Very Short-term Forecasting of Wind Power Generation using Hybrid Deep Learning Model

Md Alamgir Hossain [a,b], Ripon K. Chakrabortty [a], Sondoss Elsawah [a], Michael J. Ryan [a]

[a] *Capability Systems Centre, School of Eng. & IT, UNSW, Canberra, Australia*
[b] *Department of EEE, Dhaka University of Engineering & Technology (DUET), Gazipur, Bangladesh*

Abstract

Accurate forecasting of wind power generation plays a key role in improving the operation and management of a power system network and thereby its reliability and security. However, predicting wind power is complex due to the existence of high non-linearity in wind speed that eventually relies on prevailing weather conditions. In this paper, a novel hybrid deep learning model is proposed to improve the prediction accuracy of very short-term wind power generation for the Bodangora Wind Farm located in the New South Wales, Australia. The hybrid model consists of convolutional layers, gated recurrent unit (GRU) layers and a fully connected neural network. The convolutional layers have the ability to automatically learn complex features from raw data while the GRU layers are capable of directly learning multiple parallel sequences of input data. The data sets of five-minute intervals from the wind farm are used in case studies to demonstrate the effectiveness of the proposed model against other advanced existing models, including long short-term memory (LSTM), GRU, autoregressive integrated moving average (ARIMA) and support vector machine (SVM), which are tuned to optimise outcome. It is observed that the hybrid deep learning model exhibits superior performance over other forecasting models to improve the accuracy of wind power forecasting, numerically, up to 1.59 per cent in mean absolute error, 3.73 per cent in root mean square error and 8.13 per cent in mean absolute percentage error.

Keywords: wind power forecasting; short-term prediction; hybrid deep learning; wind farm; long short term memory; gated recurrent network and convolutional layers

Nomenclature

NSW	New South Wales	CNN	Convolutional neural network
LSTM	Long short-term memory	AGRU	Attention-based gated recurrent unit
NN	Feed-forward neural network	MAE	Mean absolute error
RNN	Recurrent neural network	RMSE	Root mean square error
Bi-LSTM	Bidirectional LSTM	MAPE	Mean absolute percentage error
NEM	National energy market	MAE-TR	MAE during training
BPTT	Back-propagation through time	RMSE-TR	RMSE during training
SVM	Support vector machine	MAPE-TR	MAPE during training

Email addresses: Md.Hossain6@unsw.edu.au or alamgir_ duet@hotmail.com (Md Alamgir Hossain), r.chakrabortty@adfa.edu.au (Ripon K. Chakrabortty), s.elsawah@adfa.edu.au (Sondoss Elsawah), m.ryan@adfa.edu.au (Michael J. Ryan)

MAE-Tst	MAE during testing	h_t	Memory unit at time t
RMSE-Tst	RMSE during testing	b	Bias term
x_t	Input data at time t	c_t	Unit state
W	Weight matrix	y	Actual value
U	Input-hidden weight matrix	$\hat{y}$	Predicted value
V	Hidden-output weight matrix	S_w	Window size
ϕ	Activation function	f_h	Forecasting horizon

1. Introduction

The installation of wind power generators has sharply increased around the globe to minimise greenhouse gas emission, fulfil power demand and reduce electricity price [1]. Their penetration into distribution networks not only becomes essential to maintain a secure power supply by taking appropriate measure but also maintains sustainable development of a country [2]. In 2018, the total wind power capacity of onshore and offshore around the globe has reached 564 GW from 7.5 GW in 1997, which is 24% of total renewable energy sources and 6.5% of the total power generation. In Australia, the power generated from wind generators in 2018 was 5.76 GW, 3.38 (1.70 GW) times higher than power generation in 2009. This wind power generation is 23.58% of the total renewable energy produced (24.43 GW) and 7.1% of Australia's total electricity generation [3]. It is expected that the wind power generation will continue to increase to fulfil the energy target from renewable energy sources that is set at 50% of total power generation by 2030. However, the massive penetration of wind power generation into a distribution network may cause serious disruption in power supply reliability, security and economy if appropriate measures, such as accurate power predictions and proper operational strategies, are not in place [4]. The accurate forecasting of power generation is a challenging task due to the non-linear behaviour of wind speed that has high rates of changes with no typical patterns and heavily depends on extant atmospheric temperature and pressure. This non-linear behaviour of wind speed makes it harder to extract features and accurately forecast wind power generation to ensure a reliable, secure power system operation by scheduling in advance.

There are a number of forecasting methods to predict wind power generation, including physical models, statistical models, artificial neural network (ANN) models and hybrid intelligent methods [5]. The physical approach, such as numerical weather prediction, that considers initial values and boundary conditions uses hydro- and thermo-dynamic models of the physics and atmosphere, leading to poor performance in forecasting wind power generation due to the development of exact mathematical models [6]. A statistical approach, such as probabilistic auto-regression and probability mass bias, builds a relationship between wind power generation and explanatory variables to forecast power generation [7, 8]. This approach has an issue with adaptability and learning capability, and its performance decreases with the increase in prediction horizons. ANN to predict wind power generation is widely applied due to its ability to map nonlinear relationships and adopt self-learning from data samples. The main advantage of this technique is that it does not require any mathematical model for building a relationship between input and output data to forecast wind power generation [9, 10]. An NN-based tool to forecast wind power density of ten minute data is developed in [11]. Although an ANN has demonstrated good performance in forecasting wind power generation, it has an issue with over-fitting and under-fitting the model. ANN is also very inefficient to update the weights and biases for more hidden layers [12, 13]. To overcome such limitations, a spatio-temporal framework using a support vector machine (SVM) algorithm to predict short-term wind power forecasting is presented in [14]. A grey wolf optimizer is used to tune the parameter of the kernel function in the SVM model. The SVM algorithm suffers from slow training and poor generalisation ability [15]. In a hybrid intelligent method, two approaches are generally combined to better predict the wind power generation, such as ANN and fuzzy logic methods [16]. This combination is useful to improve the prediction accuracy by compensating the disadvantage of each method. The main concern with the fuzzy logic approach is that it requires an empirical variable for setting the explanatory variables [17]. In [18], a combined technique of SVM and improved dragonfly algorithm is presented to forecast wind power generation. The dragonfly algorithm improved by using adaptive learning factor and differential evolution approach is used for selecting the optimal parameters of SVM.

Deep learning models have recently drawn attention for their potential to improve the forecasting accuracy of wind power generation. Several forecasting models, such as auto-encoders [19], restricted Boltzmann machine [20], convolutional neural network (CNN) [21] and long short-term memory (LSTM) [22], have been used to forecast wind power generations. Due to the ability to overcome the issues of conventional neural networks, such as over-fitting, higher time in training, slow convergence and uncertainty, deep learning models are getting wide attention among researchers [23]. In [24], an LSTM network is applied to forecast wind speed, where the maximal information coefficient is applied for auto correlation of wind speed series. A two-dimensional CNN to forecast short-term and long-term wind power generation is presented in [25]. The wavelet transform approach is used for decomposing the data and particle swarm optimisation algorithm is used for tuning the weights of the CNN to increase the prediction accuracy of the model. An attention-based Gated Recurrent Unit (AGRU) to predict short-term wind power generation is presented in [26], where an attention mechanism to recognise the essential input variables is developed as a feature selection approach. The LSTM algorithm to forecast short-term wind power generation is used in [27], where a gradient descent approach is used to train the model. In [28], an ultra short-term probability prediction method that consists of LSTM, wavelet decomposition and principal component analysis is demonstrated. A normal distribution model is developed to interpret the uncertain error in prediction. An LSTM algorithm with a two-stage attention mechanism that is used to weight the input feature and strengthen the trend of wind power feature is presented in [29]. In [30], a hybrid model of Bayesian averaging and Ensemble learning to forecast short-term wind power is presented. A combined approach of wavelet transform and CNN for probabilistic wind power prediction is used in [21], where the wavelet transform decomposes the data into various frequencies. An improved LSTM enhanced forget-gate network to predict wind power generation is presented in [31]. The LSTM network enhances the performance by adding two peepholes, using softsign activation function instead of tanh, eliminating input-gate from the LSTM architecture, subtracting the output of the forget-gate by the fully 1 matrix and using the outcome as the input value of the data update. This network improves the effect of forget-gate and accelerates the convergence process.

The majority of the existing literature deals with either short-term or long term wind generation forecasting, which is quite different from real wind farm operations. As a wind farm participates in a spot electricity market [32], it requires to submit power generation capability of the next $5 - 15$ minutes (very-short term) to meet the positive or negative ramp of the power demand in a network to balance instantaneous power supply and demand with minimum reserve capacity. Through the real-time energy dispatch (5–15 min), the reliability and stability of a power system network are ensured. As real-time dispatch needs to be matched as precisely as possible for power system security reason, the accuracy of the very short wind power generation is a crucial aspect of forecasting. The forecasting has huge impact on the optimal operation of security reserve and wind farm control in participating the spot electricity market [33]. In addition, wind farms are currently operated in real-time operation to avoid high uncertainty in power generation, which requires to send commands every 5 or 15 minute to distributed energy resources, including wind generators' power dispatch and energy storage system, to maintain reliable, secure power supply in the network [2]. Moreover, this forecasting assists to understand the dynamic behaviour of wind turbine, optimise performance, control frequency and voltage, and monitor online health of the wind farm [34]. However, the very short-term forecasting is more challenging than short- and long-term wind generation forecasting. This is because of capturing high uncertainty in very short-term data intervals, leading to higher non-linear behaviour exist in wind power forecasting.

In [35], the ultra-short-term (10 min) wind power generation is predicted using Extreme Learning Machine (ELM) whose weights and bias of the single hidden neural network are optimised using the Salp Swarm Algorithm. The input data compose of a multi-input sample set (wind speed, temperature, wind direction and climate factors) to predict wind power generation. This method suffers from over-fitting issue for many parameters and its performance deteriorates in the present of outliers. A probabilistic forecasting approach (non-parametric equation-free) based on the empirical dynamic modeling is developed in [36] to estimate the forecasting uncertainty. The approach is equation and distribution free to describe the power generation and therefore the performance of this approach solely depends on the potential interactions hidden in the time series data of the state variables. In [37], a combination of deep learning and ensemble learning is developed to forecast ultra short term (10 min) wind speed. The wind speed series is decomposed using the wavelet transform and feature is extracted using the deep belief network. In [38], a 15-min wind speed forecasting using the gated recurrent unit is presented in which the GRU method demonstrates a better result than the LSTM model. The research deals with the wind speed prediction that needs to be converted to wind power forecasting, leading to higher inaccuracy to wind power generation.

The above research has a number of limitations, including a number of forecasting variables that needs to be forecasted, conversion error from wind speed prediction to wind power generation, and short-and-long term prediction trends that, unable to participate in a spot market and real-time dispatch, are much easier than very short term forecasting. To overcome the limitations with higher accuracy of forecasting, this paper develops a novel time series forecasting model of very short term (5 min) wind power generation in the Bodangora wind farm, Australia. The novel model consists of convolution layers, gated recurrent units and neural network. In the model, convolution layers have been used to extract the high dimensional feature of complex data set, then the features are used to train the gated recurrent units for capturing long term memory of essential feature. Finally, neural network develops a relationship in between output and input data set processed by previous layers. In addition, this study collects five minute raw data from the real wind farm and pre-processes the data for training/testing a forecasting model. We improves the performance of the forecasting model through explicitly analysing several impacts on it and hyperparameter tuning of the model using a grid search technique. The main contributions of the paper are summarised as follows:

- This paper develops a novel hybrid deep learning model to improve the accuracy of wind power generation for the Bodangora wind farm in the state of New South Wales (NSW), Australia. The algorithm consists of convolutional layers, GRU layers and a fully connected neural network. The convolutional layer extracts the feature of five-minute data while the GRU retains the necessary information to improve the forecasting accuracy.

