092E-05	1.764E-04	4.033E-05	1.329E-04
EO-LSTM	Mean	1.645E-03	1.494E-03	4.056E-02	3.865E-02	3.066E-02	3.003E-02	9.901E-01	9.910E-01
	Min	1.633E-03	1.478E-03	4.041E-02	3.844E-02	3.054E-02	2.985E-02	9.900E-01	9.908E-01
	Max	1.663E-03	1.523E-03	4.078E-02	3.902E-02	3.082E-02	3.016E-02	9.902E-01	9.911E-01
	Std	1.003E-05	1.319E-05	1.235E-04	1.702E-04	9.821E-05	1.031E-04	6.182E-05	9.960E-05
MPA-LSTM	Mean	1.632E-03	1.506E-03	4.040E-02	3.880E-02	3.053E-02	3.012E-02	9.902E-01	9.909E-01
	Min	1.619E-03	1.482E-03	4.024E-02	3.849E-02	3.044E-02	2.990E-02	9.901E-01	9.906E-01
	Max	1.642E-03	1.523E-03	4.052E-02	3.903E-02	3.060E-02	3.030E-02	9.903E-01	9.911E-01
	Std	7.413E-06	1.087E-05	9.179E-05	1.403E-04	5.915E-05	1.262E-04	5.386E-05	1.370E-04
AOA-LSTM	Mean	1.640E-03	1.481E-03	4.050E-02	3.848E-02	3.058E-02	2.997E-02	9.901E-01	9.911E-01
	Min	1.635E-03	1.471E-03	4.044E-02	3.835E-02	3.054E-02	2.973E-02	9.901E-01	9.910E-01
	Max	1.649E-03	1.488E-03	4.061E-02	3.857E-02	3.069E-02	3.006E-02	9.902E-01	9.912E-01
	Std	3.841E-06	5.860E-06	4.737E-05	7.616E-05	5.271E-05	1.067E-04	2.825E-05	6.077E-05
AVMD-AOA-LSTM	Mean	1.467E-03	1.447E-03	3.830E-02	3.804E-02	2.723E-02	2.722E-02	9.913E-01	9.914E-01
	Min	1.442E-03	1.405E-03	3.797E-02	3.748E-02	2.703E-02	2.690E-02	9.911E-01	9.911E-01
	Max	1.525E-03	1.504E-03	3.906E-02	3.878E-02	2.770E-02	2.755E-02	9.914E-01	9.917E-01
	Std	3.362E-05	3.719E-05	4.361E-04	4.874E-04	2.713E-04	2.791E-04	1.241E-04	2.308E-04

