energies

MDPI

Article

Revolutionizing Wind Power Prediction—The Future of Energy Forecasting with Advanced Deep Learning and Strategic Feature Engineering

Md. Ahasan Habib and M. J. Hossain *

School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia; mdahasan.habib@student.uts.edu.au
* Correspondence: jahangir.hossain@uts.edu.au

Abstract: This paper introduces an innovative framework for wind power prediction that focuses on the future of energy forecasting utilizing intelligent deep learning and strategic feature engineering. This research investigates the application of a state-of-the-art deep learning model for wind energy prediction to make extremely short-term forecasts using real-time data on wind generation from New South Wales, Australia. In contrast with typical approaches to wind energy forecasting, this model relies entirely on historical data and strategic feature engineering to make predictions, rather than relying on meteorological parameters. A hybrid feature engineering strategy that integrates features from several feature generation techniques to obtain the optimal input parameters is a significant contribution to this work. The model's performance is assessed using key metrics, yielding optimal results with a Mean Absolute Error (MAE) of 8.76, Mean Squared Error (MSE) of 139.49, Root Mean Squared Error (RMSE) of 11.81, R-squared score of 0.997, and Mean Absolute Percentage Error (MAPE) of 4.85%. Additionally, the proposed framework outperforms six other deep learning and hybrid deep learning models in terms of wind energy prediction accuracy. These findings highlight the importance of advanced data analysis for feature generation in data processing, pointing to its key role in boosting the precision of forecasting applications.

Keywords: wind energy forecasting; decomposition; feature engineering; deep learning

check for updates

Citation: Habib, M.A.; Hossain, M.J. Revolutionizing Wind Power Prediction—The Future of Energy Forecasting with Advanced Deep Learning and Strategic Feature Engineering. *Energies* **2024**, *17*, 1215. https://doi.org/10.3390/en17051215

Academic Editors: Chuan He, Zhengmao Li, Tao Chen, Rui Wang and Ziming Yan

Received: 7 February 2024
Revised: 26 February 2024
Accepted: 28 February 2024
Published: 3 March 2024

1. Introduction

Renewable energy (RE) sources are becoming more popular and play a vital role in global attempts to prevent climate change and promote sustainable development. In light of the global transition towards environmentally favorable and sustainable energy solutions, wind energy is assuming an integral place. Wind energy is abundant, renewable, and emits no greenhouse gases, making it ideal for sustainable energy generation. The transition to RE seeks to tackle environmental concerns and is in line with the Sustainable Development Goals (SDGs), particularly Goal 7, which focuses on easily accessible and clean energy, and Goal 13, which spotlights climate action [1]. These aims emphasize sustainable energy sources to alter global energy systems by 2030. Wind energy can reduce fossil fuel use and transition nations to cleaner, more sustainable power [2]. All things considered, wind energy is not merely a part of the RE scene; rather, it is an essential driving force behind the current energy revolution, bringing us closer to a more robust and sustainable energy future.

The integration of wind energy into the conventional electrical grid faces several major challenges due to the intermittent nature of wind power. These constraints include limitations in system capacity, issues related to reactive power control, high connection costs, complications in grid planning, etc. [3–6]. Scholars are investigating potential solutions to the aforementioned obstacles associated with wind energy integration. For example, in [7,8] the authors explored the use of modern power electronics to facilitate the smooth integration of wind energy into the grid, pointing out that energy storage systems are currently being implemented to mitigate the unpredictable nature of wind energy [9,10]. In this context, smart grid technology is a significant consideration. This technology enhances the grid's capacity to promptly adjust and react to fluctuations in wind energy supply, thereby improving the overall dependability and efficiency of the electrical system [11,12]. Another key strategy to tackle the outstanding challenges is the application of highly sophisticated forecasting techniques. Accurate wind energy forecasts enable grid operators to anticipate fluctuations in power supply, resulting in better system balance and energy management decisions [13,14]. This technique is widely recognized as a fundamental tactic in the process of integrating wind energy, which ensures both grid stability and effective energy management.