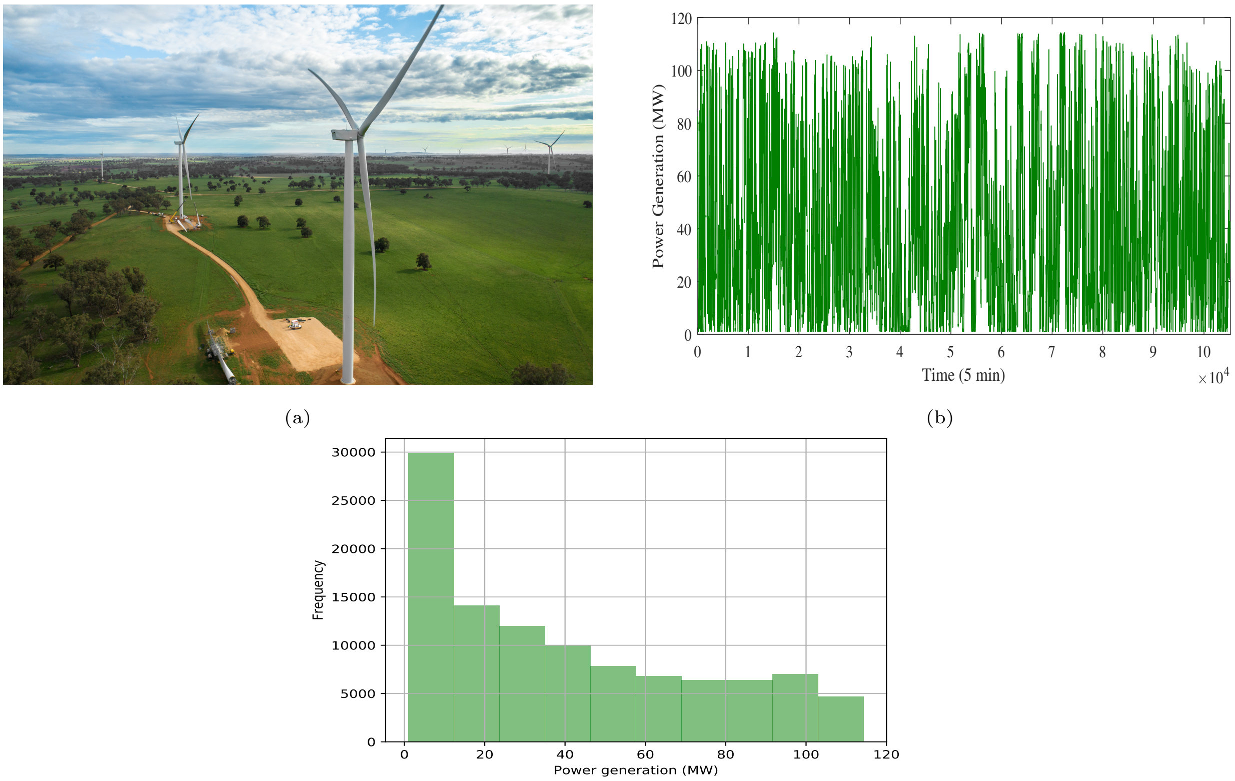
- Real five-minute data of the wind farm that has been pre-processed for machine learning algorithms are utilised to evaluate the effectiveness of the forecasting model.
- A comparative analysis with advanced forecasting methods, such as LSTM, GRU, ARIMA and SVM, is carried out to prove the efficacy of the proposed hybrid model. Sensitive and statistical analysis are also conducted to demonstrate its effectiveness.

The remainder of this paper is structured as follows. In Section 2, a description of the Bodangora wind farm in the NSW state of Australia is presented. Section 3 demonstrates the design and implementation procedures of the forecasting models, including CNN, GRU and NN architectures, with the evaluation methods used for measuring their performance. Experimental results for forecasting wind power generation of the wind farm with in-depth analysis are demonstrated in Section 4. Section 5 provides concluding remarks.

2. Description of the Bodangora wind farm

The Bodangora wind farm, situated on the hills surrounding Mount Bodangora in the district of Bodangora near Wellington in Central West of NSW, has thirty-three wind turbines that produce a total wind power generation of 113.2 MW. Each of the wind turbines can produce power around 3.4 kW. The hub height and rotor diameter of the turbines are 85 m and 130 m, respectively. The annual output from this farm is 361 GWh that is sufficient to supply approximately 49,000 NSW homes per year. The tower and rotor diameter of wind generators are 85 metres and 130 metres, respectively. Around 60% of the generated power from this farm is supplied to EnergyAustralia and the rest (40%) is actively managed within Infigen's energy markets risk framework [39]. The power is transmitted to the grid utility through the high voltage connection of 132 kV.

The wind power generation of the wind farm in every five minutes shown in Figure 1 is predicted in this paper. The one year of data (2019) of the wind farm and their distribution using histogram are shown in Figure 1b and 1c, respectively. The major frequency of the power generation in the wind farm is below 50 MW over a year and the frequency remains constant after rest of the ratings, i.e., 50–114 MW. The data are collected from the national energy market (NEM) Australia and then the data are pre-processed to train forecasting models. In processing the data, there are some negative values and zeros in the raw data, but these data are not prepared to train the forecasting model because the model has an issue to create a pattern between input and output data due to the existence of zero values in training or testing data. In addition, the zero value cannot be used in MAPE calculation due to the fact that a value divided by zero makes infinite. The negative data of the wind farm collected indicate the power consumption to maintain its power supply when power generation of the farm is not sufficient to meet its power demand. Power generation from wind generators not only depends on the wind speed (≤ 3) but also depends on the Australian energy market. Sometimes, the electricity prices become negative when power generation is much higher than power demand. In this case, if a wind farm supplies power to the electricity market, then the wind farm has to pay the electricity fees. As a result, power generation from the wind farm is shut down to avoid penalty while supplying generated power.



(a)

(b)

(c)

Figure 1: The Bodangora wind farm in NSW: (a) Pictorial view (b) Power generation in 2019 and (c) Histogram of power generation

.

Generally, the power generated by wind generators is a piece-wise function of the wind speed and it can be expressed as follows [40]:

$$P_w = \begin{cases} 0 & \text{if } v_f \leq v \quad \text{or} \quad v \leq v_c \\ P_r * \frac{v^3 - v_c^3}{v_r^3 - v_c^3} & \text{if } v_c \leq v \leq v_r \\ P_r & \text{if } v_r \leq v \leq v_f \end{cases} \tag{1}$$

where P_r and v_r are the rated electrical power and rated wind speed, respectively, and v, v_c and v_f represent wind speed, cut-in wind speed and cut-off wind speed, respectively. Power generation begins at the v_c (> 3 m/s) and stops at the v_f (≥ 25 m/s). The output power increases non-linearly in between v_c and v_r as shown in Figure 2 and remains to its rated generation until wind speed reaches v_f. The wind generator does not produce any power after the cut-off wind speed due to safety reasons.

From Eq. (1), the power generation of a wind farm can be easily determined through the measurement of wind speed; therefore, many researchers focus on forecasting wind speed to predict wind power generation. While this is an acceptable way to forecast power generation, prediction errors may increase due to several reasons, including errors in wind speed and power conversion errors from the mathematical model. Therefore, this study has selected the direct wind power forecasting rather than wind speed.

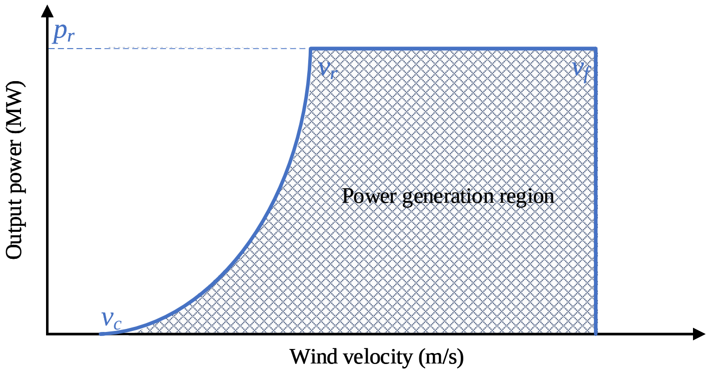


Figure 2: Power curve of wind generator.

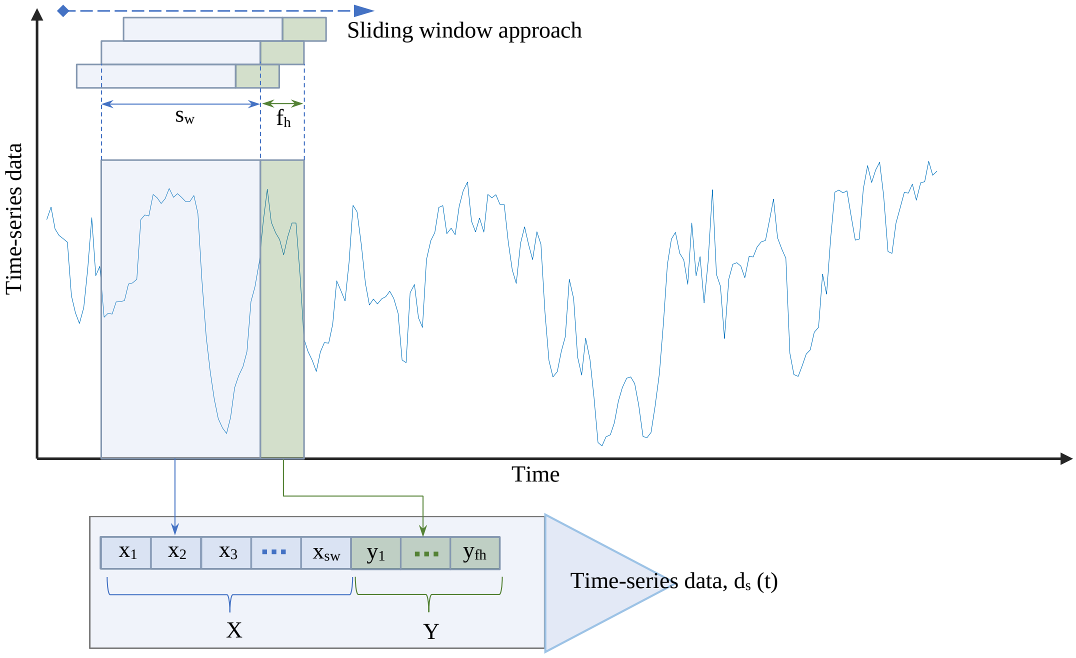


Figure 3: A sliding window approach to prepare data for machine learning.

3. Design and implementation of forecasting model

3.1. Problem formulation

This paper focuses on predicting wind power generation through a time-series approach as conventional ways of forecasting wind power generation have many disadvantages. For example, as forecasting the wind generation also has a direct relationship with the wind speed, wind direction, air temperature and air pressure data, their future data need to be estimated by other prediction methods, leading to more prediction error than the time-series technique [41]. Therefore, time-series methods that do not require any prior knowledge of wind speed, air temperature and air pressure forecast the wind power generation only by analysing the pattern of the past data. The time-series approach ($X = x_1, x_2, ..., x_t$) is a sequence of observations in a time frame, where x_i refers to the observation at time t and X defines the total number of observation. Input-output data sets from time-series data are prepared using a sliding window approach depicted in Figure 3 and 4. The time-series data are framed within the window size as input data and forecasting horizon as output data. The window is

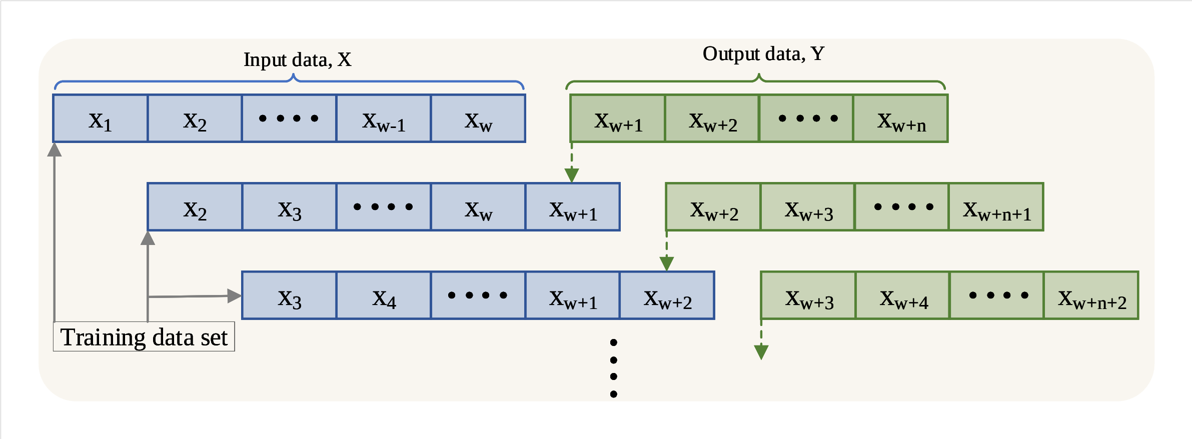


Figure 4: Input-output data set for training the network.