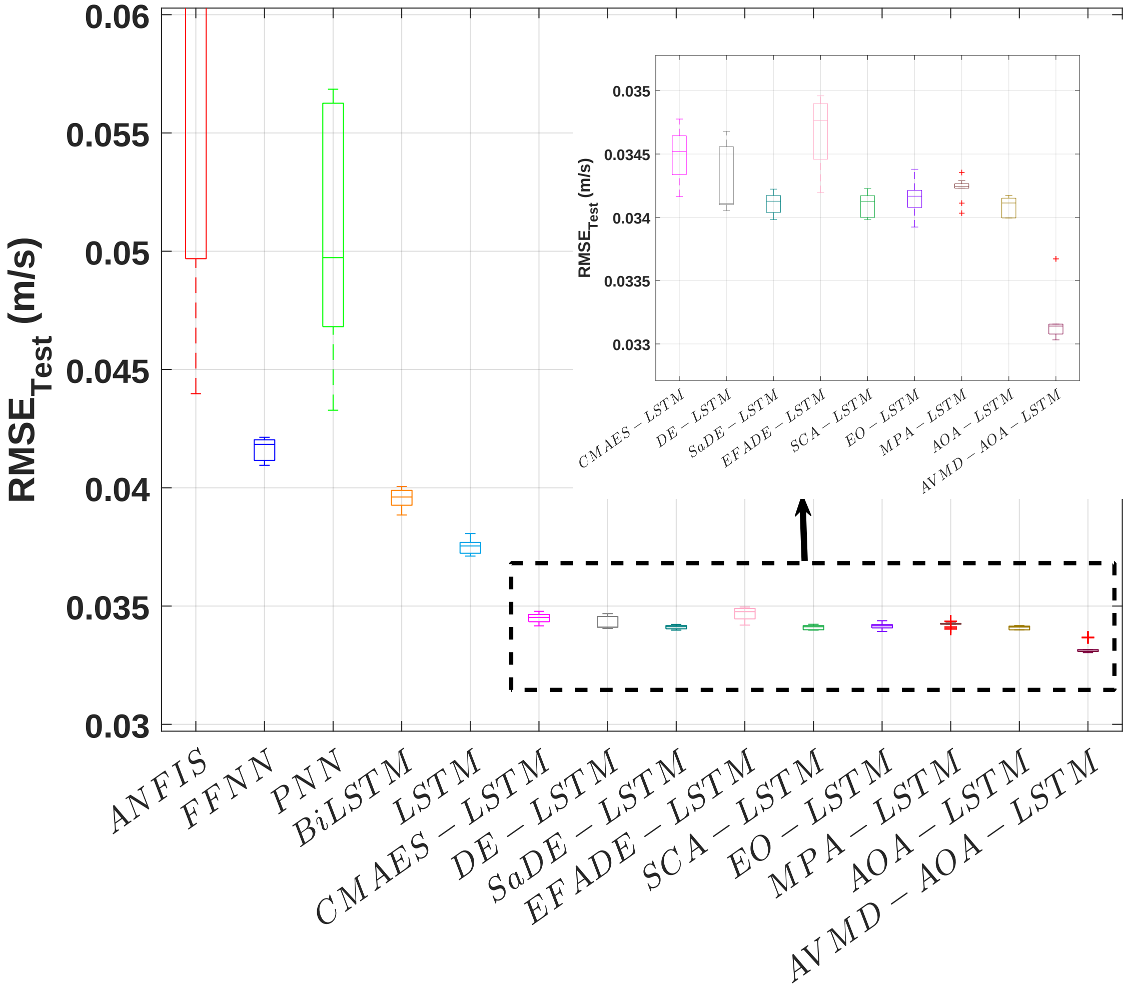


Figure 19: The performance indices comparison of the all proposed forecasting models from the case study in terms of the best achievement per each experiment. With regard to the median performance, AVMD-AOA-LSTM can overwhelm other hybrid forecasting models.forecasting results with regard to ten-minute ahead

