Optimized forecasting model to improve the accuracy of very short-term wind power prediction

M. A. Hossain1, E. MacA. Gray1, J. Lu1, M. R. Islam2, M. S. Alam3, R. K. Chakrabortty2 and H. R. Pota2

Abstract—This paper proposes a novel framework to improve the prediction accuracy of very short-term (5-30 min) wind power generation. The framework consists of complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), monarch butterfly optimization (MBO) and long short-term memory (LSTM), called CEMOLS. The CEEMDAN is employed to extract complex hidden features of time-series data into intrinsic mode functions that are predicted using LSTM models with dropout regularization to retain long-term relationships between input and output data while the optimization algorithm tunes the hyper-parameters of the forecasting model. Data from four real wind farms in New South Wales are collected and pre-processed to train and test the forecasting models. Recently developed rival models are compared to identify the best-performing prediction model. The analysis demonstrates that the proposed CEMOLS with low computation time can improve forecasting accuracy on average by 32.96% in mean absolute error, 47.10% in root mean square error and 32.33% in mean absolute percentage error as compared to the benchmark Persistence model. It also demonstrates that sensitive and statistical analysis needs to be carried out to determine robust prediction models among rival models for practical application.

Index Terms—Wind power prediction, data decomposition, very short-term forecasting and optimization algorithm.

Nomenclature
The following abbreviations are used in this manuscript:

- Adagrad: Adaptive gradient algorithm
- Adam: Adaptive moment estimation
- AEMO: Australian Energy Market Operator
- API: Application Programming Interface
- ARIMA: Autoregressive integrated moving average
- ASA: Atom search algorithm
- CEEMDAN: Complete ensemble empirical mode decomposition with adaptive noise
- CNN: Convolutional Neural Network
- CSA: Crow search algorithm
- CS: Cuckoo search algorithm
- DA: Dragonfly algorithm
- EMD: Empirical mode decomposition
- EEMD: Ensemble EMD
- FS: Feature selection
- GRU: Gated recurrent unit
- GWO: Grey-wolf optimization
- GA-ANN: Genetic algorithm-artificial NN
- GOA: Grasshopper optimization algorithm
- IMF: Intrinsic mode function
- LSTM: Long short-term memory
- LR: Linear regression
- LDS: Lorenz disturbance sequence
- MAE: Mean absolute error
- MAPE: Mean absolute percentage error
- MBO: Monarch butterfly optimization
- MDMRMR: Maximum dependency, maximum relevancy and minimum redundancy
- MI: Mutual information
- NSW: New South Wales
- NWP: Numerical weather prediction
- NN: Neural network
- p-value: Probability value
- PI: Performance indices
- Ppd: Proposed model
- PCA: Principal component analysis
- PSO: Particle swarm optimization
- QR: Quantile regression
- RNN: Recurrent neural network
- RMSE: Root mean square error
- RMSProp: Root mean squared propagation
- SVM: Support vector machine
- SSA: Singular spectrum analysis
- VMD: Variational mode decomposition
- VSM: Variable selection module
- WSTD: Wavelet soft threshold denoising
- WT: Wavelet transform
- WNN: Wavelet neural network
- CPP: Critical-peak pricing

I. INTRODUCTION

Renewable energy sources are increasingly being integrated into distribution networks to avoid the greenhouse gas emissions of conventional generators, meet the growing need for power and secure a low-cost power supply. According to REN21’s 2020 report, renewable energy supplies power around 29% of global electricity demand, where 256 GW of renewable power capacity was added globally during the year, inspired by reductions in costs by around 85% and 56% since 2010 for solar and wind generation, respectively [1]. This indicates electricity generation is more cost-effective using renewable generators than from new coal-fired power plants. However, integrating renewable energy sources brings uncertainty in power generation and reduces network reliability, especially for large-scale generators, such as wind farms. As these sources are intermittent in nature, there is a supply-demand mismatch in the real-time operation of the network. This fluctuating power is
smoothed over time either by using a large energy storage system or an extra generator, causing higher maintenance and power generation costs [2]. The size of the energy storage system can be reduced by improving the accuracy of forecasting the availability of renewable energy. Enhancing accuracy is a cost-effective and environmentally friendly solution to the spinning reserve, but the stochastic natures of wind speed, temperature, and pressure make it hard to forecast wind power generation accurately.

To improve the prediction accuracy of wind power generation, several techniques have been reported in the literature [3], including physical, statistical and machine-learning models. The physical and statistical models suffer from inaccurate mathematical models, and the inability to learn and adapt, leading to higher errors in prediction [4]. To minimize errors, machine-learning models provide more reliable forecasting. They can be divided into conventional and deep learning models. The conventional models include K-nearest neighbors, support vector machine (SVM), Bayes learning, ensemble model and neural network (NN). These models have improved forecasting accuracy, but they cannot properly interrelate input and output data by extracting deep-level features without specialized feature engineering [3]. In this regard, deep learning models are capable of modelling a non-linear relationship between input and output data with flexible network structures to provide data-driven solutions [2]. Several techniques for constructing deep learning models have been applied to improve the forecast accuracy of wind power generation.

There are many publications on deep learning models to forecast wind power generation. A wavelet data decomposition technique to forecast the short-term and long-term wind power generation is used in [5], where particle swarm optimization algorithm is used to tune the weights of Convolutional Neural Network and a feature selection technique is used to select the fittest inputs with a conclusion of better prediction engine. In [6], a gated recurrent unit (GRU) to predict short-term wind power generation is employed, where an attention mechanism is used to obtain important input variables. A grid search algorithm is used to select hyper-parameters of GRU, but this is not an optimal way of selecting parameters and it is a very time-consuming process. A combination of SVM and dragonfly algorithm to forecast short-term wind power is presented in [7], where the dragonfly algorithm with adaptive learning factor and differential evolution strategy is employed to select the parameters. A combined model of LSTM, wavelet transform (WT) and principal component analysis is developed in [8], where a conditional distribution model indicates prediction error uncertainties. In [9], a grasshopper optimization algorithm based deep auto-regressive is presented to forecast hourly wind power generation, with the conclusion of improved accuracy compared to other neuro-evolutionary models. Lorenz disturbance distribution, neural network and empirical mode decomposition are presented in [10] to predict wind speed interval, where the distribution helps to predict the interval. In [11], a wavelet soft threshold denoising (WSTD) in combination with GRU is applied to predict short-term (1 h) wind speed, where the WSTD denoises the raw data of wind speed time series for improving forecasting accuracy. A hybrid model of the crow search algorithm, WT, feature selection (FS), and LSTM is demonstrated to predict short-term (1 h) wind speed in [12], where WT is used for decomposing wind speed and FS for ranking candidate inputs. In [13], a statistical short-term (up to 48 h) wind power forecast model is developed, where weather events are clustered with respect to the most important parameters to improve forecasting accuracy. A multi-source and temporal attention network is demonstrated in [14] for wind power forecasting of three wind farms, where multi-source NWP data and historical measurements are taken as inputs and forecasted next 48 hr wind power density as output. A quantile regression neural network is presented in [15] to forecast regional wind power generation, where an optimization algorithm is used for tuning weights and thresholds of the WNN, but the hyper-parameters of the models are not tuned. A 15-min wind speed prediction model based on double decompositions (CEEMDAN and VMD), error correction strategy and deep learning algorithm is developed in [17]. The double decomposition techniques increase the complexity of the forecasting model in practice due to the increase in the number of deep learning cycles applied for each decomposed data set, leading to high computational time. In [2], a novel framework consisting of a data decomposition method, a deep learning model and an optimization algorithm is demonstrated with a conclusion of higher accuracy as compared to an existing one. A bidirectional GRU is used in [18] to forecast 10-min data of wind speed that is decomposed using singular spectrum analysis (SSA) and VMD. The model is used for deterministic and probabilistic wind speed forecasting of two different regions without considering seasonal differences. In [19], a hybrid model of WT decomposition, atom search algorithm (ASA) and SVM is demonstrated to predict short-term (1 h) wind power generation, where wind speed and wind direction are considered as input data. The search ability of ASA is enhanced through dynamic sinusoidal wave adaptive weight and crossover and mutation operations. A hybrid cuckoo search arithmetic is used to tune an SVM to predict 10-min data of wind power generation in [20]. This study used only a limited data sample of size 1000, which may lead to higher errors in practice. VMD and GA-ANN are employed to decompose and predict 10-min wind-speed data in [21]. The hierarchical cluster method is utilized to determine high similarity data and then VMD decomposes K subsequences for getting better prediction accuracy.
TABLE I: Summary of literature