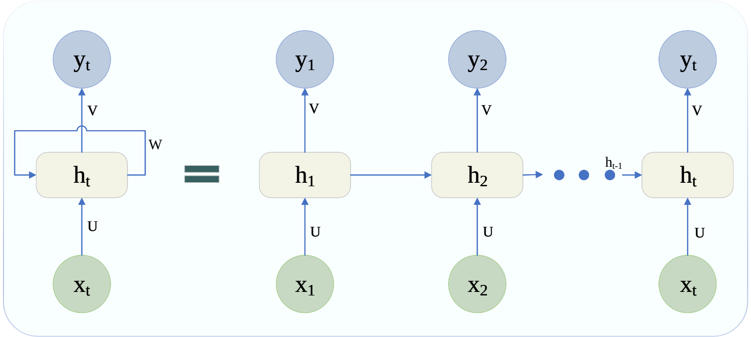


Figure 5: Architecture of a simple RNN network.

moved forward with shifting one time step as shown in Figure 4 to prepare data set for training and testing purposes.

3.2. Hybrid deep learning approach

The accurate prediction of wind power generation is a challenging task due to the existence of high uncertainty in wind speed that eventually depends on fluctuating weather conditions. Although wind speed is highly non-linear in characteristics, it follows a certain pattern over the time period that can be extracted for accurate power forecasting. Among several forecasting methods, deep learning models generally demonstrate better performance to forecast wind power generation due to their high performance in mapping nonlinear complex models [42]. The advanced recurrent neural networks (ARNN), such as LSTM and gated recurrent unit (GRU), can be effectively utilised to capture long and short-term dependencies of time-series data. This study has utilised the advantages of GRU and CNN to build a novel hybrid deep learning model shown in Figure 6. The proposed hybrid deep learning method is a data-driven approach to capture high nonlinear behaviour in the time-series data of the wind farm.

3.2.1. Simple recurrent neural network (RNN)

Artificial neural networks (ANNs) based on a collection of connected units that loosely model the neurons in a biological brain are widely utilised for either modelling a complex system or predicting a system's behaviour. These are performed by training the networks first using input and output data sets, leading to a relationship between them. Then, the networks trained are used as a model to predict outputs, provided the input data. In

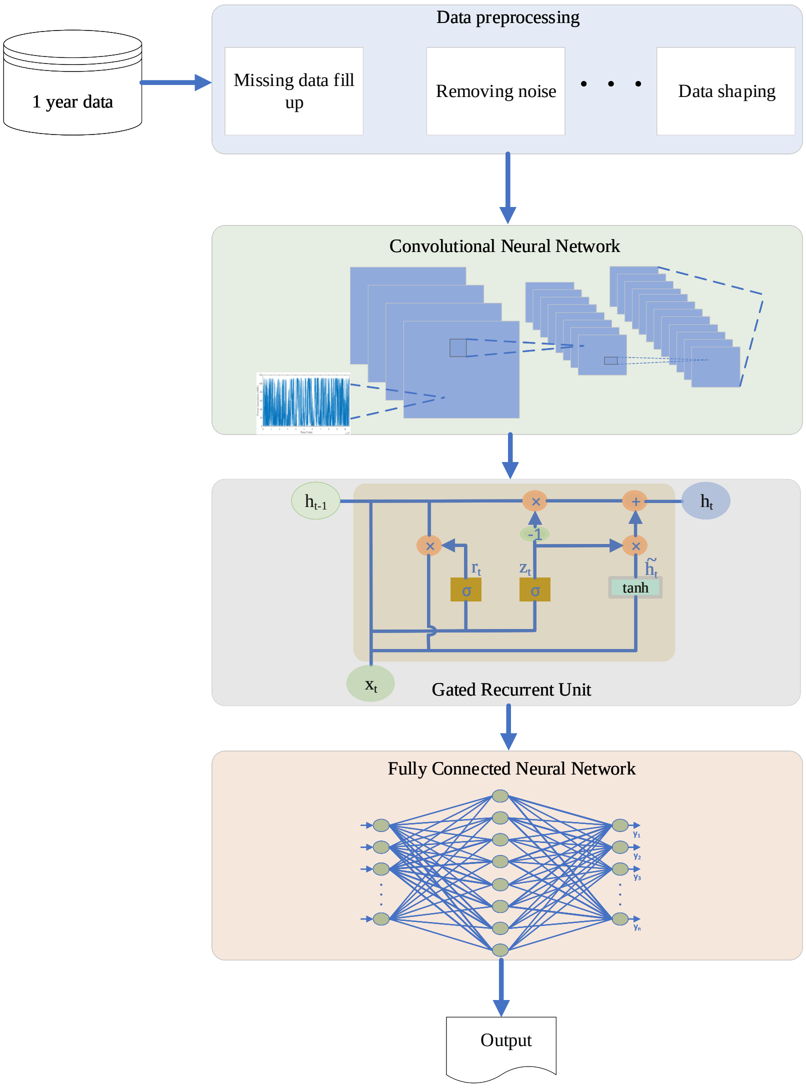


Figure 6: Proposed hybrid deep learning model for wind power forecasting.

the training, the networks update their weights (a real number) to learn the relationship as accurate as possible. Generally, a backpropagation algorithm that propagates errors from the output to input until errors decline a threshold value is used to train the networks. Although ANN models have been used widely, they are unable to handle historical data dependencies [43].

To effectively manage the historical data sequences, such as time-series, simple recurrent neural networks (RNNs) are constructed in [44]. The networks deal with the sequence of data by persisting the knowledge collected from subsequent timestamps by means of a recurrent function (network loops) as shown in Figure 5. These feedback loops give the connected units memory to incorporating the previous state as input. The computational procedure of an RNN can be mathematically represented as follows:

$$a_t = W^T h_{t-1} + U^T x_t \tag{2}$$

$$h_t = \phi(a_t) \tag{3}$$

$$y_t = \sigma(V^T h_t) \tag{4}$$

where U, V, W are the weight matrices for input-hidden, hidden-output and hidden-hidden connection, respec-

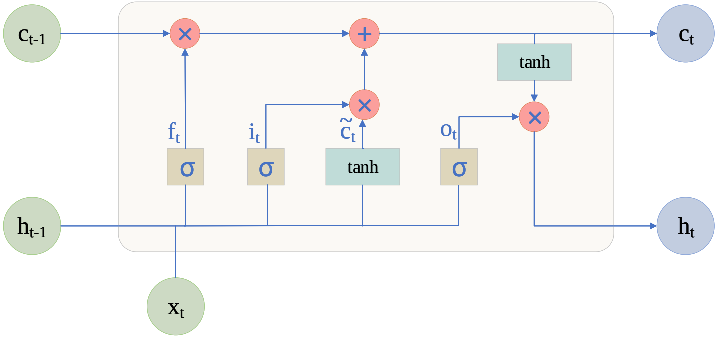


Figure 7: Architecture of LSTM network.

tively, ϕ is an activation function (normally a hyperbolic tangent function). x_t refers to the input at time step t, y_t denotes the output at time t, h_t is the memory unit of RNN at time step t. The current state of h_t is calculated from the current input x_t and previous hidden state of h_{t-1}.

3.2.2. Long short-term memory (LSTM)

Although RNNs are effective in dealing with short-term sequences, they cannot handle long-term sequences. This is because of the vanishing or exploding gradients problem, where the propagation of gradient generally fails in long-term dependencies [45]. This problem can be overcome by the use of LSTM networks that maintain the same structure of RNN with different forms of inner cells. The primary idea of LSTM networks is to utilise particular neurons to store and transfer information over a long period to (i) acquire permanent memories, (ii) abate the rate of information destruction and (iii) catch long-term dependencies [46]. Figure 7 shows the structure of an LSTM memory block with one cell. The LSTM networks facilitate the above features by executing a gated system that regulates the flow of information, i.e. the gradient flow, to handle the vanishing gradient problem. The network can retain the memory for the long term by controlling the gate system, where an internal memory is unchanged for a long time. There are four types of gate mechanisms used in the LSTM network: input, forget, update and output.

- **The input gate:** It has two parts, i_t and $\tilde{c}_t$. The former determines the input value through the sigmoidal function while the latter makes an input value through the 'tanh' function that is added later to calculate the unit state, c_t. The mathematical formulation can be expressed as follows:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{5}$$

$$\tilde{c}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{6}$$

where W_i and W_c are the weight matrices of the i_t and $\tilde{c}_t$, respectively; and b_i and b_c are the bias terms of input gate. The square brackets refer to the addition of the two vectors.

- **The forget gate:** This gate determines which information should be stored or forgotten by sending a value within 0 and 1. The value one indicates all reserved state while zero refers to the all forgotten states. The gate can read the output of the previous layer and the input of current state to calculate value as follows:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{7}$$

where W_f is the weight matrix of the gate layer used for storing or deleting information, σ is the sigmoidal function, and b_f is the bias term of the gate layer.

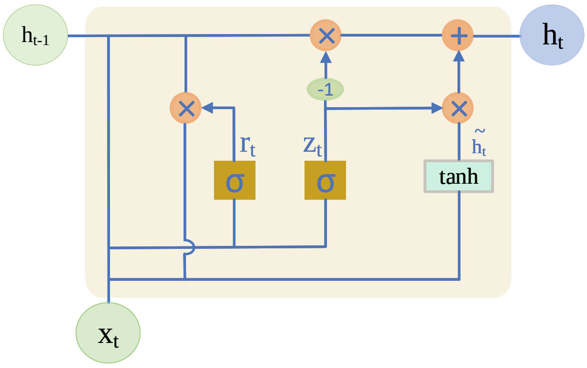


Figure 8: Architecture of GRU network.

- **The update gate:** It updates the state of old cell. The value of the cell state that encodes the - so far learned- information from the input sequence is the sum of the state products of forget value with the previous time state and the state of two input values as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{8}$$

where $\odot$ is the Hadamard product of two states.

- **The output gate:** This gate gives a filter version of the cell state as a network's output and determines the information after multiplying the input values by the sigmoidal function, then the value is again multiplied by the tanh function to obtain the final output of the gate as follows:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t \odot \tanh(c_t) \tag{10}$$

where W_o and b_o are the weight matrix of the output gate layer and the bias term of the output gate layer, respectively.

3.2.3. Gated recurrent unit (GRU)

The gated recurrent unit (GRU), a simplified version of the LSTM presented, is a gating mechanism of recurrent network, introduced by Kyunghyun Cho in 2014 [47]. The gates are just neural networks that control information flow passing through sequence chain. The GRU also solve the issue of the vanishing gradient problem like the LSTM model but with fewer parameters. In the GRU architecture, only two gates, a reset gate and update gate, as shown in Figure 8 are used to transfer information as compared to the LSTM architecture which uses four gates. The update gate decides what information is essential to keep and what information need to be deleted similar to the activities of the forget and input gates in the LSTM architecture. The reset gate is utilised to determine how much information need to be forgotten. The mathematical formulation can be represented as follows:

$$z_t = \sigma(W_z.[h_{t-1}, x_t] + b_z) \tag{11}$$

$$r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \tag{12}$$

$$\tilde{h_t} = \tanh(W_h.[h_{t-1} \odot r_t, x_t] + b_h) \tag{13}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \tag{14}$$

where z_t is the update gate at time step t, r_t is the reset gate, $\tilde{h_t}$ is the memory content for using the reset gate to store the relevant information from the previous information.