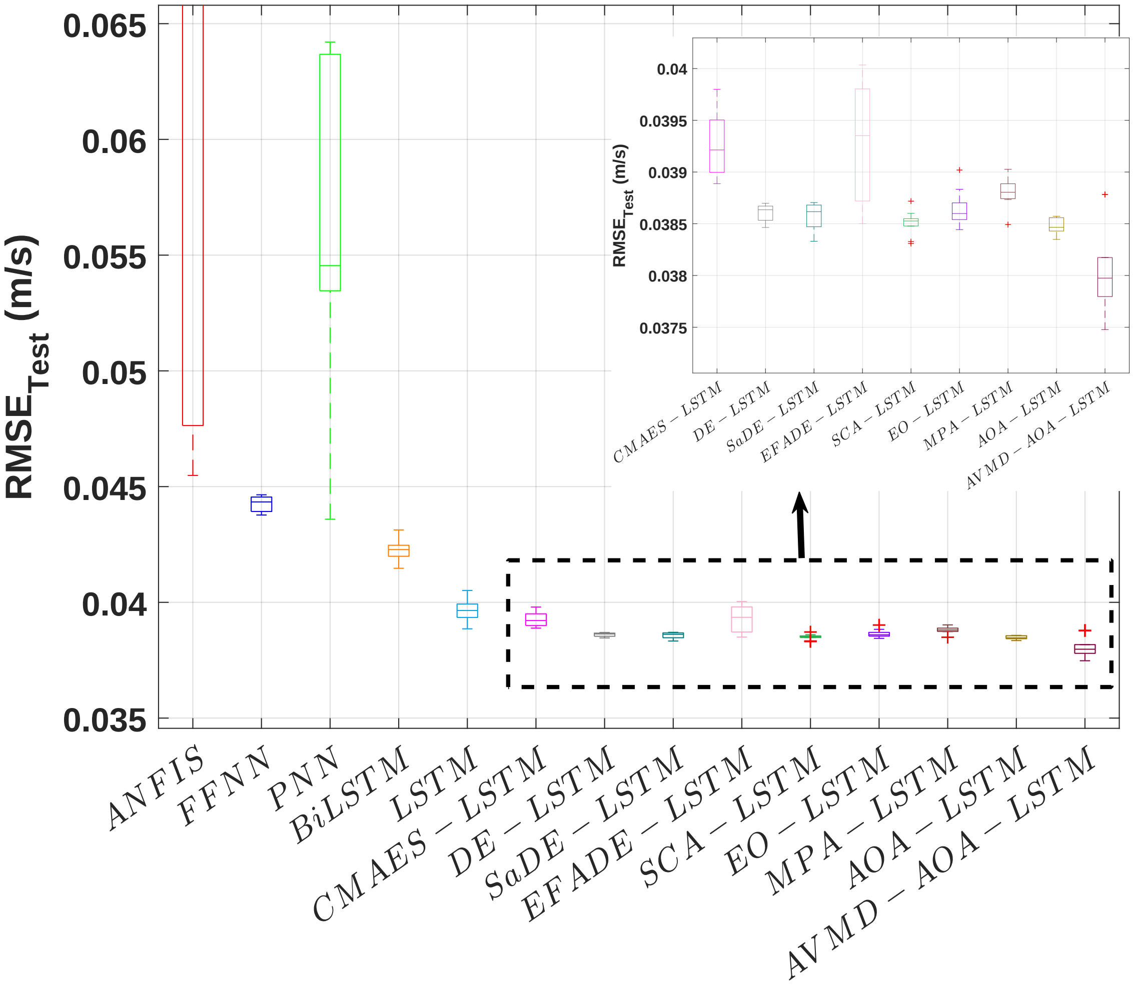
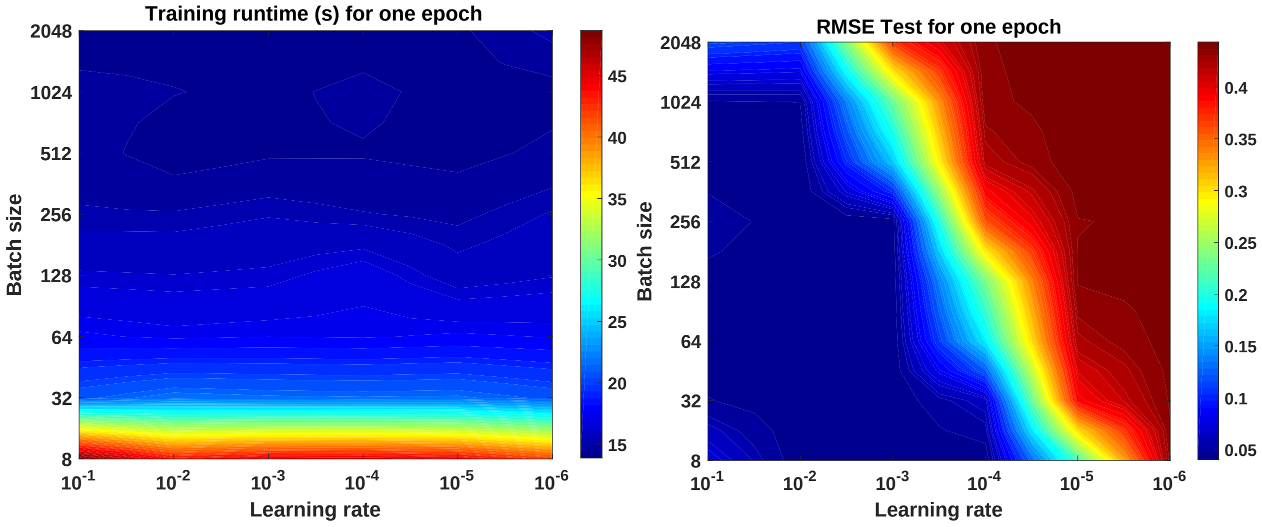