In recent years, machine learning (ML) and deep learning (DL) methods have been widely used to predict wind energy generation [15–17]. Researchers have experimented with a variety of models and algorithms to improve the accuracy and reliability of wind power estimates. Advanced ML methods such as Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), Gradient Boosting Machine (GBM), AdaBoost, XGBoost, LightGBM, and CatBoost are popular for wind speed and energy production predictions [18–21]. The ability of these models to handle big datasets and identify complicated, nonlinear relationships has led to more precise predictions than conventional statistical methods. Furthermore, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Bidirectional LSTM (Bi-LSTM) have demonstrated outstanding capabilities in capturing the temporal dynamics and inherent variability of wind energy patterns [16,17,22]. As a result, these models offer more accurate and precise forecasts in energy forecasting applications. The further development and implementation of these methods could promote renewable energy technology and accelerate the transition to cleaner energy systems.

The primary challenge of adopting deep learning algorithms for forecasting is the requirement for large and comprehensive datasets for use in training. In order to create reliable predictions, these models require features that are highly linked with the target variable. Although modern storage technologies make historical wind power-generating data more accessible, obtaining associated data for different meteorological conditions that affect wind energy remains challenging. Wind patterns are heavily influenced by factors such as air pressure, wind direction, gusts, rainfall, precipitation, and humidity [23,24]. However, reliable and detailed data on these factors are sometimes unavailable. Deep learning models rely on accuracy and comprehensiveness; hence, these errors can reduce their performance. To address this difficulty, researchers are concentrating on examining historical trends in wind generation to forecast future energy output, with the intention of extracting deeper insights from the available wind generation data through the implementation of numerous decomposition techniques. Frequently employed decomposition techniques include the Fourier Transform (FT), Wavelet Transform (WT), Empirical Mode Decomposition (EMD), Ensemble Empirical Mode Decomposition (EEMD), Variational Mode Decomposition (VMD), Singular Spectrum Analysis (SSA), etc. [25–27]. In the absence of data, this strategy for determining wind generation data patterns may improve wind energy prediction. The following discussion highlights recent studies in which researchers utilize various decomposition strategies to generate features for wind energy generation prediction.

In [28], the authors sought to enhance wind power forecasting by incorporating a decomposition algorithm to select features effectively, and additionally investigated the use of data-driven approaches and gated recurrent neural networks in the forecasting process. They employed Recursive Feature Elimination (RFE) and the Extra Tree Classifier (ETC) to choose the most relevant features from SCADA data. The variables included in the data were wind speed at various heights, generator and gearbox temperature, pitch angle, rotor speed, and other turbine parameters. However, their article did not include the decomposition outcome or input parameters. In addition, it is possible that not all input data were available for every wind turbine, which would mean that the model could not be used everywhere.

A new decomposition technique called Complete EEMD (CEEMD) which breaks down the wind energy signal into five intrinsic mode functions and a residual was introduced in [29]. To improve forecasting accuracy, these data were combined with atmospheric and turbine-specific information such as wind direction, temperatures, generator speed, nacelle direction, and wind speed. A Stacking Ensemble Learning model then used these data for short-term predictions in different months. For instance, the model achieved an MAE (Mean Absolute Error) of 89.18, an MAPE (Mean Absolute Percentage Error) of 7.67, and an RMSE (Root Mean Square Error) of 125.55 for a 20 min advance forecast in October. However, the suggested wind energy forecasting approach uses one model for each month, making it difficult to precisely acquire input parameter values and guarantee the same consistency.

In [30], the authors optimized a model for predicting wind speed using Bi-LSTM by adding four decomposition methods: WT, EMD, Empirical WT (EWT), and EEMD. These techniques break down the wind speed into low-frequency and high-frequency signals to enhance forecasting accuracy. The EWT+Bi-LSTM model performed well on the Dhanushkodi and Melamandai datasets, achieving MAEs of 0.1521 and 0.2791 m/s and R squared values of 98.94% and 97.99%, respectively. However, the research focused on a single decomposition method for the proposed model, rather than exploring multiple feature engineering strategies, which might boost the results.