3.2.4. Convolutional Neural Network

The Convolutional Neural Network (CNN), a family member of deep learning algorithms, is primarily used for two or three-dimensional data of grid-like topologies, such as visual imagery, but it can also be used for one-dimensional data, such as univariate time-series data. Comprising a convolutional layer, pooling and fully-connected layers, it has the ability to capture the same patterns of data and therefore it can be successfully used at complex data format. Due to the effective solution approaches, it has been adopted for various applications, including computer vision tasks, such as image recognition and sequence modelling tasks: time-series load forecasting, machine translation and audio generation. The convolutional layer uses a mathematical convolution operator and cross-correlation instead of general matrix multiplication, and it provides extracted features of input data through sliding a kernel (filter) over the input data. The number of kernels calculate the number of output feature maps. Two hyper-parameters, stride and padding, are utilised to regulate other spatial features [48]. Stride is the difference between two consecutive input patches to indicate a direction of motion, and padding represents expanding the input by summing zeros to regulate the output size. A pooling layer, a max or an average, is used to decrease the dimension of data by mixing the output of neuron clusters at one layer into a single neural in the next layer. Maximum value from each of a neuron's cluster at the previous layer is used in a max-pooling, and mean value of a cluster is used in an average pooling. In fully connected layers, every neuron of one layer is connected to each neuron in another layer, similar to multi-layer perception neural network.

3.3. Huber loss function

A loss function determines the ability of a model to predict the true value by taking the forecasted output and real output. The output of the function indicates the suitability of the model developed, for example, the high output value refers to the poor performance of the model and the low value defines well performance in predicting the output. There are three common loss functions used in machine learning: MSE, MAE and Huber loss [49]. It is essential to use an appropriate loss function for training the model due to its impact on learning in a specific way. For example, the MSE emphasizes training a model well if there are outliers while the MAE is effective for ignoring the outliers during training the models.

The Huber loss function offers a balanced result that considers outliers in a moderate amount for building a robust model [49]. The function is a combination of both loss functions (MSE and MAE) as follows:

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

where δ, a hyperparameter, is a positive real number that controls the transition between MSE and MAE. Eq. (15) indicates that if the loss values are less than δ, it acts as MSE and uses the MAE when the loss values are higher than δ. This is the main reason of the function being robust to consider outliers.

3.4. Procedures of the proposed approach

The detail steps are proposed to predict the wind power generation as follows.

Stage 1: Data pre-processing

- Load and extract the one year data of wind power generation as shown in Figure 1b;
- Divide the data into two main components: time and series formats;
- Split the data into training and testing sets;
- Define the window size, batch size and shuffle buffer size;
- Prepare data of input (X) sets and output (Y) sets for training a model;

Stage 2: Develop forecasting model

- Use the convolutional layers to extract data feature from input data;
- Use the gated recurrent layers to memorise the important features;
- Use the fully connected NN layers to map the input-output relationship;
- Normalised the data by multiplying the maximum magnitude;

Stage 3: Training the model

- Use an optimisation algorithm for tuning parameters' weights;
- Compile the model with Huber loss function;
- Train the deep learning model of Stage 2 with input data until the maximum iteration is reached;

Stage 4: Forecasting and measuring performance

- Use the model trained from Stage 3 to forecast the entire data;
- Separate the prediction data into training and forecasting errors;
- Use the performance indices to measure the effectiveness of the model presented.

3.5. Performance index

The performance of the proposed model for the very short-term prediction has been measured by using three indices, the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE). These indices are briefly described as follows.

• **Mean Absolute Error (MAE):** The MAE refers to the average of the absolute errors between the actual and forecast values, describing a typical magnitude of errors. It considers absolute value to avoid cancellation between negative and positive values of errors and therefore it does not refer to under performance or over performance of the model. MAE is useful as a performance index to avoid the influence of outliers, extreme values in data sets. A small value of MAE indicates high accuracy in prediction, i.e., the zero value of MAE suggests the accurate model for predicting output data. The mathematical equation is represented as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} \mid \hat{y}_i - y_i \mid \tag{16}$$

where $\hat{y}_i$ is the predicted value of the power generation, y_i is the actual value of the power generation, and n is the total number of predicted points.

- **Root Mean Square Error (RMSE):** The RMSE indicates a sample standard deviation of prediction errors. It suggests the spread out of errors, i.e., the larger value indicates higher spread out of the error. The value of RMSE is usually higher than MSE due to the presence of outliers, i.e., the larger difference in between actual and forecast values has a higher value on RMSE than MSE. The following equation is used to calculate RMSE value:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)}. \tag{17}$$

- **Mean Absolute Percentage Error (MAPE):** The MAPE, the percentage equivalent of MAE, measures how far the forecasting values of the model are off from their actual values on average. As this method also uses absolute value, it is free from the influence of outliers like MAE. This method is useful for using percentages, i.e. forecasting values are scaled against the true value, to interpret the performance of the model as both MAE and RMSE show value from zero to positive infinity. Therefore, it may calculate unexpected value when the actual data are zero or very small. In addition, it can calculate a bias value due to the division of the actual value. The following formula is used:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \mid \frac{\hat{y}_i - y_i}{y_i} \mid \times 100\%. \tag{18}$$

The smaller are the values of MAE, RMSE and MAPE, the better is the forecasting performance.

4. Experimental results

This section highlights the forecasting results of the hybrid deep learning model and its effectiveness through extensive analysis to predict wind power generation. To begin with, the hyper-parameter tuning of the hybrid network is executed to determine the best prediction model. We have conducted grid search methods for selecting the best parameters among the possible parameter settings as they have a significant effect on the outcomes of the model. Then, the forecasting results of various models, such as LSTM, Bidirectional-LSTM (Bi-LSTM), GRU, ARIMA and SVM, are presented to compare their effectiveness in predicting wind power generation. The prediction models are implemented on a personal computer of central processing unit (CPU) configured by a 3.4-GHz Intel Core i7 processor with 16 GB RAM, using the Python 3.7 software that utilises the Keras API integrated with the TensorFlow framework.

4.1. Data preprocessing

As the data recorded by smart meters or sensors are taken from the real wind farm in the NSW state, they are subjected to noise, outliers and missing data. Communication failure is the main reason for missing data. Therefore, the data to prepare for machine learning are generally modified by the following procedure.

Missing values: The missing values of wind power generation are filled with the mean value of three previous and next time steps of missing data.

Data transformation: As the data recorded by capturing wind power generation are not compatible with analysis, data aggregation are first performed to make a suitable format. Then, data are transformed to scaling the differences between time-series using a linear transformation.

Data structure: To forecast the wind power generation of the wind farm for h next time steps, the data are divided into input and output formats by transforming the input time-series into input-output pairs using a moving window process as follows:

$$[x_{t+1}, x_{t+2}, ..., x_{t+h}] = f(x_t, x_{t-1}, ..., x_{t-w}) \tag{19}$$

where w is the window size of input data, x_t is the time-series data, h is the forecasting horizon and f is the deep learning model established by training.

Table 1: Data features of Bodangora wind farm in 2019.

Feature	Quantity	Feature	Quantity
Count	105121	50%	31.55
Mean	39.82	75%	66.00
Std	33.48	Max	114.36
Min	1.00	Data sample	5 min
25%	9.79		

4.2. Hyper-parameter tuning

This section demonstrates tuning of hybrid deep learning models by using a grid search method to achieve the best forecasting model of wind power generation. Although we have tuned all the parameters (such as neuron size, loss functions, activation functions, optimisation algorithms, batch size and window size) of the forecasting models, only three tuning results are displayed here to avoid redundant spaces: network selection, convolutional filter and kernel size selection. As hybrid deep learning models are stochastic, i.e., the same prediction model will have different performance for the same input data, the models are evaluated multiple times (eleven runs) to obtain a high performing configuration that is robust in prediction. To evaluate the parameters' performance, we have considered median value of the samples as the median value is located in the centre of all data, whereas the mean value is easily influenced by outlier values, leading to misguided information sometimes. Validation losses are used as a criterion for selecting the best configuration.

4.2.1. Number of layer selection in hybrid network

In the deep learning models, the number of neuron layers have an impact on improving the forecasting accuracy due to the over-fitting and under-fitting issue. It is not true in deep learning that an increasing number of layers will improve the prediction accuracy. This is because of over-fitting issues that indicate prediction models learn well during the training period, but it has higher errors at the prediction time. In contrast, a few numbers of layers may lead to under-fitting issues due to inability to map the relationship between input and output data. Therefore, five combinations of various layers tabulated in Table 2 are carried out to find out the best model for the wind power prediction of the Bodangora wind farm. In the table, M(0...4) represents model numbers from zero to four, CL indicates a convolutional layer, 2GRU refers to two GRU layers and 2NN defines two neural network layers.

Table 2: Selection of several network configurations.

Models	Structure	Models	Structure
M0	CL_GRU_2NN	M3	2CL_2GRU_2NN
M1	2CL_GRU_2NN	M4	2CL_3GRU_2NN
M2	CL_2GRU_2NN		

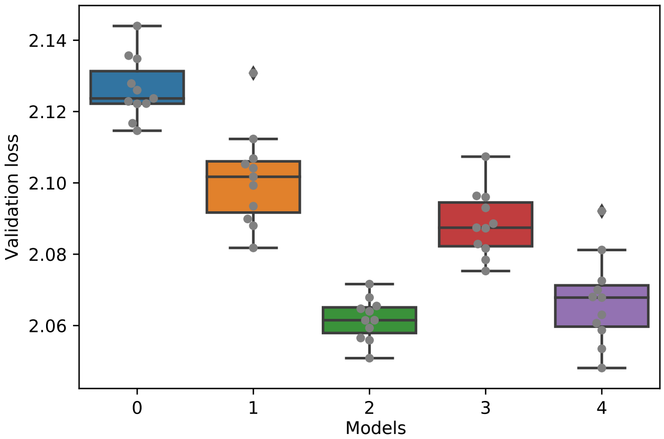
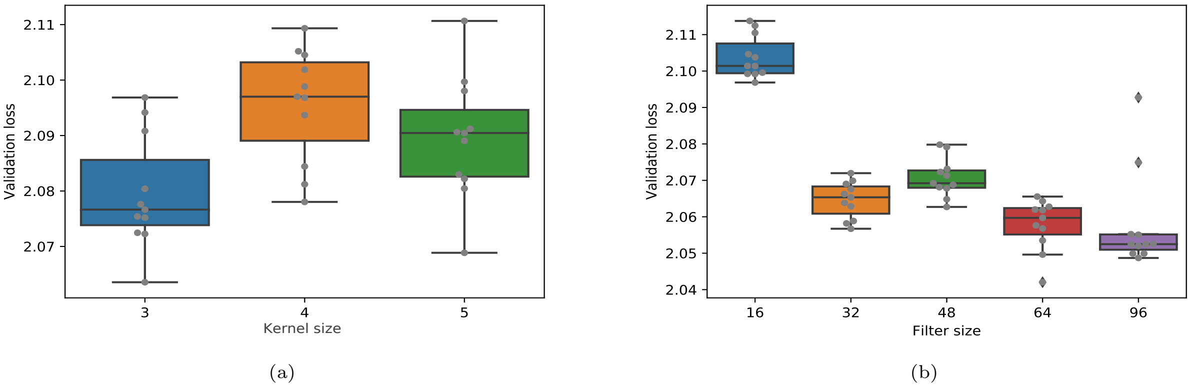


Figure 9: Layer selection of hybrid deep learning model.

Simulation is carried out multiple times to select the best possible combination of the hybrid layers while ensuring the robustness of the model developed. From Figure 9, it can be observed that the median values of all the combinations are swinging throughout the combination from zero to four. Initially, the value decreases in the first three combinations, then it increases in configuration four, after that it again decreases in the fifth configuration. The lower performance is found in model 1, followed by model 2. The superior performance is observed in model 2 which consists of a single convolutional layer, two GRU layers and two NN layers. Model 2 demonstrates a robust forecasting model with low deviation and no outliers. It should be noted that Model 4 has also shown a competitive performance but it has a bad outlier with high variance than Model 2. Therefore, Model 2 is selected for this study to improve the accuracy of the forecasting model. From the study, it can be concluded that an increasing number of layers may not improve the performance of the prediction model, but



(a)

(b)

Figure 10: Selection of CNN parameters: (a) kernel size and (b) filter size.