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Technique</th>
<th>Horizon</th>
<th>Tuning</th>
<th>Inputs</th>
<th>Com fairly</th>
<th>Data points</th>
<th>Stat anal</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5]</td>
<td>PSO + MDMRMR + CNN</td>
<td>1 h</td>
<td>Weights</td>
<td>Time-series</td>
<td>X</td>
<td>8760</td>
<td>No</td>
</tr>
<tr>
<td>[6]</td>
<td>GRU + attention mechanism</td>
<td>1 h</td>
<td>Parameters</td>
<td>Multi-variate</td>
<td>Yes</td>
<td>8760</td>
<td>No</td>
</tr>
<tr>
<td>[7]</td>
<td>SVM + DA</td>
<td>1 h</td>
<td>Kernel parameters</td>
<td>Multi-variate</td>
<td>X</td>
<td>8778</td>
<td>No</td>
</tr>
<tr>
<td>[8]</td>
<td>LSTM + WT + PCA</td>
<td>15 min</td>
<td>No</td>
<td>Multi-variate</td>
<td>X</td>
<td>6000</td>
<td>No</td>
</tr>
<tr>
<td>[9]</td>
<td>GOA + RNN</td>
<td>1 h</td>
<td>Hyper-parameters</td>
<td>Multi-variate</td>
<td>X</td>
<td>8760</td>
<td>No</td>
</tr>
<tr>
<td>[10]</td>
<td>EMD + NN + LDS</td>
<td>10 min</td>
<td>NA</td>
<td>Wind speed</td>
<td>X</td>
<td>1008</td>
<td>No</td>
</tr>
<tr>
<td>[11]</td>
<td>WSTD + GRU</td>
<td>1 h</td>
<td>NA</td>
<td>Wind speed</td>
<td>X</td>
<td>1000</td>
<td>No</td>
</tr>
<tr>
<td>[12]</td>
<td>CSA + WT + FS + MI + LSTM</td>
<td>1 h</td>
<td>Parameters</td>
<td>Wind speed</td>
<td>X</td>
<td>1200</td>
<td>No</td>
</tr>
<tr>
<td>[13]</td>
<td>K-means + LR</td>
<td>1 h</td>
<td>NA</td>
<td>Multi-variate</td>
<td>X</td>
<td>5040</td>
<td>No</td>
</tr>
<tr>
<td>[14]</td>
<td>LSTM + VSM</td>
<td>1 h</td>
<td>Parameters</td>
<td>Multi-variate</td>
<td>X</td>
<td>8760</td>
<td>No</td>
</tr>
<tr>
<td>[15]</td>
<td>NN + QR</td>
<td>1 h</td>
<td>NA</td>
<td>No</td>
<td>X</td>
<td>4392</td>
<td>No</td>
</tr>
<tr>
<td>[16]</td>
<td>CEEMDAN + WNN</td>
<td>10 min</td>
<td>Weights</td>
<td>Uni-variate</td>
<td>X</td>
<td>1500</td>
<td>No</td>
</tr>
<tr>
<td>[17]</td>
<td>CEEMDAN + VMD + LSTM</td>
<td>15 min</td>
<td>NA</td>
<td>Wind speed</td>
<td>X</td>
<td>2160</td>
<td>No</td>
</tr>
<tr>
<td>[2]</td>
<td>VMD + LSTM + GWO</td>
<td>5 min</td>
<td>Parameters</td>
<td>Uni-variate</td>
<td>X</td>
<td>2102</td>
<td>No</td>
</tr>
<tr>
<td>[18]</td>
<td>SSA + VMD + GRU</td>
<td>10 min</td>
<td>NA</td>
<td>Wind speed</td>
<td>Yes</td>
<td>8784</td>
<td>No</td>
</tr>
<tr>
<td>[19]</td>
<td>WT + ASA + SVM</td>
<td>1 h</td>
<td>Parameters</td>
<td>Multi-variate</td>
<td>X</td>
<td>8760</td>
<td>No</td>
</tr>
<tr>
<td>[20]</td>
<td>CS + SVM</td>
<td>10 min</td>
<td>Parameters</td>
<td>Multi-variate</td>
<td>X</td>
<td>1000</td>
<td>No</td>
</tr>
<tr>
<td>[21]</td>
<td>VMD + GA-ANN</td>
<td>10 min</td>
<td>No</td>
<td>Wind speed</td>
<td>X</td>
<td>4464</td>
<td>No</td>
</tr>
<tr>
<td>This study</td>
<td>CEEMDAN + MBO + LSTM</td>
<td>5 min</td>
<td>Hyper-parameters</td>
<td>Time-series</td>
<td>Yes</td>
<td>8640</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Description: Com: Compare; Stat anal: Statistical Analysis; X: Not available in details
The above literature review, summarized in Table I, reveals a number of opportunities to improve forecasting accuracy. The majority of forecasting models are: (1) for either short or long-term forecasting, which is relatively easy compared to very short-term forecasting due to missing dynamic characteristics of power generation [2]; (2) predicted using multiple variables, which also need to be forecasted, resulting in higher errors; (3) written on wind speed forecasting, but wind speed needs to be converted into wind power generation, leading to cumulative forecasting errors; and (4) often overlooked the parameter tuning, especially for rival prediction models, therefore a fair comparison could not be made. Moreover, wind farms in Australia have recently begun to participate in real-time operations (spot market) to avoid high uncertainty in power generation and achieve economic benefits [22]. This is because the Australian Energy Market Commission has changed the rules of the settlement interval for the electricity spot price from 30 min to 5 min, starting in September 2021 [23]. All the participating generators must submit the price and quantity of electricity to the system operator of Australian Energy Market Operator (AEMO) [24], where the central dispatch engine is run every five minutes and orders the generators’ offers from least to most expensive for dispatching the lowest cost mix of generators to fulfill power demand. A wind farm, willing to participate in the wholesale market, requires to bid to supply the electricity in 5 min blocks. Therefore, wind farms need to predict power generation 5 min ahead, the prediction that eventually participates in fulfilling the mismatched power demand resulting from sudden positive and negative ramps. Through the real-time energy dispatch of a wind farm, a network’s stability and reliability can also be enhanced. Studying this very short-term power generation assists in understanding the dynamic behaviors of a wind turbine, monitoring the online health of the wind farm, optimizing performance, and controlling the frequency and voltage [3]. The majority of existing literature reports have overlooked the essence of the very short-term wind power generation with hyper-parameter tuning using an optimization algorithm in addition to sensitivity and statistical analysis. To bridge the research gaps, this paper contributed to knowledge as follows.