Figure 20: The comparison of the proposed forecasting models from the case study in terms of the best achievement per each experiment. AVMD-AOA-LSTM performs best compared with other models. forecasting results with regard to one-hour ahead



(a) (b)

Figure 21: Runtime analysis of the LSTM training for one epoch with different range of learning rates and batch sizes. a) training runtime analysis b) RMSE test of LSTM

Table 6: Computational complexity analysis of the applied hybrid forecasting models for one iteration

Methods	**Computational time (s)**
CMAES-LSTM	4.633E+03
DE-LSTM	5.108E+03
SaDE-LSTM	4.970E+03
EFADE-LSTM	4.743E+03
SCA-LSTM	4.672E+03
EO-LSTM	4.716E+03
MPA-LSTM	4.707E+03
AOA-LSTM	4.733E+03
AVMD-AOA-LSTM	3.233E+03

suggest, the proposed hybrid model outperforms its counterparts in terms of four performance criteria, in both ten-minute and one-hour intervals.

In the future, a greater variety of power curve datasets derived from different kinds of wind turbines from different regions will be evaluated to further enhance our model. Ultimately, another investigation of this study is to employ various outlier detection with decomposition and optimization approaches to improve the forecasting results.

References

[1] Abdel-Basset, M., Mohamed, R., Mirjalili, S., Chakrabortty, R.K., Ryan, M.J.: Solar photovoltaic parameter estimation using an improved equilibrium optimizer. Solar Energy **209**, 694–708 (2020)

[2] Adams, R.P.: K-means clustering and related algorithms. Princeton University (2018)

[3] Aghajani, G., Ghadimi, N.: Multi-objective energy management in a micro-grid. Energy Reports **4**, 218–225 (2018)

[4] An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE **2**(1) (2015)

[5] Azami, H., Escudero, J.: Improved multiscale permutation entropy for biomedical signal analysis: Interpretation and application to electroencephalogram recordings. Biomedical Signal Processing and Control **23**, 28–41 (2016)

[6] Bandt, C., Pompe, B.: Permutation entropy: A natural complexity measure for time series. Physical review letters **88**, 174102 (05 2002). https://doi.org/10.1103/PhysRevLett.88.174102

[7] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)

[8] Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Information Sciences **191**, 192–213 (2012)

[9] Bhaskar, K., Singh, S.: Awnn-assisted wind power forecasting using feed-forward neural network. IEEE transactions on sustainable energy **3**(2), 306–315 (2012)

[10] Chang, K.H.: Stochastic nelder–mead simplex method–a new globally convergent direct search method for simulation optimization. European journal of operational research **220**(3), 684–694 (2012)

[11] Chen, J., Zeng, G.Q., Zhou, W., Du, W., Lu, K.D.: Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Conversion and Management **165**, 681–695 (2018)

[12] Chen, M.R., Zeng, G.Q., Lu, K.D., Weng, J.: A two-layer nonlinear combination method for short-term wind speed prediction based on elm, enn, and lstm. IEEE Internet of Things Journal **6**(4), 6997–7010 (2019)

[13] Chen, W., Zhuang, J., Yu, W., Wang, Z.: Measuring complexity using fuzzyen, apen, and sampen. Medical engineering & physics **31**(1), 61–68 (2009)

[14] Chen, X., Yang, Y., Cui, Z., Shen, J.: Vibration fault diagnosis of wind turbines based on variational mode decomposition and energy entropy. Energy **174**, 1100–1109 (2019)