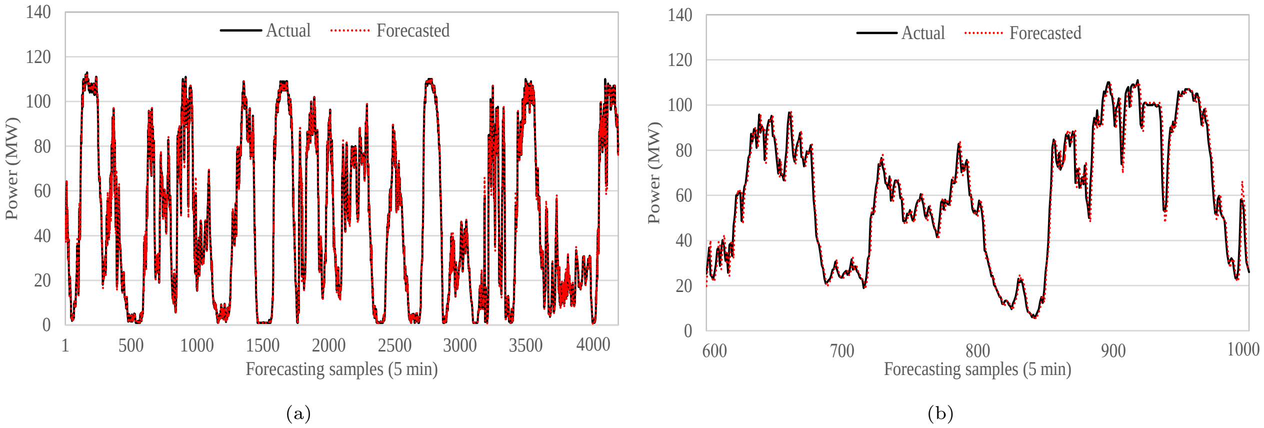


Figure 11: Forecasting wind power generation: (a) original and (b) zoom in.

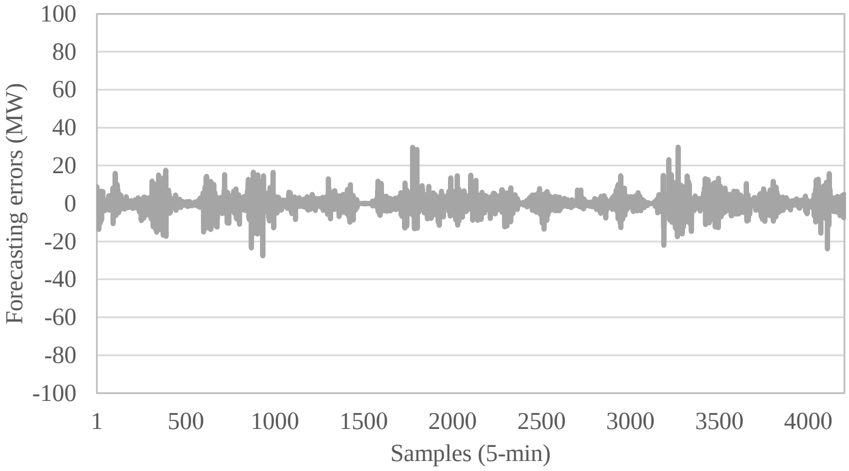


Figure 12: Residual of the hybrid deep learning model.

the less number of network layers can improve the performance. An effective combination of network layers for the hybrid deep learning model is essential to improve its performance.

4.2.2. Selection of parameters in convolutional layers

In convolutional layers, there are mainly two parameters (kernel and filter sizes) that need to be tuned to improve the performance of convolutional layers in a hybrid deep learning model. A kernel is an integral part of the architecture and is defined as an operator used in the whole data set to transform the information. Figure 10a demonstrates the performance of various types of kernel size and it can be observed that the best performance among eleven independent runs is in the selection of kernel 3, followed by kernel 5. It is found the majority of the loss values are in the lower portion of the box as shown by gray dot points in Figure 10a. The kernel size 4 shows a poor performance as compared to others. The filter size tuning of convolutional layers is shown in Figure 10b. It is observed that there is a swing around the median of filters although this has less variation. Selection of 64 and 96 filters has much better performance as compared to other filter sizes. Filter size 16 shows the worst performance followed by 48. It should be pointed out that the filter size 96 has shown a promising result with less deviation but two bad outliers, while the filter size 64 has a consistent result with a good outlier.

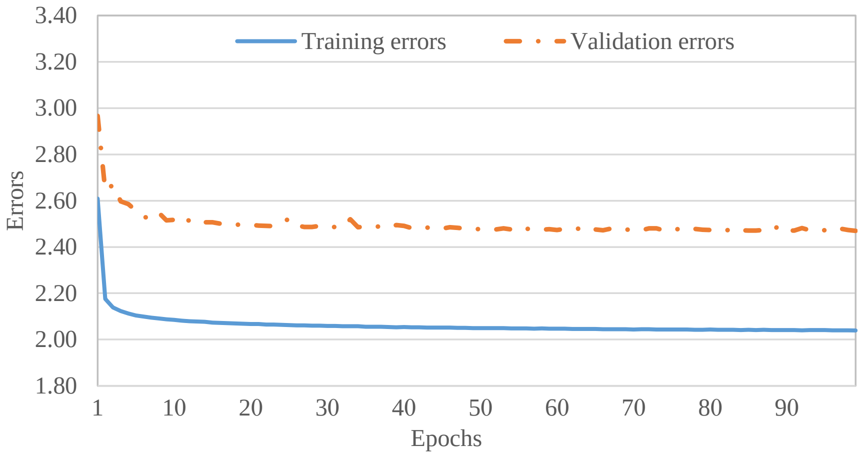


Figure 13: Convergence curve of the hybrid deep learning model.

4.3. Forecasting power generation

Wind power generation of the Bodangora wind farm described in Section 2 is predicted in this section as a single-step rolling forward prediction. One year data (105,121 samples) of the wind farm is collected for training and testing purposes, where 95% of data is used for training the forecasting models and only 5% of the data is employed to test the effectiveness of the models presented. As data samples are captured for very short term (5-min), leading to a large volume of data, only 5% of data is sufficient for measuring the effectiveness of the models. The feature of wind generation data in 2019 is tabulated in Table 1.

The parameters of the forecasting model, consisting of convolutional layers, GRU layers and fully connected NN, are tuned using grid search methods to improve the forecasting performance. In the proposed model, we have determined a convolutional layer, two GRU layers and two NN layers for better performance as demonstrated in Figure 9. The time-series input data are arranged in a way that can help to build a strong relationship between input and output data. In order to transform the time-series data into a data frame, 30 samples of time lags are employed using the sliding window method described in Figure 3, i.e., the past 30 input data are chosen to predict future data after several parameter tests. The data are used to train the model as a mini-batch size of 32 after several trails for improving training efficiency. To select random data sets for training and testing the models, shuffle buffer size is chosen 1000. We have trained the model to update its weight using the Truncated Backpropagation Through Time (BPTT) algorithm.

After training the proposed model, it is used to test the performance of the forecasting model to predict wind power generation of the wind farm. The forecasting results of the model is depicted in Figure 11a and the zoom in version from 600 to 1000 data points is illustrated in Figure 11b. It is observed that the forecasting value is close to the actual value. Most importantly, the errors of the forecasting values are very low even after several spikes exist in the forecasting data depicted in Figure 12. The majority of the error remains as low as less than 10% of the total generation capacity. The performance indices are used to measure the effectiveness of the forecasting model. In forecasting the test data, the average absolute difference between the forecasting value and actual value is 2.48, while the standard deviation of the predicted errors, i.e., the spread out of errors, is 3.87. As the both error values, MAE and RMSE, are not much different, it indicates a consistent size of errors, with the lower effect of outliers. The relative size of the forecasting errors for the wind farm can be observed on average as 9.94% of its power generation capacity. The convergence curve of the training and testing errors are demonstrated in Figure 13 from where it is observed that the model converges sharply, approximately within 10 epochs, and it does not have the issues of over- and under-fittings.

Table 3: Forecasting errors of prediction models.

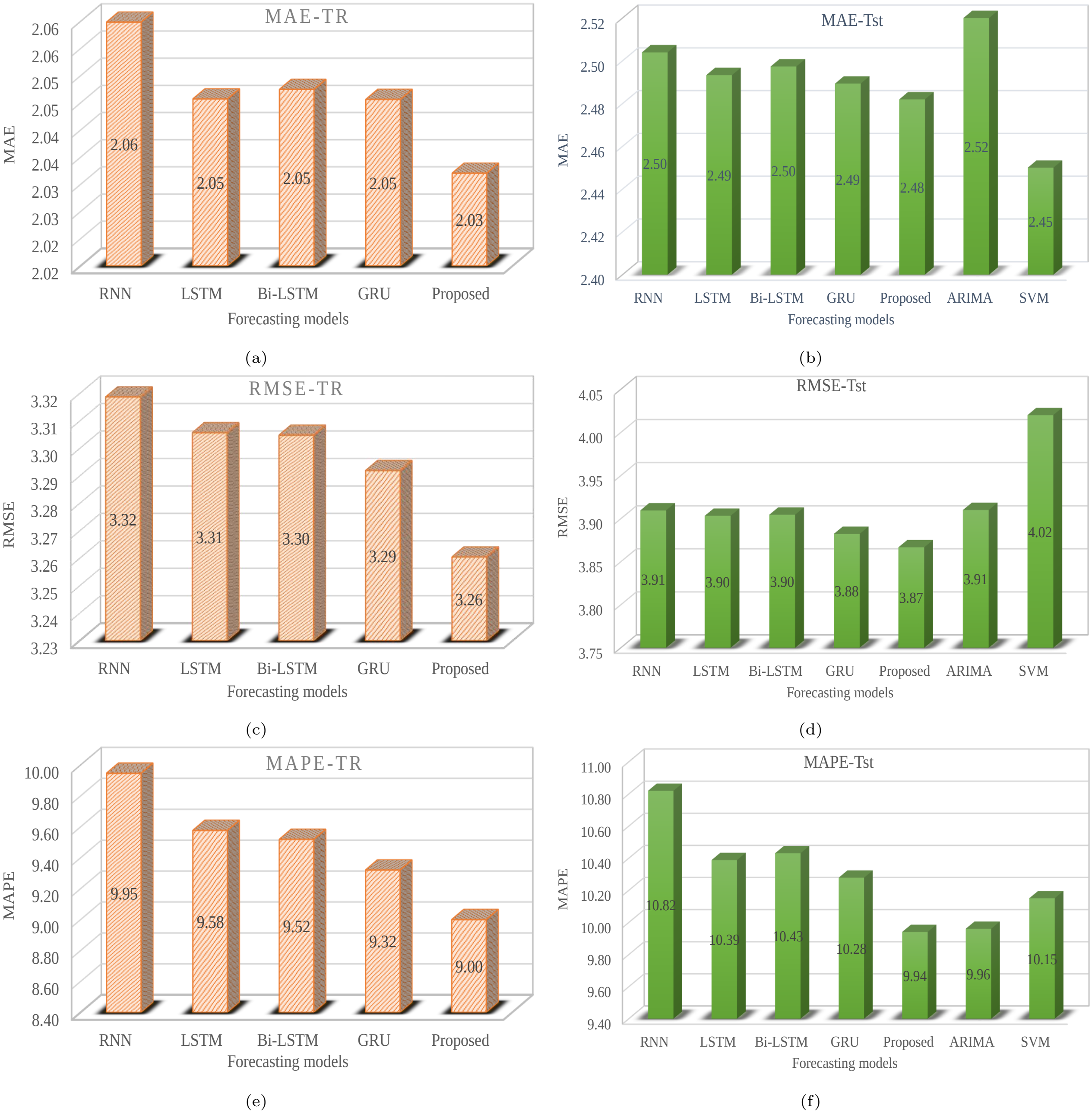
Models	**MAE**		**RMSE**		**MAPE**	
	Training	**Testing**	**Training**	**Testing**	**Training**	**Testing**
RNN	2.06	2.50	3.32	3.91	9.95	10.82
LSTM	2.05	2.49	3.31	3.90	9.58	10.39
Bi-LSTM	2.05	2.50	3.30	3.90	9.52	10.25
GRU	2.05	2.49	3.29	3.88	9.32	10.28
Proposed	2.03	2.48	3.26	3.87	9.00	9.94
ARIMA	—	2.52	—	3.91	—	9.96
SVM	—	2.45	—	4.02	—	10.15

Table 4: Outcome of parameter tuning for rival forecasting models.