- A novel framework (CEMOLS) is developed to enhance the prediction accuracy of very short-term wind power generation. The CEMOLS consists of data decomposition, deep learning and optimization techniques. The CEMOLS is employed for time-series data that do not depend on other forecasting input data, resulting in low forecasting error.
- A new Monarch Butterfly Optimization (MBO) algorithm, a nature-inspired meta-heuristic algorithm, is employed to automatically tune the hyper-parameters of the prediction model after a certain time, say one year. Selecting the best hyper-parameters plays a critical role in improving the performance of the deep learning model that is carried out with less computational time in the MBO algorithm.
- Complete ensemble empirical mode decomposition with adaptive noise is employed to decompose time-series data through several sequences of different frequencies, called Intrinsic Mode Function (IMF). This data decomposition technique is crucial to extract hidden features of time-series data and convert them into IMFs to enhance the accuracy of the forecasting data by reducing prediction errors caused by data fluctuation.
- Sensitivity and statistical analysis are carried out for very short-term wind generation of four real wind farms to demonstrate the effectiveness of the proposed forecasting model. This analysis ensures that the forecasting model can be employed in practice within the range of accuracy. Most of the previous literature has overlooked this analysis, leading to poor performance in practical application.

In the existing forecasting models, consisting of data decomposition, deep learning and an optimization algorithm, the weights of deep learning are usually determined using the optimization algorithm without properly tuning the hyper-parameters of the model. Unlike the other models, the proposed CEMOLS has some unique features, including an automatic hyper-parameter tuning framework of deep learning to improve forecasting accuracy after a certain time say one year, integration of a new optimization algorithm in a forecasting model, optimization problem formulations, fair comparisons with a benchmark model and a new application for very short-term wind power forecast. These attributes of a forecasting model are lacking in the existing literature.

The remainder of the paper is structured as follows. Section II presents the data decomposition technique and deep learning model. In Section III, the MBO algorithm is integrated into the forecasting model to tune its hyper-parameters for enhancing prediction accuracy of very short-term (5-min) wind power generation. Experimental results from the proposed CEMOLS are analyzed using sensitivity and statistical methods for four wind farms in Section IV. Section VI summarizes the outcome of the analysis and delivers concluding remarks.

II. Forecasting Models

A. Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)

As wind power generation is complex, nonlinear and non-stationary in nature [25], an empirical mode decomposition (EMD) algorithm can be applied to extract important features of power generation. In this process, time-series data are decomposed into a series of IMFs and a residual to obtain the intrinsic mode of the signal from time-scale data [26]. IMFs are more stable compared to the original data for predicting complex time-series wind power generation. While the EMD technique is an efficient signal processing method, its main drawback is mode aliasing endpoint effects. To avoid these effects, an improved version of EMD is developed in [27], called EEMD, which is a multiple-trial process to add a limited magnitude of Gaussian white noise into the original signal every time. As Gaussian white noise has the characteristics of uniform frequency distribution, the various time-scale signals are automatically distributed to proper reference scales that lead to the avoidance of mode aliasing. This technique may lead to incomplete decomposition and reconstruction errors if the trial numbers are not sufficient. To avoid these problems in EEMD, a
CEEMDAN was presented in [28]. The following steps are used to decompose time-series data.

- Step 1: \( P_o(t) = P(t) + \epsilon_o w^o(t) \)
  where \( f \in \{1, 2, \ldots, F\} \) is the noise realization index and \( \epsilon_o \) is the coefficient to measure the signal-to-noise ratio. By decomposing \( P_o(t) \) using the EMD method, we can obtain the first EMD mode \( C_1^f \) and then determine
  \[
  \tilde{c}_1(t) = \frac{1}{F} \sum_{f=1}^{F} c_1^f(t). \tag{1}
  \]

- Step 2: Determine the 1st residue as follows
  \[
  \tilde{r}_1(t) = P(t) - \tilde{c}_1(t). \tag{2}
  \]

- Step 3: Decompose the noise-added residue \( \tilde{r}_1^f(t) = \tilde{r}_1(t) + \epsilon_1 \mathbb{E}_1[w^f(t)] \) to find the second CEEMDAN IMFs as follows:
  \[
  \tilde{c}_2(t) = \frac{1}{F} \sum_{f=1}^{F} \mathbb{E}_1[\tilde{r}_1^f(t)] \tag{3}
  \]
  where \( \mathbb{E}_f[.] \) is a function/operator to extract the jth IMF decomposed by EMD modes of [.]

- Step 4: Determine \( i \)th residue for \( i = 2, 3, \ldots \) as
  \[
  \tilde{r}_i(t) = \tilde{r}_{i-1}(t) - \tilde{c}_i(t). \tag{4}
  \]

- Step 5: Decompose the noise-added residue \( \tilde{r}_i^f(t) = \tilde{r}_i(t) + \epsilon_i \mathbb{E}_i[w^f(t)] \) to define \((k+1)\)th CEEMDAN IMF as follows
  \[
  \tilde{c}_{i+1}(t) = \frac{1}{F} \sum_{f=1}^{F} \mathbb{E}_i[\tilde{r}_i^f(t)]. \tag{5}
  \]

Repeat the process from Steps 4 and 5 until it reaches a state from which the residue \( \tilde{r}_i \) can not be decomposed anymore.

The effective IMFs are selected based on the correlation coefficient of the raw signal and each IMF. The IMFs of the CEEMDAN are used as input data for the deep learning model (LSTM). These IMFs are more convenient to predict than original time-series data.