[15] Cui, Y., Bangalore, P., Tjernberg, L.B.: An anomaly detection approach based on machine learning and scada data for condition monitoring of wind turbines. In: 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS). pp. 1–6. IEEE (2018)

[16] Dasgupta, K., Roy, P.K., Mukherjee, V.: Power flow based hydro-thermal-wind scheduling of hybrid power system using sine cosine algorithm. Electric Power Systems Research **178**, 106018 (2020)

[17] Dorotić, H., Pukšec, T., Duić, N.: Multi-objective optimization of district heating and cooling systems for a one-year time horizon. Energy **169**, 319–328 (2019)

[18] Dragomiretskiy, K., Zosso, D.: Variational mode decomposition. IEEE transactions on signal processing **62**(3), 531–544 (2013)

[19] Duan, J., Wang, P., Ma, W., Tian, X., Fang, S., Cheng, Y., Chang, Y., Liu, H.: Short-term wind power forecasting using the hybrid model of improved variational mode decomposition and correntropy long short-term memory neural network. Energy **214**, 118980 (2021)

[20] Fadlallah, B., Chen, B., Keil, A., Príncipe, J.: Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information. Physical Review E **87**(2), 022911 (2013)

[21] Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H.: Marine predators algorithm: A nature-inspired metaheuristic. Expert Systems with Applications p. 113377 (2020)

[22] Feng, Z.k., Niu, W.j., Liu, S., Luo, B., Miao, S.m., Liu, K.: Multiple hydropower reservoirs operation optimization by adaptive mutation sine cosine algorithm based on neighborhood search and simplex search strategies. Journal of Hydrology **590**, 125223 (2020)

[23] Fu, W., Wang, K., Tan, J., Zhang, K.: A composite framework coupling multiple feature selection, compound prediction models and novel hybrid swarm optimizer-based synchronization optimization strategy for multi-step ahead short-term wind speed forecasting. Energy Conversion and Management **205**, 112461 (2020)

[24] Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems. pp. 1019–1027 (2016)

[25] Hamian, M., Darvishan, A., Hosseinzadeh, M., Lariche, M.J., Ghadimi, N., Nouri, A.: A framework to expedite joint energy-reserve payment cost minimization using a custom-designed method based on mixed integer genetic algorithm. Engineering Applications of Artificial Intelligence **72**, 203–212 (2018)

[26] Hansen, N., Kern, S.: Evaluating the cma evolution strategy on multimodal test functions. In: International Conference on Parallel Problem Solving from Nature. pp. 282–291. Springer (2004)

[27] He, Y., Li, H.: Probability density forecasting of wind power using quantile regression neural network and kernel density estimation. Energy conversion and management **164**, 374–384 (2018)

[28] Hu, Y.L., Chen, L.: A nonlinear hybrid wind speed forecasting model using lstm network, hysteretic elm and differential evolution algorithm. Energy conversion and management **173**, 123–142 (2018)

[29] Huang, Q., Cui, Y., Bertling Tjernberg, L., Bangalore, P.: Wind turbine health assessment framework based on power analysis using machine learning method. pp. 1–5 (09 2019). https://doi.org/10.1109/ISGTEurope.2019.8905495

[30] Huang, Q., Cui, Y., Tjernberg, L.B., Bangalore, P.: Wind turbine health assessment framework based on power analysis using machine learning method. In: 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe). pp. 1–5. IEEE

[31] Humeau-Heurtier, A., Wu, C.W., Wu, S.D.: Refined composite multiscale permutation entropy to overcome multiscale permutation entropy length dependence. IEEE signal processing letters **22**(12), 2364–2367 (2015)

[32] Jägersküpper, J.: Rigorous runtime analysis of the (1+1) es: 1/5-rule and ellipsoidal fitness landscapes. In: Wright, A.H., Vose, M.D., De Jong, K.A., Schmitt, L.M. (eds.) Foundations of Genetic Algorithms. pp. 260–281. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)