Models	**# of layers (ranges, step)**	**# of neurons (ranges, step)**
RNN	3 (1–4, 1)	40 (10– 80, x2)
LSTM	1 (1–4, 1)	80 (10– 80, x2)
Bi-LSTM	2 (1–4, 1)	80 (10– 80, x2)
GRU	2 (1–4, 1)	40 (10– 80, x2)

4.3.1. Comparative analysis with parameter tuning

This subsection compares the outcome of the proposed model with the advanced forecasting models, namely, RNN, LSTM, Bi-LSTM, GRU, ARIMA and SVM. As the performance of all the models heavily depends on the parameter tuning, we have conducted an extensive tuning, a very time consuming approach, for all the models using the grid search approach. As the structure of the ARIMA and SVM forecasting model are different from the deep learning models, we have taken an extra care to tune their hyper-parameters and improve the outcome. As the performance of SVM model generally relies on the selection of SVM kernel, penalty parameter (C) and gamma [50], these parameters are varied to obtain better results using the grid search technique. For the kernel of radial basis function (rbf) and the loss function (epsilon) value of 0.01, the gamma value is chosen $1e^{-4}$ from the range [$1e^{-4}$ - 0.9] and C is selected 100 from the range [1–10,000]. Ten k-fold cross-validation is used to minimise mean square error during the grid search process. In the ARIMA model, the performance depends on the selection of pth-order auto-regression process, qth-order moving average and differencing parameter (d) [51]. For the tuning purpose, the maximum p and q are set to five while the d value is set to 1 for selecting the best value using auto ARIMA parameter tuning (pmdarima). The parameters are selected based on the minimum value of the Akaike information criterion (AIC) and other factors, such as sample quantiles, auto-correlation and statistical significance. The value of p, d and q are selected as 4, 1, 0, respectively, from the parameter tuning.



(a)

(b)

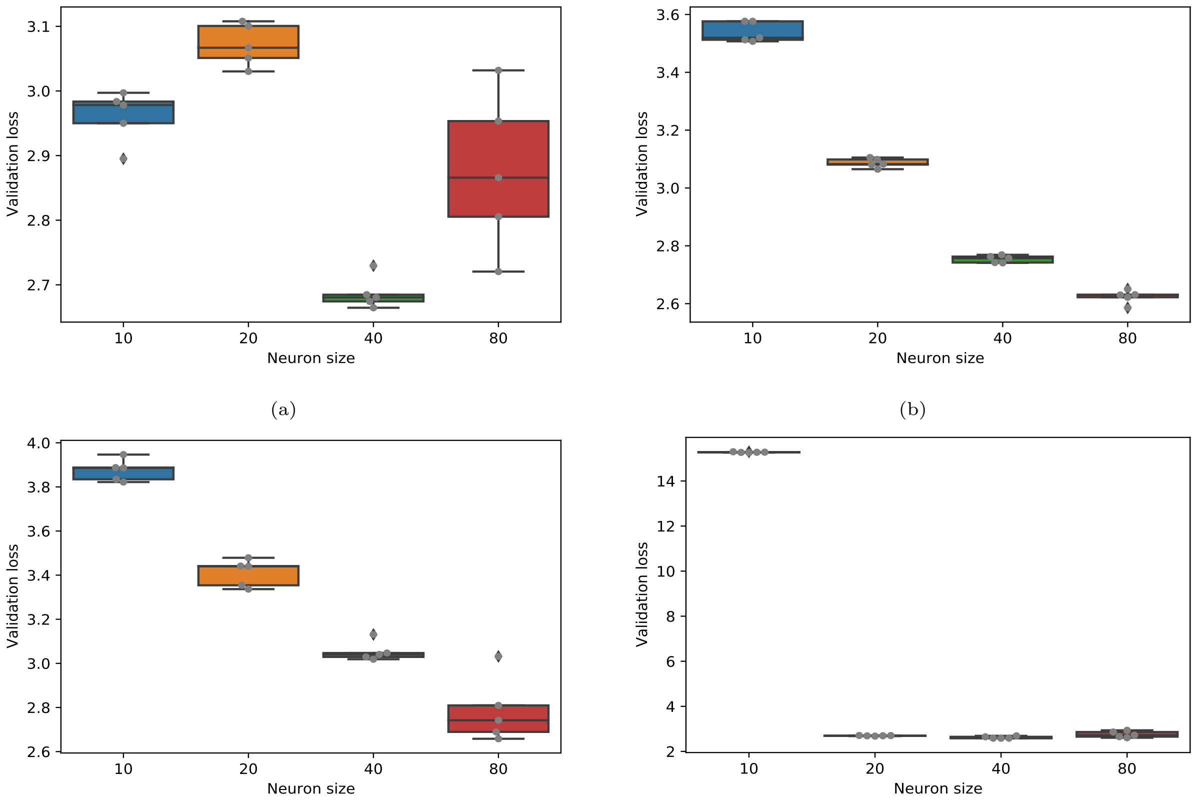
(c)

(d)

(e)

(f)

Figure 14: Errors in forecasting models: (a) MAE in training (b) MAE in testing (c) RMSE in training (d) RMSE in testing (e) MAPE in training and (f) MAPE in testing.



(a)

(b)

(c)

(d)

Figure 15: The neuron tuning of rival forecasting models: (a) RNN (b) LSTM (c) Bi-LSTM and (d) GRU.

The parameter tuning of the rival deep learning model is carried out using the grid search approach similar to the proposed model. As all the deep learning models are developed in the same frame, the previous selection of input data and mini-batch size are the same, i.e., window size = 30 and batch size = 32. Two parameters, the number of layers and the number of neurons, are chosen to save time in parameter tuning without compromising the performance. For avoiding too much time in computation, we have chosen five independent run to tune the parameters for presenting results statistically. The ranges, step of the grid search and chosen parameters for the rival models are tabulated in Table 4. It is observed that the number of layers and the neurons are not the same for all the forecasting models, indicating they have an effect to improve the performance. It is interesting to see that the LSTM has performed better in a layer with 80 neurons while GRU has taken two layers with 40 neurons in each layer. The tuning of neuron size in the models is demonstrated in Figure 15, where the validation loss is selected as a selection criteria. While the increasing neuron size enhances the performance for the LSTM and Bi-LSTM model, it was not continue with the further increase in neuron size after 80. In the GRU model, the distribution of the box shows in a compact form as compared to the others. This is because of a large scale in the validation axis that happens for the neuron size 10 whose validation loss is more than 15. From the analysis, it can be concluded that as the response of the rival forecasting models are different with the variation of the parameters, it is essential to tune all the competitive forecasting models using a tuning technique.

The outcome of the comparative models are illustrated in Figure 14 and tabulated in Table 3 for both the training and testing errors in terms of performance indices. The highest errors (2.52) in MAE for testing are found in the ARIMA model followed by the RNN and Bi-LSTM models. The SVM model shows the highest error (4.02) in RMSE for the testing set followed by the ARIMA and RNN models. In the performance index MAPE, the RNN model has highest errors (10.82) followed by the LSTM and GRU models. The proposed hybrid model demonstrates the best performance in terms of MAE, RMSE and MAPE, although the ARIMA model has also a good prediction accuracy in terms of MAPE. The training results of the ARIMA and SVM models are not demonstrated in Table 3. This is due to the fact that we have used 70 percentage of data in training the models and 30 per cent (equivalent to the 5 percentage of data in 2019 data set) for testing the models' performance since the large scale of data brings complexity in ARIMA and SVM models. Although the performances of the advanced deep learning models (LSTM, Bi-LSTM and GRU) are similar, the GRU model has demonstrated slightly better performance than other models during both the training and testing. The performance of the proposed model is improved due to the combined features of a convolutional layer, GRU layers and fully connected NN. Therefore, it can be concluded that the proposed method has improved the performance 1.59% in MAE compared to the ARIMA model, 3.73% in RMSE compared to the SVM model and 8.13% in MAPE compared to the RNN model.

4.3.2. Sensitivity analysis

This section is dedicated to the sensitivity analysis of the forecasting models to predict the wind power generation. The sensitivity analysis in the forecasting model indicates multiple runs of the model to determine its effectiveness during actual wind power forecasting. As machine learning models work on random numbers, it is essential to run the deep learning models several times to check its suitability during practical implementation. To present the results statistically, we have experimented the models eleven times with different seed values for each run. This experiment can be any run, but we have selected eleven to reduce computational time and find the median value easily. Forecasting the wind power using testing data are considered to represent the analysis as the data are unknown to the model. The median value is used as an evaluation criterion for determining the effectiveness due to their position among multiple runs.

As the ARIMA and SVM models demonstrate a deterministic result, i.e. the result is the same with separate runs, these are not included in the sensitivity analysis. The results of the multiple runs for MAE, RMSE and MAPE are demonstrated in Table 5, 6 and 7, respectively. The results of the forecasting models, especially for MAE and RMSE, are similar, with a small variation in their results. In terms of median value, the proposed model demonstrates the lowest error in forecasting the wind power generation of the wind farm although the standard deviation is slightly higher than others. The GRU model shows the second best model in reducing the prediction error in terms of MAE and RMSE. In MAPE, the proposed model also shows the higher performance followed by the Bi-LSTM model and GRU model. The higher error, i.e. 10.82, is found in the RNN model followed by the LSTM. The highest standard deviation in MAPE is in the RNN model followed by the Bi-LSTM model. From the analysis, it can be concluded that the proposed model shows a superior performance over other advanced models followed by the GRU model to predict the wind power generation of the wind farm.

4.3.3. Statistical analysis

To further demonstrate the effectiveness of the deep learning forecasting models, we have conducted two non-parametric tests: a Wilcoxon signed-rank test and Fredman's ranking test. The test determines whether a

Table 5: Comparative summary of the forecasting models using MAE-testing.

Models	Best	Median	Mean	Worst	Std.
Proposed	2.45	2.48	2.48	2.50	0.01203
GRU	2.48	2.49	2.49	2.50	0.00851
Bi-LSTM	2.49	2.50	2.50	2.51	0.00583
LSTM	2.49	2.49	2.50	2.51	0.00647
RNN	2.50	2.49	2.51	2.53	0.00900

Table 6: Comparative summary of the forecasting models using RMSE-testing.