B. Long short-term memory (LSTM)

The LSTM network is an improved version of recurrent neural network (RNN) to (a) obtain permanent memories, (b) minimize rate loss in signal and (c) maintain long-term dependencies [29]. These characteristics are achieved by implementing a gate system to regulate signal flow, i.e. the gradient flow, for overcoming the vanishing gradient issues in RNN. The four gates (input, forget, update and output) used in this network can be mathematically expressed as follows [2]:

\[
\begin{align*}
  i_t &= \sigma(W_i[x_{t-1,x_t} + b_i]) \tag{6} \\
  \tilde{c}_t &= \tanh(W_c[x_{t-1,x_t} + b_c]) \tag{7} \\
  f_t &= \sigma(W_f[x_{t-1,x_t} + b_f]) \tag{8} \\
  c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{9} \\
  o_t &= \sigma(W_o[x_{t-1,x_t} + b_o]) \tag{10} \\
  h_t &= o_t \odot \tanh(c_t) \tag{11}
\end{align*}
\]

where \( W_i \) is the weight matrix of the \( i \)th; \( W_c \) the weight matrix of \( \tilde{c}_t \); \( W_f \) the weight matrix of the gate layer to delete or store information; \( W_o \) the weight matrix of the output gate layer; \( \sigma \) the sigmoidal function; \( b_i \) and \( b_c \) the bias terms of input gate, \( b_f \) the bias term of the gate layer, and \( b_o \) the bias term of the output gate layer. The \( \odot \) is the Hadamard product of two states and the square brackets refer to the addition of two vectors.

A dropout regularization is introduced to the LSTM model in the process of signal transmission to avoid the over-fitting issue. To produce a set of random vectors in between 0 and 1 to erase random status, the mathematical model can be written as follows:

\[
r \sim \text{Bernoulli}(p)
\]

where \( r \) and \( p \) are the random vectors and the dropout rate, respectively.

III. INTEGRATED FRAMEWORK PROPOSED

A. Monarch Butterfly Optimization algorithm

The MBO algorithm in [30], simulated the migration behavior of monarch butterflies in nature, has been successfully employed in many applications, such as dynamic vehicle routing problems, test categorization, and energy management, due to its fast convergence, high accuracy and easy implementation [31]. The algorithm is based on two equal-sized Subpopulations: Subpopulation 1 and Subpopulation 2, and thereby two strategies: migration operator and butterfly adjusting operator.

1) Migration operator: The objective of this operator is to improve communication in subpopulations, where the movement of each butterfly \( i \) in Subpopulation 1 is influenced by the position of others that is balanced by an adjusting ratio \( p \). The mathematical model of the individual \( i \) on the \( k \)th dimension in Subpopulation 1 can be written as follows:

\[
x_{i,k}^{t+1} = \begin{cases} 
  x_{i,k}^{t}, & \text{if } r \leq p \\
  x_{j,k}^{t+1}, & \text{otherwise}
\end{cases}
\]

where \( x_{i,k}^{t+1} \) is the \( k \)th element of \( X_i \) at generation \( t+1 \) for representing the position of the monarch butterfly \( i; r_1, r_2 \) is the randomly chosen integer value from Subpopulations; and \( r = \mathbb{R} \times \mathbb{P} \) is the switching mechanism, where \( \mathbb{R} \) stands for random real number in [0, 1] and \( \mathbb{P} \) indicates migration period.

2) Butterfly adjusting operator: The position of monarch butterflies can also be updated by the butterfly adjusting operator. For all elements in monarch butterfly \( j \), the new individual can be mathematically expressed as:

\[
x_{j,k}^{t+1} = \begin{cases}
  x_{\text{best},k}^{t}, & \text{if } R \leq p \\
  x_{j,k}^{t+1}, & \text{if } R > p \\
  x_{j,k}^{t+1} + \alpha \times (dx_k - 0.5) & \text{if } B_r < R > p
\end{cases}
\]

where \( x_{\text{best},k}^{t} \) represents the \( k \)th element of \( x_{\text{best}}^{t} \) for identifying the best monarch butterfly in Subpopulations, \( r_3 \) is a set of integer index, and \( B_r \) stands for the butterfly adjusting rate. The \( dx \), the walk step of the monarch butterfly \( j \), and \( \alpha \), the weighting factor, can be calculated as follows:

\[
dx = \text{Levy}(x_{i}^j) / \alpha = S_{\text{max}} / t^2
\]

where \( S_{\text{max}} \) is the max walk step.
B. Objective function

The objective of the study is to minimize the Huber loss function to improve the forecast accuracy by updating weights and hyper-parameters of the deep learning model. We have chosen the Huber loss function due to its robustness against outliers of the input data [3]. The function is mathematically represented as follows:

\[ L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2} (y - \hat{y})^2 & \text{if } |y - \hat{y}| \leq \delta \\ \delta |y - \hat{y}| - \frac{1}{2} \delta^2 & \text{otherwise} \end{cases} \] (15)

where \( \delta \) is the control mechanism for switching between mean square error (MSE) and mean absolute error (MAE) and it is a positive real number.

1) Constraints: The stopping criteria on the optimization algorithm is the number of iterations, set to 100. After reaching the maximum number of iterations, the program will stop and deliver the optimal hyperparameters of the deep learning models. This can be mathematically represented as \( t_{\text{max}} < 100 \).

C. CEMOLS

The integration of the optimization algorithm into the data decomposition technique and deep learning model called CEMOLS is depicted in Fig. 2. In the integration process, there are two inputs: one is for data incoming from where new data will be fed to predict wind power generation and another input is to run the MBO algorithm. For the first-time use, the MBO will optimize the hyper-parameters of the deep learning model while training the model using time-series data. Once the best hyper-parameters are selected, the MBO algorithm will not be run and it has no function until it is called in the algorithmic process. In this stage, time-series data will be forecasted. The detailed process of data pre-processing, model training and hyper-parameter selections are hierarchically described below.

1) Data processing: Before training the forecasting model with the MBO algorithm, it is important to process the raw data for a better accuracy model of prediction data. The data processing is described as follows.

a) The raw data of wind power generation is loaded and processed to eliminate missing, outliers and noise data that are introduced during the data collection process in smart meters.

b) These data are normalized using the maximum peak value to improve forecasting accuracy.

c) The normalized data are decomposed using CEEM-DAN from time-series data to IMFs to remove non-stationary and non-linearity.

d) The data are split into training (75%) and testing (25%) sets.

2) Model training: The time series data are now ready to train the deep learning model. The following procedure is followed to train the deep learning model.

a) The training data set of each IMF is used to train each LSTM model with dropout regularization.

b) The outcome of each model is added to reconstruct the original time-series data predicted.

c) The normalized data are backed to raw data magnitude by multiplying the maximum wind power generation.

d) The performance indices (MAE, RMSE and MAPE) are calculated.

e) The developed model is employed to forecast the test data set to demonstrate the effectiveness of the forecasting model.

3) Hyper-parameter selection: The MBO algorithm is integrated into the first step of the model training process as shown in Fig. 2 as a flowchart. The objective of the algorithm is to minimize the Huber loss function of the LSTM model described in section III-B. The integration to tune hyper-parameters of the model is carried out in the following steps.

a) The parameters of the MBO algorithm are defined, such as cross-validation (10), population size (50), run times (5) and the maximum generations (100).

b) The search space of hyperparameters is set as tabulated in Table II.

c) Call Model training, III-C3, from a to c procedures.

In this stage, the LSTM models are operated to update their weights for selecting better parameters depending on the Huber loss function.

d) The LSTM model is evaluated with reconstructed time-series prediction using the performance matrices: mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE).

e) The optimization algorithm keeps running until the large search space reduces to a narrow one with an effective solution for predicting wind power generation.

f) The optimization algorithm returns the best combination of hyperparameters shown in Table II.