[33] Jiang, M.F., Tseng, S.S., Su, C.M.: Two-phase clustering process for outliers detection. Pattern recognition letters **22**(6-7), 691–700 (2001)

[34] Kavasseri, R.G., Seetharaman, K.: Day-ahead wind speed forecasting using f-arima models. Renewable Energy **34**(5), 1388–1393 (2009)

[35] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[36] Komusanac, I., Fraile, D., Brindley, G.: Wind energy in europe in 2018, trends and statistics. Wind Europe (2019)

[37] Lange, M., Focken, U.: Physical approach to short-term wind power prediction, vol. 208. Springer (2006)

[38] Leen, T.K., Orr, G.B.: Optimal stochastic search and adaptive momentum. In: Advances in neural information processing systems. pp. 477–484 (1994)

[39] Leng, H., Li, X., Zhu, J., Tang, H., Zhang, Z., Ghadimi, N.: A new wind power prediction method based on ridgelet transforms, hybrid feature selection and closed-loop forecasting. Advanced Engineering Informatics **36**, 20–30 (2018)

[40] Li, X., Yang, X., Huang, T.: Persistence of delayed cooperative models: Impulsive control method. Applied Mathematics and Computation **342**, 130–146 (2019)

[41] Lin, Z., Liu, X., Collu, M.: Wind power prediction based on high-frequency scada data along with isolation forest and deep learning neural networks. International Journal of Electrical Power & Energy Systems **118**, 105835 (2020)

[42] Liu, H., Mi, X., Li, Y.: Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, lstm network and elm. Energy Conversion and Management **159**, 54–64 (2018)

[43] Liu, Y., Wang, W., Ghadimi, N.: Electricity load forecasting by an improved forecast engine for building level consumers. Energy **139**, 18–30 (2017)

[44] López, E., Valle, C., Allende, H., Gil, E., Madsen, H.: Wind power forecasting based on echo state networks and long short-term memory. Energies **11**(3), 526 (2018)

[45] Lydia, M., Kumar, S.S., Selvakumar, A.I., Kumar, G.E.P.: A comprehensive review on wind turbine power curve modeling techniques. Renewable and Sustainable Energy Reviews **30**, 452–460 (2014)

[46] Majidi Nezhad, M., Shaik, R., Heydari, A., Razmjoo, A., Arslan, N., Astiaso Garcia, D.: A swot analysis for offshore wind energy assessment using remote-sensing potential. Applied Sciences **10**, 6398 (09 2020). https://doi.org/10.3390/app10186398

[47] Malhotra, P., Vig, L., Shroff, G., Agarwal, P.: Long short term memory networks for anomaly detection in time series. In: Proceedings. vol. 89, pp. 89–94. Presses universitaires de Louvain (2015)

[48] Mirjalili, S.M., Mirjalili, S.Z., Saremi, S., Mirjalili, S.: Sine cosine algorithm: theory, literature review, and application in designing bend photonic crystal waveguides. In: Nature-inspired optimizers, pp. 201–217. Springer (2020)

[49] Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. Knowledge-based systems **96**, 120–133 (2016)

[50] Mirzapour, F., Lakzaei, M., Varamini, G., Teimourian, M., Ghadimi, N.: A new prediction model of battery and wind-solar output in hybrid power system. Journal of Ambient Intelligence and Humanized Computing **10**(1), 77–87 (2019)

[51] Moeini, H., Torab, F.M.: Comparing compositional multivariate outliers with autoencoder networks in anomaly detection at hamich exploration area, east of iran. Journal of Geochemical Exploration **180**, 15–23 (2017)

[52] Mohamed, A.W., Suganthan, P.N.: Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. Soft Computing **22**(10), 3215–3235 (2018)

[53] Morshedizadeh, M., Kordestani, M., Carriveau, R., Ting, D.S., Saif, M.: Improved power curve monitoring of wind turbines. Wind engineering **41**(4), 260–271 (2017)