Models	Best	Median	Mean	Worst	Std.
Proposed	3.85	3.87	3.87	3.89	0.01537
GRU	3.87	3.88	3.88	3.89	0.00767
Bi-LSTM	3.90	3.90	3.91	3.92	0.00683
LSTM	3.89	3.90	3.90	3.91	0.00727
RNN	3.90	3.91	3.91	3.94	0.01232

Table 7: Comparative summary of the forecasting models using MAPE-testing

Models	Best	Median	Mean	Worst	Std.
Proposed	9.80	9.94	10.08	10.79	0.30524
GRU	9.93	10.28	10.34	11.17	0.37298
Bi-LSTM	10.06	10.25	10.50	11.44	0.44547
LSTM	10.12	10.39	10.47	11.26	0.34625
RNN	10.40	10.82	10.98	12.39	0.57119

Table 8: Results of the Wilcoxon test using MAE-testing

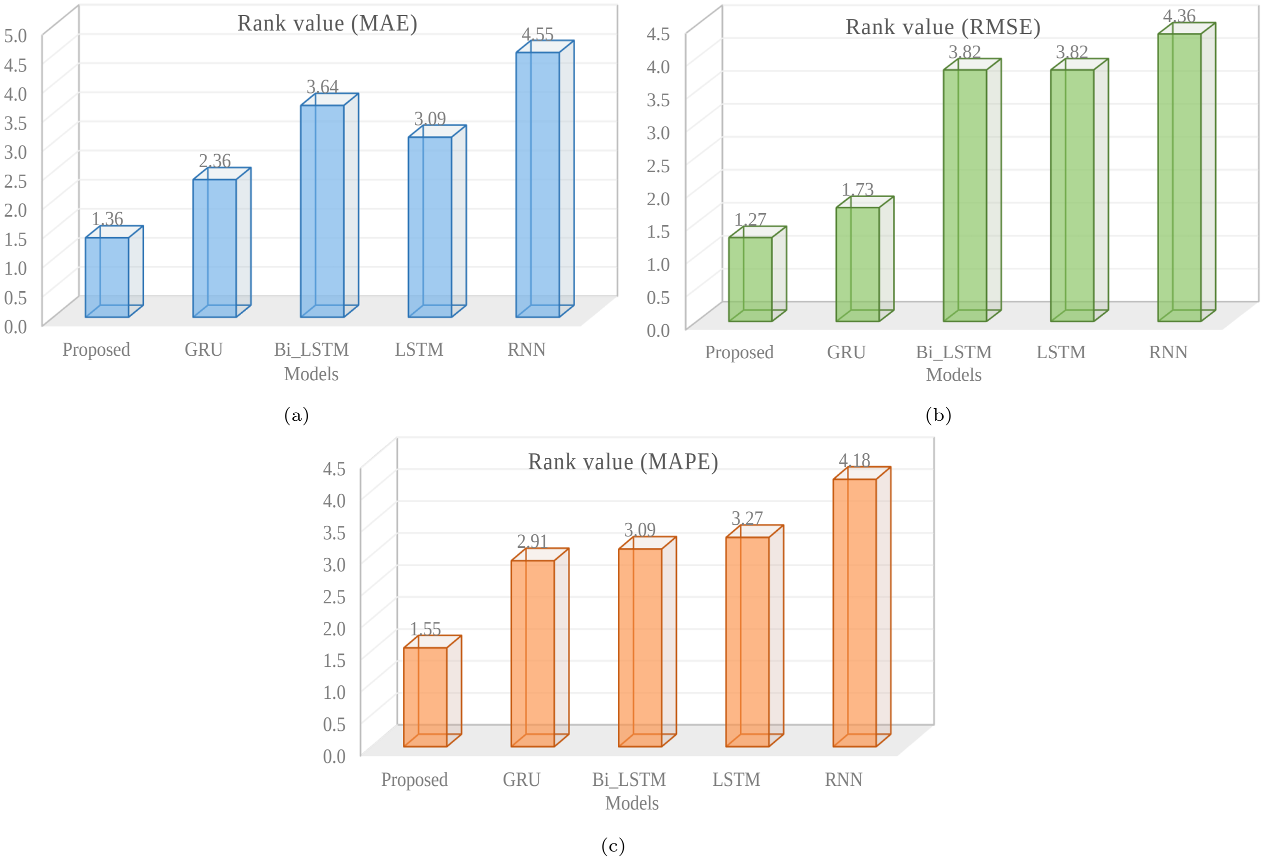
Models	R^+	R^-	P-value	Dec.
Proposed vs. GRU	57.0	9.0	0.029383	+
Proposed vs. Bi-LSTM	66.0	0.0	0.002897	+
Proposed vs. LSTM	62.0	4.0	0.008719	+
Proposed vs. RNN	66.0	0.0	0.002897	+

Table 9: Results of the Wilcoxon test using RMSE-testing

Models	R^+	R^-	P-value	Dec.
Proposed vs. GRU	58.0	8.0	0.023376	+
Proposed vs. Bi-LSTM	66.0	0.0	0.002897	+
Proposed vs. LSTM	66.0	0.0	0.002897	+
Proposed vs. RNN	66.0	0.0	0.002897	+

Table 10: Results of the Wilcoxon test using MAPE-testing

Models	R^+	R^-	P-value	Dec.
Proposed vs. GRU	51.0	15.0	0.100001	$\approx$
Proposed vs. Bi-LSTM	57.0	9.0	0.029383	$+$
Proposed vs. LSTM	61.0	5.0	0.011278	$+$
Proposed vs. RNN	66.0	0.0	0.002897	$+$



(a)

(b)

(c)

Figure 16: Friedman's ranking test: (a) MAE-testing (b) RMSE-testing and (c) MAPE-testing.

Table 11: Multi-step forecasting errors (median) of the proposed model.

Time steps	MAE	RMSE	MAPE
Second	2.52	3.90	10.40
Third	2.58	3.96	10.97
Fifth	2.75	4.14	12.36
Seven	2.94	4.31	13.73
Nine	3.11	4.49	14.41

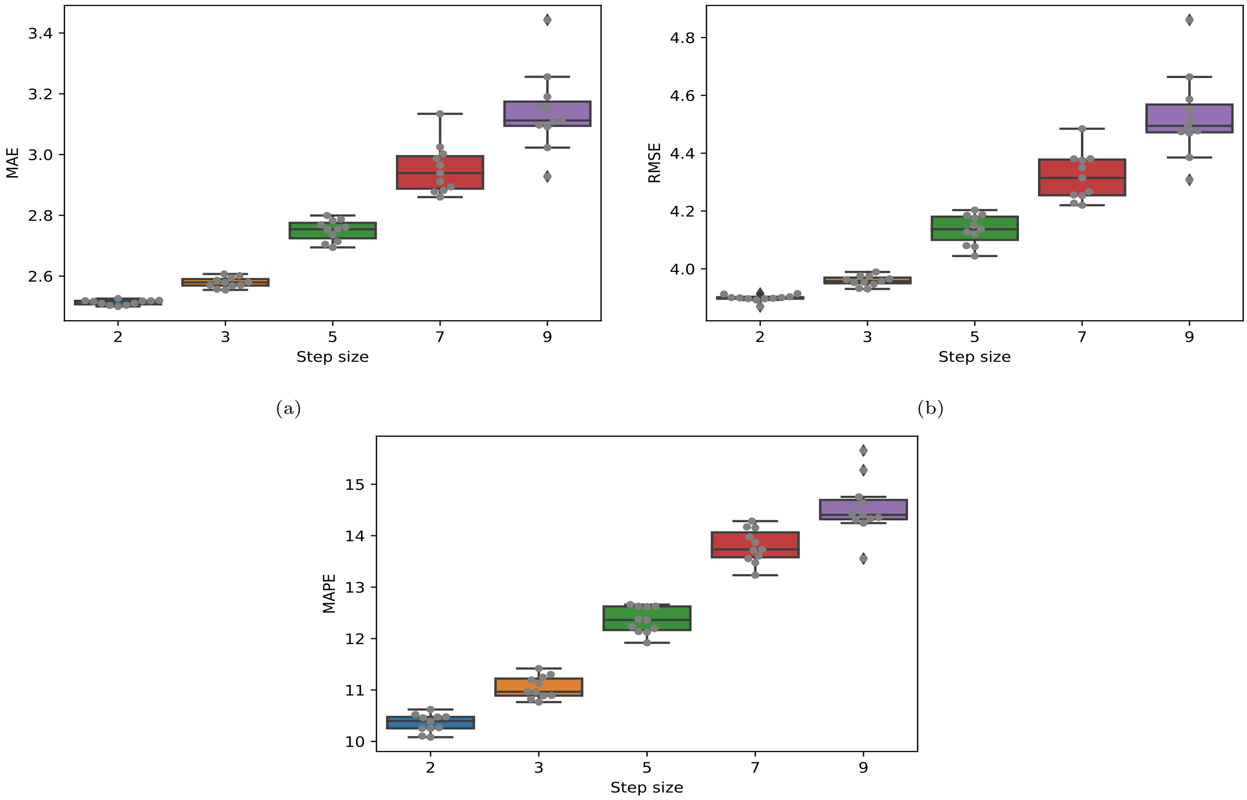
model has a significant difference in results as compared to the opponent models, where P-values less than 0.05 indicate the statistical significance of the proposed model [4]. In the Wilcoxon signed-rank test, two models are compared, where "+" sign refers to the significant superiority of the proposed model over the other, "−" sign indicates opposite, i.e., the proposed model is inferior to other one, and "≈" sign shows no statistical significant difference between the models' performances. The results obtained from the Wilcoxon signed-rank test using the performance indices of MAE, RMSE and MAPE are presented in Table 8, 9 and 10. The R^+ indicates the sum of positive ranks that is 66 if the model outperforms in each run. The ranking value is determined by multiplying the number of models with the number of independent runs, i.e., $6 \times 11 = 66$. The R^- refers to the sum of negative ranks with respect to the proposed model, indicating the inferiority of the proposed model over others for specific runs.

The results of the Wilcoxon test using the performance indices, MAE and RMSE, are demonstrated in Tables 8 and 9, respectively. The proposed model is compared with other models through the one-by-one approach. In MAE, the proposed model shows a statistical significance over other models, with negative rank 9 for GRU and 4 for LSTM. The model has again received the statistical significance over others in RMSE, with negative rank 8 for the GRU model. In the MAPE test shown in Table 10, there is no statistical significance in the performance of the proposed model with the GRU model as P-value is higher than 0.05. However, the proposed model shows the statistical significance as compared to the other models, such as LSTM, Bi-LSTM and RNN. The GRU model in the Wilcoxon test is found to be a competitive forecasting model as compared to other models.

To further analyse the effectiveness of the deep learning models, Friedman's rank test has been conducted among the models. The test ranks among the participated models from one to six depending on their performance. The lowest value indicates the best models while the highest value refers to inferior performance as compared to other models. The results of the test data in terms of MAE, RMSE and MAPE are depicted in Figure 16. In all the tests, the proposed model has performed the best followed by the GRU model. The LSTM model has shown a better result in MAE than the Bi-LSTM model that shows a better performance in MAPE than the LSTM one. In RMSE, the both LSTM and Bi-LSTM models have the same ranking. The RNN model performs the worst in all the performance indices. From the analysis, it can be concluded that the hybrid deep learning model is better than all other forecasting models based on the Wilcoxon signed-rank and Friedman's rank tests.

4.4. Multi-step forecasting

This section presents the multi-steps forecasting of the proposed model to determine the response in forecasting accuracy. The structure and hyper-parameters of the model are remained the same, with different time steps of training and testing sets, namely second, third, fifth, seven and nine. The multi-step predictions can be conducted two ways: updating prediction after each single time interval (i.e. 5 min in this study) and updating prediction after the time step size considered (i.e., 10 or 15 min). The later technique may lead to higher error in the forecasting as it updates the forecasting model after the long time step considered. Therefore, the former approach widely adopted in the literature and industry [52] is taken into account in this study to present the multi-step forecasting. The performance of the forecasting model for the multi-time steps is evaluated eleven times to statistically present the result as shown Figure 17. It is observed that the increasing time-steps rises



(a)

(b)

(c)

Figure 17: Multi-step forecasting errors: (a) MAE-testing (b) RMSE-testing and (c) MAPE-testing.

the value of performance indices, i.e., higher errors in the prediction. In MAE and RMSE, the error values are increasing exponentially while in MAPE the errors are increasing most like linearly. The median value of the performance indices for the multi-step prediction is tabulated in Table 11. Therefore, it can be concluded that the higher number of step size induces higher errors in forecasting the very short-term wind power generation of the Bodangora Wind Farm.

4.5. Discussion

This study has proposed a hybrid deep learning model to forecast the very-short term wind power generation of the Bodangora wind farm in Australia. The effectiveness of the forecasting model is compared with the other advanced forecasting models, including ARIMA and SVM models. The proposed hybrid model could be used using other combinations, such as the combination of CNN and LSTM models. We have conducted the experiment with the CNN-LSTM combination, not shown in the study, but the result of the proposed model demonstrates higher performance than the CNN-LSTM model. In addition, as the GRU model performed well as compared to the other deep learning models, we have built the hybrid model on it with the convolution layers.