The convergence curve of the objective function is demonstrated in Fig. 1 from where the best parameters are achieved by less than 50 iterations.

Remark: Many articles have reported the performance indices (PI) with normalized values, resulting in low numeric values in PI, such as PI < 0.1. This may misinterpret the forecasting errors because low normalized values will produce low PI values regardless of actual wind power capacity.

TABLE II: Outcome of parameter tuning for forecasting models.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
<td>3.1 (3)</td>
</tr>
<tr>
<td>MAE</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
</tr>
<tr>
<td>MAPE</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
<td>0.4 (0–1)</td>
</tr>
</tbody>
</table>

Fig. 1: Convergence curve of objective function.

IV. EXPERIMENTAL RESULTS

This section is dedicated to demonstrating the effectiveness of the proposed CEMOLS in forecasting the 5-
min wind power generation of four wind farms in New South Wales (NSW), Australia. The proposed CEMOLS consists of CEEMDAN, LSTM and MBO techniques to automatically select the best hyperparameters for the deep learning model. The MBO algorithm is mainly responsible for fine-tuning the parameters as described in Section III while the CEEMDAN decomposes the non-linear time-series data into IMFs for better prediction accuracy. All the data from the wind farms are collected and pre-processed to train the LSTM model. In this model, we have used 8,640 data points in total, where 75% of the data are used for training the model and 25% for testing purposes. All the coding is implemented in Python 3.9 software using the Keras API integrated with TensorFlow.

The forecasting model is trained using 5-min data set of the Boco Rock wind farm and it is used for parameter tuning of the model using the MBO algorithm described in Section II. The competitive forecasting model is also tuned in the same way as the proposed model using the MBO algorithm to fairly compare the effectiveness of the models presented. After grid-search analysis for selecting a number of LSTM layers, we have selected two network layers with a dropout rate at the middle followed by a Dense layer. MBO algorithm is then applied to select the number of neurons in a layer, a dropout rate, an optimization algorithm for weight update, an optimal number of input values, and the number of batch sizes.

A. Data set 1: Boco Rock Wind Farm

The wind farm is situated on ridgelines 175 km away from the capital of Australia, Canberra. The farm produces wind power up to 113 MW by 67 wind turbines with a rotor diameter of 100 m and hub heights from 80 m – 100 m: 58×1.7 MW turbines with 4.62 m²/kW power density and 9×1.6 MW turbines with 4.91 m²/kW power density. The cut-in, rated and cut-off wind speeds are 3 m/s, 11 m/s and 23 m/s, respectively. The rotor diameter and swept area of the wind turbines are 100 m and 7,854 m², respectively.

B. Data set 2: Bodangora Wind Farm

The Bodangora wind farm, located in the Central West of NSW and 370 km away from Canberra, has a total power generation capacity of 113.2 MW from 33×3.4 MW wind turbines with a hub height of 85 m and a rotor diameter of 130 m. From the power generated, around 60% is supplied to Energy Australia and the rest 40% is managed by Infigen’s energy market risk framework.

The wind power generation shown in Fig. 3b is initially small with a high variation of up to 500 samples, and then there are some rising and falling ramps with maximum power generation capacity. No power is generated in some circumstances, this effect is predictable, for example, operation in the cut-off zone in extreme weather conditions and in the cut-in zone at low wind speeds. Wind generators are also not operated during maintenance periods. The performance of the forecasting model is measured using MAE, RMSE and MAPE. From the power generation prediction of the wind farm, the forecasting errors for MAE, RMSE and MAPE are 1.31, 1.89, and 3.83, respectively. These performance indices are 40.99%, 48.92% and 33.62% less as compared to the performance indices of the Persistence or Naïve model which is considered a benchmark model in the prediction area. The Persistence model considers the prediction value at t + 1 to be equal to the value at t. The calculated performance matrices are 2.22, 3.70 and 5.77 for MAE, RMSE and MAPE, respectively.

The forecasting outcome of the Boco Rock wind farm is demonstrated in Fig. 3a. The forecast power generation closely follows the actual power generation. All the errors are less than 10 MW of power generation, mostly in the 2 MW range. During a transition either from low to high or vice versa, prediction errors are found higher than normal wind power generation as it is always difficult to accurately predict these types of transient conditions. In some circumstances, this effect is predictable, for example, operation in the cut-off zone in extreme weather conditions and in the cut-in zone at low wind speeds. Wind generators are also not operated during maintenance periods. The performance of the forecasting model is measured using MAE, RMSE and MAPE. From the power generation prediction of the wind farm, the forecasting errors for MAE, RMSE and MAPE are 1.31, 1.89, and 3.83, respectively. These performance indices are 40.99%, 48.92% and 33.62% less as compared to the performance indices of the Persistence or Naïve model which is considered a benchmark model in the prediction area. The Persistence model considers the prediction value at t + 1 to be equal to the value at t. The calculated performance matrices are 2.22, 3.70 and 5.77 for MAE, RMSE and MAPE, respectively.
around samples 200 and 875. The normal error values are within 2 MW. The MAE, RMSE and MAPE of the wind power forecasting are 1.75, 2.33 and 3.36, respectively. The Persistence model records 1.97 for MAE, 3.68 RMSE and 5.45 MAPE. Therefore, the proposed model demonstrates 11.17%, 36.68%, and 38% less errors as compared to the Persistence model.

C. Data set 3: Capital Wind Farm

The Capital wind farm, situated in NSW, is built to offset the power usage of the Kurnell Desalination Plant. The power generation capacity of the farm is around 140.7 MW from 67×2.1 MW turbines with a hub height of 80 m and a rotor diameter of 88 m. Total energy generation averages 374 GWh/year.

In this wind farm, the forecasting power generation capacity has high uncertainty in the middle of the prediction sample shown in Fig. 3c, say from sample 800 to 1500 samples. The wind farm is mostly operated in this sample period, except for around 250 – 500 samples. The forecasting outcome of the wind farm is demonstrated in Fig. 3c. The forecasting values are near to actual wind power generation. There is no noticeable spike in the errors and all the errors are within 5 MW. The mean absolute error of this forecasting is 1.23, the spread of the error is 1.62 which is slightly higher than MAE, and the percentage error is only 2.97. In comparison to the Persistence model (1.97, 3.69, and 5.45), the proposed model shows 37.56%, 56.1%, and 33.85% less errors for MAE, RMSE and MAPE, respectively.

D. Data set 4: Gullen Range Wind Farm

Gullen Range Wind Farm, located in the Southern Tablelands region of NSW and 106 km away from Canberra, has 73 wind turbines generating up to 165.5 MW at their rated wind speed of 12 m/s. Two types of wind turbines, GRW100-2.5 MW (56 turbines, 80 m hub height and 100 m rotor diameter) and GW82-1.5 MW (17 turbines, 85 m hub height and 82 m rotor diameter), are installed.