[54] Neshat, M., Nezhad, M.M., Abbasnejad, E., Tjernberg, L.B., Garcia, D.A., Alexander, B., Wagner, M.: An evolutionary deep learning method for short-term wind speed prediction: A case study of the lillgrund offshore wind farm. arXiv preprint arXiv:2002.09106 (2020)

[55] Nielson, J., Bhaganagar, K., Meka, R., Alaeddini, A.: Using atmospheric inputs for artificial neural networks to improve wind turbine power prediction. Energy **190**, 116273 (2020)

[56] Nikolić, V., Mitić, V.V., Kocić, L., Petković, D.: Wind speed parameters sensitivity analysis based on fractals and neuro-fuzzy selection technique. Knowledge and Information Systems **52**(1), 255–265 (2017)

[57] Olaofe, Z., Folly, K.: Wind power estimation using recurrent neural network technique. In: IEEE Power and Energy Society Conference and Exposition in Africa: Intelligent Grid Integration of Renewable Energy Resources (PowerAfrica). pp. 1–7. IEEE (2012)

[58] Peng, L., Liu, S., Liu, R., Wang, L.: Effective long short-term memory with differential evolution algorithm for electricity price prediction. Energy **162**, 1301–1314 (2018)

[59] Petković, D., Ab Hamid, S.H., Ćojbašić, Ž., Pavlović, N.T.: Adapting project management method and anfis strategy for variables selection and analyzing wind turbine wake effect. Natural hazards **74**(2), 463–475 (2014)

[60] Petković, D., Ćojbašič, Ž., Nikolić, V.: Adaptive neuro-fuzzy approach for wind turbine power coefficient estimation. Renewable and Sustainable Energy Reviews **28**, 191–195 (2013)

[61] Petković, D., Ćojbašić, Ž., Nikolić, V., Shamshirband, S., Kiah, M.L.M., Anuar, N.B., Wahab, A.W.A.: Adaptive neuro-fuzzy maximal power extraction of wind turbine with continuously variable transmission. Energy **64**, 868–874 (2014)

[62] Petković, D., Nikolić, V., Mitić, V.V., Kocić, L.: Estimation of fractal representation of wind speed fluctuation by artificial neural network with different training algorothms. Flow Measurement and Instrumentation **54**, 172–176 (2017)

[63] Petković, D., Pavlović, N.T., Ćojbašić, Ž.: Wind farm efficiency by adaptive neuro-fuzzy strategy. International Journal of Electrical Power & Energy Systems **81**, 215–221 (2016)

[64] Piacentino, A., Catrini, P., Markovska, N., Guzović, Z., Vad Mathiesen, B., Ferrari, S., Duić, N., Lund, H.: Editorial: Sustainable development of energy, water and environment systems. Energy **190**, 116432 (2020). https://doi.org/https://doi.org/10.1016/j.energy.2019.116432, `http://www.sciencedirect.com/science/article/pii/S0360544219321279`

[65] Pousinho, H.M.I., Mendes, V.M.F., Catalão, J.P.d.S.: A hybrid pso–anfis approach for short-term wind power prediction in portugal. Energy Conversion and Management **52**(1), 397–402 (2011)

[66] Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE

congress on evolutionary computation. vol. 2, pp. 1785–1791. IEEE (2005)

[67] Richman, J.S., Moorman, J.R.: Physiological time-series analysis using approximate entropy and sample entropy. American Journal of Physiology-Heart and Circulatory Physiology **278**(6), H2039–H2049 (2000)

[68] Rodríguez, F., Florez-Tapia, A.M., Fontán, L., Galarza, A.: Very short-term wind power density forecasting through artificial neural networks for microgrid control. Renewable Energy **145**, 1517–1527 (2020)

[69] Saleh, A.E., Moustafa, M.S., Abo-Al-Ez, K.M., Abdullah, A.A.: A hybrid neuro-fuzzy power prediction system for wind energy generation. International Journal of Electrical Power & Energy Systems **74**, 384–395 (2016)