As the ARIMA and SVM models demonstrate a deterministic result in forecasting the single-step wind power generation, these models are not participated in the sensitive and statistical analysis. To forecast through the ARIMA and SVM model, the data size was reduced keeping the testing data same as these model has complexity in dealing with the large number of data sample. The parameters of all the prediction models are chosen after the parameter tuning using the grid search approach. It is found that the performance of the forecasting model depends largely on the parameter tuning of the rival forecasting models. All the compared models are tuned separately, as a result, the outcome of the proposed model is not much higher as compared to other models. The outcomes of several case studies show the superiority of the hybrid deep learning model over others. The improved performance depends on the parameter tuning of its structures, number of neurons, number of layers, input data, batch size and other factors. If these parameters are not taken into consideration, the performance can easily deteriorate. Therefore, it is strongly recommended to conduct parameter tuning of all the forecasting models before demonstrating a comparison.

5. Conclusion

Accurate prediction of wind power generation as early as possible is crucial to enable better power dispatch, scheduling and unit commitment of energy resources including energy storage systems. In addition, accurate forecasting decreases the risk of economic and technical losses due to uncertain wind power generation, leading to participation of a wind farm in a more competitive energy market. To improve the prediction of very short-term wind power generation of the Bodangora wind farm in Australia, this paper proposes a novel hybrid deep learning model, consisting of convolutional layers, gated recurrent unit layers and neural network layers. The convolutional layers extract data feature while gated recurrent units retain information in memory to improve the forecasting accuracy. The forecasting model is trained and tested using a 5-min data sample of the real wind power generation in the wind farm. The performance of the model proposed to forecast power generation is compared with the advanced forecasting models and evaluated using performance indices, namely MAE, RMSE and MAPE. Sensitivity and statistical analysis of multiple runs with different seed numbers are also carried out to further assess the effectiveness of the models. Moreover, the forecasting model is used to predict the multi-step forecasting to demonstrate the error in a large horizon. It is found that the hybrid deep learning model demonstrates higher accuracy than all other models, especially up to 1.59% in MAE, 3.73% in RMSE and 8.13% in MAPE, to predict the very short-term wind power generation of the wind farm. In addition, it is also observed that the GRU model shows better performance than the other deep learning models, such as LSTM, Bi-LSTM and RNN. The outcomes from the parameter tuning indicate that each model has its own feature that needs to be explored for improving the performance.

The proposed forecasting model can also be utilised to improve the accuracy of other prediction tasks, such as wind speed, solar power generation, power demand, electricity prices, stock index and traffic flow. This can be performed by training and validating the model with time-series data while the structure of the hybrid model may depend on the degree of complexity in the forecasting data. Although a large volume of data set has been used in this study, a small volume of data can also be utilised to train the model for prediction purposes, which is common with limited sets of available data.

In the future, the proposed model will be further evaluated against recent forecasting models using a number of wind farm data, with several case studies. The parameter turning of the model will also be performed using advanced optimisation algorithms for reducing parameter time and more accurate results.

Acknowledgement

The authors would like to thank Geoff Eldridge for helping in the data collection of the wind farm.

References

[1] M. A. Hossain, H. R. Pota, W. Issa, M. J. Hossain, Overview of ac microgrid controls with inverter-interfaced generations, Energies 10 (9) (2017) 1300.

[2] M. A. Hossain, H. R. Pota, M. J. Hossain, F. Blaabjerg, Evolution of microgrids with converter-interfaced generations: challenges and opportunities, International Journal of Electrical Power & Energy Systems 109 (2019) 160–186.

[3] A. Dhabi, Renewable capacity statistics 2018, IRENA,≪Renewable Capacity Statistics (2019).

[4] M. A. Hossain, H. R. Pota, S. Squartini, F. Zaman, J. M. Guerrero, Energy scheduling of community microgrid with battery cost using particle swarm optimisation, Applied Energy 254 (2019) 113723.

[5] Y.-Y. Hong, C. L. P. P. Rioflorido, A hybrid deep learning-based neural network for 24-h ahead wind power forecasting, Applied Energy 250 (2019) 530–539.

[6] N. Chen, Z. Qian, I. T. Nabney, X. Meng, Wind power forecasts using gaussian processes and numerical weather prediction, IEEE Transactions on Power Systems 29 (2) (2013) 656–665.

[7] C. Wan, J. Wang, J. Lin, Y. Song, Z. Y. Dong, Nonparametric prediction intervals of wind power via linear programming, IEEE Transactions on Power Systems 33 (1) (2017) 1074–1076.

[8] J. Dowell, P. Pinson, Very-short-term probabilistic wind power forecasts by sparse vector autoregression, IEEE Transactions on Smart Grid 7 (2) (2015) 763–770.

[9] J. Wang, W. Yang, P. Du, T. Niu, A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm, Energy Conversion and Management 163 (2018) 134–150.

[10] X. Luo, J. Sun, L. Wang, W. Wang, W. Zhao, J. Wu, J.-H. Wang, Z. Zhang, Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy, IEEE Transactions on Industrial Informatics 14 (11) (2018) 4963–4971.

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

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

[13] G. Sideratos, N. D. Hatziargyriou, Probabilistic wind power forecasting using radial basis function neural networks, IEEE Transactions on Power Systems 27 (4) (2012) 1788–1796.

[14] P. Lu, L. Ye, W. Zhong, Y. Qu, B. Zhai, Y. Tang, Y. Zhao, A novel spatio-temporal wind power forecasting framework based on multi-output support vector machine and optimization strategy, Journal of Cleaner Production (2020) 119993.

[15] V. A. Natarajan, N. S. Kumari, Wind power forecasting using parallel random forest algorithm, in: Soft Computing for Problem Solving, Springer, 2020, pp. 209–224.

[16] A. Kavousi-Fard, A. Khosravi, S. Nahavandi, A new fuzzy-based combined prediction interval for wind power forecasting, IEEE Transactions on Power Systems 31 (1) (2015) 18–26.

[17] F. Valencia, J. Collado, D. Sáez, L. G. Marín, Robust energy management system for a microgrid based on a fuzzy prediction interval model, IEEE Transactions on Smart Grid 7 (3) (2015) 1486–1494.

[18] L.-L. Li, X. Zhao, M.-L. Tseng, R. R. Tan, Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm, Journal of Cleaner Production 242 (2020) 118447.

[19] J. Yan, H. Zhang, Y. Liu, S. Han, L. Li, Z. Lu, Forecasting the high penetration of wind power on multiple scales using multi-to-multi mapping, IEEE Transactions on Power Systems 33 (3) (2018) 3276–3284.

[20] M. Khodayar, J. Wang, M. Manthouri, Interval deep generative neural network for wind speed forecasting, IEEE Transactions on Smart Grid 10 (4) (2018) 3974–3989.

[21] H.-z. Wang, G.-q. Li, G.-b. Wang, J.-c. Peng, H. Jiang, Y.-t. Liu, Deep learning based ensemble approach for probabilistic wind power forecasting, Applied energy 188 (2017) 56–70.

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

[23] A. S. Devi, G. Maragatham, K. Boopathi, A. Rangaraj, Hourly day-ahead wind power forecasting with the eemd-cso-lstm-efg deep learning technique, Soft Computing (2020) 1–21.

[24] Q. Qin, X. Lai, J. Zou, Direct multistep wind speed forecasting using lstm neural network combining eemd and fuzzy entropy, Applied Sciences 9 (1) (2019) 126.

[25] O. Abedinia, M. Bagheri, M. S. Naderi, N. Ghadimi, A new combinatory approach for wind power forecasting, IEEE Systems Journal (2020).

[26] Z. Niu, Z. Yu, W. Tang, Q. Wu, M. Reformat, Wind power forecasting using attention-based gated recurrent unit network, Energy 196 (2020) 117081.

[27] C. Li, G. Tang, X. Xue, X. Chen, R. Wang, C. Zhang, The short-term interval prediction of wind power using the deep learning model with gradient descend optimization, Renewable Energy (2020).

[28] Y. Sun, P. Wang, S. Zhai, D. Hou, S. Wang, Y. Zhou, Ultra short-term probability prediction of wind power based on lstm network and condition normal distribution, Wind Energy 23 (1) (2020) 63–76.

[29] P. Li, X. Wang, J. Yang, Short-term wind power forecasting based on two-stage attention mechanism, IET Renewable Power Generation 14 (2) (2020) 297–304.

[30] G. Wang, R. Jia, J. Liu, H. Zhang, A hybrid wind power forecasting approach based on bayesian model averaging and ensemble learning, Renewable Energy 145 (2020) 2426–2434.

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

[32] R. Liu, M. Peng, X. Xiao, Ultra-short-term wind power prediction based on multivariate phase space reconstruction and multivariate linear regression, Energies 11 (10) (2018) 2763.

[33] P. Sorensen, N. A. Cutululis, A. Vigueras-Rodríguez, L. E. Jensen, J. Hjerrild, M. H. Donovan, H. Madsen, Power fluctuations from large wind farms, IEEE Transactions on Power Systems 22 (3) (2007) 958–965.

[34] A. Kusiak, H. Zheng, Z. Song, On-line monitoring of power curves, Renewable Energy 34 (6) (2009) 1487–1493.

[35] L. Tan, J. Han, H. Zhang, Ultra-short-term wind power prediction by salp swarm algorithm-based optimizing extreme learning machine, IEEE Access 8 (2020) 44470–44484.

[36] J. Ma, M. Yang, Y. Lin, Ultra-short-term probabilistic wind turbine power forecast based on empirical dynamic modeling, IEEE Transactions on Sustainable Energy 11 (2) (2019) 906–915.

[37] H. Jiajun, Y. Chuanjin, L. Yongle, X. Huoyue, Ultra-short term wind prediction with wavelet transform, deep belief network and ensemble learning, Energy Conversion and Management 205 (2020) 112418.

[38] Y.-D. Syu, J.-C. Wang, C.-Y. Chou, M.-J. Lin, W.-C. Liang, L.-C. Wu, J.-A. Jiang, Ultra-short-term wind speed forecasting for wind power based on gated recurrent unit, in: 2020 8th International Electrical Engineering Congress (iEECON), IEEE, 2020, pp. 1–4.

[39] S. Vorrath, Energyaustralia signs ppa for 113mw wind farm in nsw, Renew Economy (2017).

[40] M. A. Hossain, H. R. Pota, S. Squartini, A. F. Abdou, Modified PSO algorithm for real-time energy management in grid-connected microgrids, Renewable energy 136 (2019) 746–757.

[41] A. Gasparin, S. Lukovic, C. Alippi, Deep learning for time series forecasting: The electric load case, arXiv preprint arXiv:1907.09207 (2019).

[42] X. Liu, Z. Zhang, Z. Song, A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning, Renewable and Sustainable Energy Reviews 119 (2020) 109632.

[43] J. Bedi, D. Toshniwal, Deep learning framework to forecast electricity demand, Applied energy 238 (2019) 1312–1326.

[44] J. L. Elman, Finding structure in time, Cognitive science 14 (2) (1990) 179–211.

[45] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (2) (1994) 157–166.

[46] S. Han, Y.-h. Qiao, J. Yan, Y.-q. Liu, L. Li, Z. Wang, Mid-to-long term wind and photovoltaic power generation prediction based on copula function and long short term memory network, Applied energy 239 (2019) 181–191.

[47] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.

[48] D.-X. Zhou, Universality of deep convolutional neural networks, Applied and computational harmonic analysis 48 (2) (2020) 787–794.

[49] G. P. Meyer, An alternative probabilistic interpretation of the huber loss, arXiv preprint arXiv:1911.02088 (2019).

[50] A. Zendehboudi, M. Baseer, R. Saidur, Application of support vector machine models for forecasting solar and wind energy resources: A review, Journal of cleaner production 199 (2018) 272–285.

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

[52] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on lstm recurrent neural network, IEEE Transactions on Smart Grid 10 (1) (2017) 841–851.