The power forecasting pattern of this wind farm starts with zero power generation until sample 500, and after
that exhibits high uncertainty in power generation. Fig. 3d demonstrates the forecasting accuracy of wind power generation. From the error scale, it is observed that all the errors are within 5 MW. There is a small error spike around 825 samples due to a sudden power generation spike. The errors are within 5 MW. There is a small error spike around 3d demonstrates the forecasting accuracy of wind power generation. Fig. 4 demonstrates the scatter-plot of wind power forecasting vs actual wind power for all rival four models. Please note that the Persistence model's output does not vary with multiple runs. The solid line demonstrates the perfect fit for input and output data that are measured by $R^2$. $R^2$ measures the amount of variation in the data, a higher value close to 1 indicates the perfect fit. From Fig. 4, it is observed that the best-performing forecasting model based on $R^2$ values is LSTM-VMD-MBO for data set 1 followed by CEMOLS and CNN-GRU-NN. The performance of both data decomposition techniques are very close in terms of $R^2$, only the third digit after the dot is the difference. For data set 2 shown in Fig. 5, the LSTM-VMD-MBO model is again showing the best performance in the third digit of $R^2$ followed by CEMOLS and the Persistence model. The lowest performance is observed by the CNN-GRU-NN model. In the case of data set 3, the proposed model demonstrates the highest performance as compared to the other models shown in Fig. 6. The Persistence model shows the lowest performance to predict wind power generation as data set 3 are fluctuating in higher degrees than data set 2. In the last comparison of data set 4 shown in Fig. 7, it is found that the proposed CEMOLS demonstrates again the highest performance to predict wind power generation followed by LSTM-VMD-MBO and the Persistence model. From a random single run, it is not clear which model is the best to predict very short-term wind power generation as both models show similar performance due to model training and testing in the same framework. As the outcome can be altered in the next random runs, a comparison needs to be further carried out with multiple runs to demonstrate the robustness of the practical application. In the following section, the prediction models are compared with sensitivity analysis.

### TABLE II: Sensitivity analysis of models for data set 1.

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE (MW)</th>
<th>Median</th>
<th>Mean</th>
<th>Worst</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
<tr>
<td>LSTM-VMD-MBO</td>
<td>2.35</td>
<td>2.41</td>
<td>2.45</td>
<td>3.17</td>
<td>0.61</td>
</tr>
<tr>
<td>CNN-GRU-NN</td>
<td>2.40</td>
<td>2.50</td>
<td>2.54</td>
<td>3.25</td>
<td>0.64</td>
</tr>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
</tbody>
</table>

### TABLE IV: Sensitivity analysis of models for data set 2.

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE (MW)</th>
<th>Median</th>
<th>Mean</th>
<th>Worst</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
<tr>
<td>LSTM-VMD-MBO</td>
<td>2.35</td>
<td>2.41</td>
<td>2.45</td>
<td>3.17</td>
<td>0.61</td>
</tr>
<tr>
<td>CNN-GRU-NN</td>
<td>2.40</td>
<td>2.50</td>
<td>2.54</td>
<td>3.25</td>
<td>0.64</td>
</tr>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
</tbody>
</table>

### TABLE V: Sensitivity analysis of models for data set 3.

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE (MW)</th>
<th>Median</th>
<th>Mean</th>
<th>Worst</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
<tr>
<td>LSTM-VMD-MBO</td>
<td>2.35</td>
<td>2.41</td>
<td>2.45</td>
<td>3.17</td>
<td>0.61</td>
</tr>
<tr>
<td>CNN-GRU-NN</td>
<td>2.40</td>
<td>2.50</td>
<td>2.54</td>
<td>3.25</td>
<td>0.64</td>
</tr>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
</tbody>
</table>

### TABLE VI: Sensitivity analysis of models for data set 4.

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE (MW)</th>
<th>Median</th>
<th>Mean</th>
<th>Worst</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
<tr>
<td>LSTM-VMD-MBO</td>
<td>2.35</td>
<td>2.41</td>
<td>2.45</td>
<td>3.17</td>
<td>0.61</td>
</tr>
<tr>
<td>CNN-GRU-NN</td>
<td>2.40</td>
<td>2.50</td>
<td>2.54</td>
<td>3.25</td>
<td>0.64</td>
</tr>
<tr>
<td>Persistence</td>
<td>2.14</td>
<td>2.18</td>
<td>2.23</td>
<td>3.04</td>
<td>0.50</td>
</tr>
</tbody>
</table>

### Comparative study without sensitivity analysis

In this section, we present a random single-run comparative forecasting analysis with other competitive models (with and without data decomposition techniques) to determine the best-performing model. To compare with data decomposition techniques, VMD, one of the latest signal decomposition techniques, is selected as a competitive model because of its higher accuracy in prediction [32]. This data decomposition technique is reported to be robust in the presence of sampling and noise data, aiding the proper reconstruction of the actual signal from the decomposed IMFs. It is also reported in [2] that VMD demonstrates better data decomposition performance compared to WT. The outcome of the VMD data decomposition is fed to the same LSTM models used in the proposed CEMOLS. The forecasting model is called LSTM-VMD-MBO. To avoid mode aliasing and noisy IMFs that are related to the prediction accuracy of the VMD model, we have observed the interference effect of the IMF frequency and have selected 10 modes of IMFs, as increasing the number of modes has the effect of mode aliasing while only 6 IMFs are chosen for CEEMDAN to reduce computational complexity while retaining good performance. The proposed model (CEMOLS) is also compared with a newly developed hybrid deep learning model consisting of CNN, GRU and NN (CNN-GRU-NN) in [3] to understand the performance improvement compared to modelling without data decomposition techniques. The tuning process of the CNN-GRU-NN model is elaborately analyzed in that study using a grid search technique for different hyper-parameters, including the number of neurons, optimization algorithm and input data. The model has demonstrated the lowest prediction error compared to other existing models, including ARIMA and SVM, for several sets of power generation data. All the rival models are tuned properly to fairly present their weaknesses and strengths.

For a random single run, Fig. 4 demonstrates the scatter-plot of wind power forecasting vs actual wind power for all rival four models. Please note that the Persistence model’s output does not vary with multiple runs. The solid line demonstrates the perfect fit for input and output data that are measured by $R^2$. $R^2$ measures the amount of variation in the data, a higher value close to 1 indicates the perfect fit. From Fig. 4, it is observed that the best-performing forecasting model based on $R^2$ values is LSTM-VMD-MBO for data set 1 followed by CEMOLS and CNN-GRU-NN. The performance of both data decomposition techniques are very close in terms of $R^2$, only the third digit after the dot is the difference. For data set 2 shown in Fig. 5, the LSTM-VMD-MBO model is again showing the best performance in the third digit of $R^2$ followed by CEMOLS and the Persistence model. The lowest performance is observed by the CNN-GRU-NN model. In the case of data set 3, the proposed model demonstrates the highest performance as compared to the other models shown in Fig. 6. The Persistence model shows the lowest performance to predict wind power generation as data set 3 are fluctuating in higher degrees than data set 2. In the last comparison of data set 4 shown in Fig. 7, it is found that the proposed CEMOLS demonstrates again the highest performance to predict wind power generation followed by LSTM-VMD-MBO and the Persistence model. From a random single run, it is not clear which model is the best to predict very short-term wind power generation as both models show similar performance due to model training and testing in the same framework. As the outcome can be altered in the next random runs, a comparison needs to be further carried out with multiple runs to demonstrate the robustness of the practical application. In the following section, the prediction models are compared with sensitivity analysis.
Fig. 8: Median error values with standard deviation for (a) data set 1 (b) data set 2 (c) data set 3 and (d) data set 4.