[70] Shamshirband, S., Petković, D., Amini, A., Anuar, N.B., Nikolić, V., Žarko Ćojbašić, Mat Kiah, M.L., Gani, A.: Support vector regression methodology for wind turbine reaction torque prediction with power-split hydrostatic continuous variable transmission. Energy **67**, 623–630 (2014)

[71] Shanshan, Q., Liu, F., Wang, J., Song, Y.: Interval forecasts of a novelty hybrid model for wind speeds. Energy Reports **1**, 8–16 (11 2015). https://doi.org/10.1016/j.egyr.2014.11.003

[72] Shi, Z., Liang, H., Dinavahi, V.: Direct interval forecast of uncertain wind power based on recurrent neural networks. IEEE Transactions on Sustainable Energy **9**(3), 1177–1187 (2017)

[73] Stojiljković, M.M.: Bi-level multi-objective fuzzy design optimization of energy supply systems aided by problem-specific heuristics. Energy **137**, 1231 – 1251 (2017). https://doi.org/https://doi.org/10.1016/j.energy.2017.06.037, `http://www.sciencedirect.com/science/article/pii/S0360544217310319`

[74] Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization **11**(4), 341–359 (1997)

[75] Sun, Y., Kirley, M., Li, X.: Cooperative co-evolution with online optimizer selection for large-scale optimization. In: Proceedings of the 2018 Genetic and Evolutionary Computation Conference. pp. 1079–1086. ACM (2018)

[76] Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4**(2), 26–31 (2012)

[77] Tvrdík, J., Misik, L., Krivy, I.: Competing heuristics in evolutionary algorithms. Intell. Technol. Theory Applicat pp. 159–165 (2002)

[78] Ulazia, A., Esnaola, G., Serras, P., Penalba, M.: On the impact of long-term wave trends on the geometry optimisation of oscillating water column wave energy converters. Energy **206**, 118146 (2020)

[79] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research **11**(Dec), 3371–3408 (2010)

[80] Xiaoyun, Q., Xiaoning, K., Chao, Z., Shuai, J., Xiuda, M.: Short-term prediction of wind power based on deep long short-term memory. In: 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC). pp. 1148–1152. IEEE (2016)

[81] Yu, R., Gao, J., Yu, M., Lu, W., Xu, T., Zhao, M., Zhang, J., Zhang, R., Zhang, Z.: Lstm-efg for wind power forecasting based on sequential correlation features. Future Generation Computer Systems **93**, 33–42 (2019)

[82] Yuan, X., Tan, Q., Lei, X., Yuan, Y., Wu, X.: Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine. Energy **129**, 122–137 (2017)

[83] Zanin, M., Rodríguez-González, A., Menasalvas Ruiz, E., Papo, D.: Assessing time series reversibility through permutation patterns. Entropy **20**(9), 665 (2018)

[84] Zhang, C., Zhou, J., Li, C., Fu, W., Peng, T.: A compound structure of elm based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting. Energy Conversion and Management **143**, 360–376 (2017)

[85] Zhao, F., Zeng, G.Q., Lu, K.D.: Enlstm-wpeo: Short-term traffic flow prediction by ensemble lstm, nnct weight integration, and population extremal optimization. IEEE Transactions on Vehicular Technology **69**(1), 101–113 (2019)

[86] Zhou, B., Ma, X., Luo, Y., Yang, D.: Wind power prediction based on lstm networks and nonparametric kernel density estimation. IEEE Access **7**, 165279–165292 (2019)

[87] Zhou, S., Xing, L., Zheng, X., Du, N., Wang, L., Zhang, Q.: A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times. IEEE Transactions on Cybernetics (2019)

[88] Zhu, Q., Li, H., Wang, Z., Chen, J., Wang, B.: Short-term wind power forecasting based on lstm. Dianwang Jishu/Power System Technology **41**, 3797–3802 (12 2017). https://doi.org/10.13335/j.1000-3673.pst.2017.1657

[89] Zuluaga, C.D., Alvarez, M.A., Giraldo, E.: Short-term wind speed prediction based on robust kalman filtering: An experimental comparison. Applied Energy **156**, 321–330 (2015)