F. Comparative study with sensitivity analysis

We have compared the accuracy of the forecasting models for multiple independent runs to feed into sensitivity analysis in order to identify the best-performing model in practice. This analysis plays an important role in determining a model’s effectiveness, as the deep learning model provides different results for each run. We have conducted eleven independent runs with different seed values. Although a higher number of runs will provide a more robust outcome, it has a high computational cost. Median values with standard deviation are selected as a measuring scale for comparison purposes because of their position among multiple outcomes. In this analysis, the Persistence model is not compared as its outcome does not change with independent runs.

TABLE VII: Wilcoxon signed-rank test for data set 1.

<table>
<thead>
<tr>
<th></th>
<th>MAE</th>
<th>R</th>
<th>p-value</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ppd vs LSTM-VMD-MBO</td>
<td>12.0</td>
<td>20.0</td>
<td>0.00178</td>
<td>≥</td>
</tr>
<tr>
<td>Ppd vs CNN-GRU-NN</td>
<td>66.0</td>
<td>0.0</td>
<td>9.766E-4</td>
<td>+</td>
</tr>
<tr>
<td>RMSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ppd vs LSTM-VMD-MBO</td>
<td>41.0</td>
<td>25.0</td>
<td>0.0122</td>
<td>=</td>
</tr>
<tr>
<td>Ppd vs CNN-GRU-NN</td>
<td>66.0</td>
<td>0.0</td>
<td>9.766E-4</td>
<td>+</td>
</tr>
<tr>
<td>MAPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ppd vs LSTM-VMD-MBO</td>
<td>52.0</td>
<td>14.0</td>
<td>0.00186</td>
<td>=</td>
</tr>
<tr>
<td>Ppd vs CNN-GRU-NN</td>
<td>66.0</td>
<td>0.0</td>
<td>9.766E-4</td>
<td>+</td>
</tr>
</tbody>
</table>

TABLE VIII: Wilcoxon signed-rank test for data set 2.

<table>
<thead>
<tr>
<th></th>
<th>MAE</th>
<th>R</th>
<th>p-value</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ppd vs LSTM-VMD-MBO</td>
<td>12.0</td>
<td>20.0</td>
<td>0.00178</td>
<td>≥</td>
</tr>
<tr>
<td>Ppd vs CNN-GRU-NN</td>
<td>66.0</td>
<td>0.0</td>
<td>9.766E-4</td>
<td>+</td>
</tr>
<tr>
<td>RMSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ppd vs LSTM-VMD-MBO</td>
<td>16.0</td>
<td>20.0</td>
<td>0.0122</td>
<td>=</td>
</tr>
<tr>
<td>Ppd vs CNN-GRU-NN</td>
<td>66.0</td>
<td>0.0</td>
<td>9.766E-4</td>
<td>+</td>
</tr>
<tr>
<td>MAPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ppd vs LSTM-VMD-MBO</td>
<td>28.0</td>
<td>38.0</td>
<td>0.00186</td>
<td>=</td>
</tr>
<tr>
<td>Ppd vs CNN-GRU-NN</td>
<td>66.0</td>
<td>0.0</td>
<td>9.766E-4</td>
<td>+</td>
</tr>
</tbody>
</table>

The accuracy of the forecasting models to data set 1 is demonstrated in Table III. The highest errors for MAE, RMSE and MAPE can be observed in the CNN-GRU-NN model although it has a consistent outcome, i.e., the lowest standard deviation (Std.). The performance of both data decomposition techniques (LSTM-VMD-MBO and CEMOLS) is similar, which is confirmed in Fig. 8a with standard deviation. For predicting data set 2, the sensitivity analysis is presented in Table IV where it can be seen that LSTM-VMD-MBO produces the lowest median value, but with a higher standard deviation, compared to the proposed one shown in Fig. 8b. The CNN-GRU-NN model is the worst-performing model. In terms of spreading the forecasting errors, the proposed model demonstrates reliable forecasting capability. For data set 3, the proposed model produces the lowest median value errors compared to LSTM-VMD-MBO, as shown in Table V. The LSTM-VMD-MBO model has lower median MAE and RMSE values, but the standard deviation is quite high compared to CNN-GRU-NN. In terms of MAE, CNN-GRU-NN has lower errors than LSTM-VMD-MBO, as shown in Fig. 8c. For data set 4, the forecasting results of eleven independent runs are tabulated in Table VI. The proposed forecasting model demonstrates the lowest performance errors (MAE 1.03 MW, RMSE 1.61 MW and MAPE 3.00%) as compared to LSTM-VMD-MBO and CNN-GRU-NN. Surprisingly, the CNN-GRU-NN model has achieved the second-best prediction model to improve the forecast accuracy of wind power generation shown in Fig. 8d. The proposed model has shown 62% and 41% higher accuracy in MAE than LSTM-VMD-MBO and CNN-GRU-NN, respectively. In terms of RMSE, the proposed model achieves 49% and 46% higher accuracy than LSTM-VMD-MBO and CNN-GRU-NN, respectively. In terms of MAE, the forecasting accuracy of the proposed model is 66% and 24% higher compared to VMD and CNN-GRU-NN, respectively. From the above analysis, we cannot definitively identify the best-performing model. Therefore, in the next section, statistical analysis is carried out to determine the best forecasting model.

G. Statistical analysis

Two statistical tests are carried out: Wilcoxon signed-rank test and Friedman’s ranking test. In these tests, the performance of forecasting models can be effectively distinguished from one another. The aim of the statistical tests is to ensure data-driven support from multiple independent measurements for selecting the best forecasting model by comparing with one another models. In Wilcoxon signed-rank test, two independent models are compared for different measurement values while the Friedman test, an extension of the Wilcoxon test, confirms the position of
each model based on data-driven analysis. In the Wilcoxon test, if p-values are less than 0.05, then the proposed model has statistical significance over other models that can be represented as “+” in the decision (Dec). The “−” sign refers to the inferior model and the “≈” sign indicates no significant difference between the models compared. The maximum and minimum ranking values are 66 (6 × 11) and 0, respectively. The outcome of the Wilcoxon test for data set 1 is shown in Table VII which indicates that the proposed model has no statistical significance over the LSTM-VMD-MBO model, but it has significance over the CNN-GRU-NN model for all the performance indices: MAE, RMSE, and MAPE. The proposed model has shown inferior results in terms of MAE, RMSE and MAPE compared to the LSTM-VMD-MBO model, although the outcome shows statistically significant improvement over CNN-GRU-NN in data set 2 presented in Table VIII. The results of the Wilcoxon test for data set 3 are demonstrated in Table IX. In this case, the proposed model achieves statistically significant improvement compared to both LSTM-VMD-MBO and CNN-GRU-NN. Applying the Wilcoxon test to data set 4, the proposed model achieves statistically significant improvement compared to LSTM-VMD-MBO and CNN-GRU-NN.

To further confirm the relative performance of the studied forecasting models, Friedman’s ranking test was applied. This test ranks the three compared models on a scale of 1 – 3, where 1 corresponds to best and 3 to worst. The ranking may be a fractional number, depending on the outcome of the independent runs. The ranking for data set 1 is depicted in Fig. 9a in which the proposed model achieves the lowest value, indicating superiority. The proposed model shows equal ranking in the RMSE performance index. In the analysis of data set 2 shown in Fig. 9b, the LSTM-VMD-MBO model achieves better ranking values compared to both the proposed model and CNN-GRU-NN. For data set 3, shown in Fig. 9c, the proposed model achieves the best ranking. For data set 4, shown in Fig. 9d, the proposed model achieves the best ranking again, followed by CNN-GRU-NN. The LSTM-VMD-MBO model is found to have the worst performance.

![Fig. 9: Friedman’s ranking test for (a) data set 1 (b) data set 2 (c) data set 3 and (d) data set 4.](image-url)

### H. Computational Efficiency

The computational efficiency of the forecasting engines is performed on a Laptop computer with Intel (R) Core (TM) i7-1065G7 CPU @1.30 GHz 1.50 GHz with 16 GB RAM. The calculated training time is based on the average time of each iteration for the data set (8640 points) of Boco Rock Wind Farm for the three prediction methods. This computation time is presented as an indication of time consumption for training the method, but it has no effect on the performance of the prediction engine. The reason is that once the model is ready, it takes moments to forecast the next values. The computation time varies greatly depending on the data sources. If the data size is large, the computation time increases significantly even in days, weeks or months depending on computer configurations and other components. Table XI indicates the training time of the three models. The high efficiency is obtained by the proposed prediction method followed by CNN-GRU-NN and LSTM-VMD-MBO. Generally, the computation efficiency should be similar for the proposed model and LSTM-VMD-MBO, but the time required for training is around 72% higher than the CEMOLS. This is because the data set is decomposed into 10 IMFs resulting in ten forecasting engines trained to forecast each IMF requires more computational time than six forecasting engines (6 IMFs) used in the proposed model. These IMFs are determined based on the best outcome from the data decomposition techniques with the consideration of mode aliasing. Considering a single engine of CNN-GRU-NN, it was supposed to take the lowest computational time, but its time consumption burden is the second highest. This is because the number of epochs is considered 300 for the better prediction outcome, resulting high computational burden while the other techniques have used only 35 epochs.

### V. Discussion

The very short-term (5-min) power generation data of four wind farms were predicted to identify the best-performing forecasting model. The performance of the studied models was not consistent across all the data sets. In a single-run comparison, the proposed CEMOLS outperformed in the last two data sets (2 and 3) while LSTM-VMD-MBO outperformed in data sets 1 and 2. For comparison with sensitivity analysis, the proposed model has outperformed the compared models in three data sets out of four based on median values. The LSTM-VMD-MBO has earned the second-best-performing model by placing second in three data sets. The CNN-GRU-NN model was generally the worst-performing model. This indicates a vulnerability of data decomposition techniques, i.e., data decomposition techniques do not always produce a better model to accurately forecast time-series data, even after fine-tuning. From the analysis, it can be concluded that a single run output may not be trustworthy for better prediction accuracy. Therefore, the robustness of the forecasting outcome needs to be justified by multiple independent runs, sensitivity and statistical analysis.

### Table XI: Computational time of models.

<table>
<thead>
<tr>
<th>Models</th>
<th>Proposed</th>
<th>LSTM-VMD-MBO</th>
<th>CNN-GRU-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>1260</td>
<td>2170</td>
<td>1800</td>
</tr>
</tbody>
</table>


The prediction model can improve the forecasting accuracy but after months or years, the accuracy may reduce this is especially true with the global weather changes every year. For example, the summer or winter season is not realized on time rather it is penetrating one another. The forecast model will work in this case, but its best performance is not guaranteed. Therefore, it may be useful to train the model again with new data to investigate whether any improvement in predicting wind power generation can be observed as compared to the existing one. The training of the model with new data may or may not improve the accuracy of the model, but it is an advisory for achieving the best practice to improve forecasting accuracy. To evaluate the models’ performance, the prediction and true values must be monitored over a long period of time, especially using plotting their performance index to decide which model is performing well. The new model must robustly outperform to replace the existing one. In updating the model, the model structure should be fixed and only the weights and hyperparameters need to be adjusted using the framework proposed in this study with new data.

It is well-known that deep-learning-based forecasting models can take many hours, days or weeks to prepare as a prediction engine depending on the computational speed of a computer and the data set burden. In such a situation, the model can be kept fit on the train data set without using the test set. This indicates a trade-off between the model testing and the computational cost of fitting a new model. The forecasting model should be selected based on the median architecture and therefore the modal may not provide high accuracy. To decide which model has a more stable set of prediction values, the sensitivity analysis must be monitored over a long period of time, especially using plotting their performance index to decide which model is performing well. The new model must robustly outperform to replace the existing one. In updating the model, the model structure should be fixed and only the weights and hyperparameters need to be adjusted using the framework proposed in this study with new data.

The data decomposition techniques have a limitation to decompose time series data. It cannot decompose the data if data points cross a certain number such as 10,000. Therefore, all the existing literature related to data decomposition techniques has used data points less than 10,000 either for short- or long-term prediction shown in Table XI. To implement in practice year-round, we may require training for short- or long-term prediction shown in Table XI. To determine the effectiveness of the model in addition to the benchmark Persistence or Naive forecasting model. The LSTM-VMD-MBO model is also processed in the same way as the proposed model to ensure a fair comparison.

From the analysis, it is evident that the proposed CEMOLS did not always achieve the best performance. This is in keeping with the ‘No Free Lunch Theorem’, which states that no single model can always perform better than any other model without understanding the modeling problem deeply. The proposed model produced higher errors compared to the LSTM-VMD-MBO model for the Bodongara wind farm (data set 2), possibly because of the large, very sharp transients in the real power output. Overall, however, we can conclude that the proposed CEMOLS with low computation time has consistently performed well compared to other models, improving forecasting errors on average by 41% in MAE, 46.21% in RMSE and 33.71% in MAPE compared to the CNN-GRU-NN model. Moreover, it is observed from the result section that sensitivity and statistical analysis needs to be carried out to determine the robustness of the proposed model for practical applications.

In future, CEMOLS will be tested at the Hydrogen DC microgrid lab at Griffith University in Australia after conducting several case studies.

Acknowledgement
The authors acknowledge financial support from the Blue Economy Cooperative Research Centre, established and supported under the Australian Government’s CRC Program, grant number CRC-20180101. The CRC Program supports industry-led collaborations between industry, researchers and the community.

References