The article in [31] presented a new approach for combining wind speed forecasts that employ decomposition techniques and an advanced optimization approach. In a first step, the historical wind speed data were divided into stable components with different frequencies using the VMD algorithm. The authors used two datasets to test the model and generated nine IMFs for each using VMD. After that, an echo state network predicted outputs for each IMF and an upgraded whale optimization approach was used to improve accuracy. The final prediction was obtained by adding the forecasting values together. VMD decomposes a single signal into many frequencies based on a predetermined number of modes; however, no explanation was provided in the manuscript for selecting nine IMFs and optimizing modes.

In [32], complimentary EEMD, the whale optimization algorithm (WOA), and the Elman neural network (ENN) model were used to estimate ultra-short-term wind power. This method reduces modal mixing during decomposition, cuts down on rebuilding mistakes, and runs more smoothly. The authors used data from two wind turbines to verify their proposed approach. They evaluated nine additional features from historical data by using the decomposition technique, and obtained MAPEs of 2.63% and 2.41% and RMSEs of 16.13 and 13.26, respectively, for the two turbines. Regrettably, the model only works for these specific wind turbines, and is not applicable to other models or large wind power firms.

The wind speed series analysis in [33] was carried out using EMD with linear and nonlinear Quantile Regression (QR) models, including QRNN. Kernel Density Estimation (KDE) was used in the framework to predict probability density functions of wind speed. The decomposition procedure yielded just two IMFs and a single residual feature; however, the reason for this small number of features for wind speed forecasting was not clearly defined in the research.

The authors of [34] recommended a hybrid methodology for wind speed forecasting combining VMD, Contrastive Learning of Seasonal Trend representations (CoST), and Support Vector Regression (SVR). The VMD-CoST-SVR approach entails denoising wind speed data using VMD, extraction of disentangled seasonal and trend information by CoST, and the application of SVR for prediction. However, the CoST technique is computationally inefficient, lacks scalability when dealing with big or real-time datasets, is sensitive to training data quality and volume, and is limited to certain types of time series data.

A hybrid VMD-CNN-GRU model for short-term wind power forecasting using VMD for data preprocessing, a CNN for spatial feature extraction, and a GRU unit for temporal feature processing was presented in [35]. The model provided accurate and reliable forecasting, with an RMSE of 1.5651 MW, MAE of 0.8161 MW, MAPE of 11.62%, and R-squared score of 0.9964. In this case, the proposed approach combined the four IMFs produced by the VMD algorithm with other meteorological characteristics such as wind direction, speed, temperature, humidity, air pressure, and air density. The investigation revealed that the expected output was more closely tied to meteorological factors than IMFs; thus, the absence of atmospheric parameters leads to system failure.

The study described in [36] presented a comprehensive wind power forecasting approach combining four fundamental forecasting engines for enhanced accuracy. DBSCAN was used to clean the data, Lagrange interpolation to fill in empty data, and a two-stage decomposition method combining EEMD and WT to prepare the data. The EEMD decomposes the wind power data into multiple IMFs with varying frequencies and one residual during the decomposition phase, then WT further divides the IMF1 into many modes. Instead of analyzing all IMFs highly associated with wind energy data, the authors randomly selected one, which represents a major drawback of this method.

The authors of [37] used the EEMD with permutation entropy (EEMD-PE) technique to break down original wind power time series data into several different complex features. The authors used the gravitational search algorithm (GSA) to maximize performance and the least squares support vector machine model (LSSVM) for prediction. The EEMD-PE decomposition technique yielded twelve features, which were categorized into four entropy groups. In this case, the authors did not provide a clear explanation for the random selection of the number of groups. However, a single model is not appropriate for the entire analysis period, as the total data are split into four dates based on China's seasonal pattern. Furthermore, the authors ignored the most prominent performance metrics (MAE, MAPE, RMSE, R-squared score) when demonstrating their model's superiority. Similarly, in [38] the authors divided time zones into four groups and analysed proposed models for different horizons. In this framework, the least absolute shrinkage and selection operator–quantile regression neural network (LASSO-QRNN) model is combined with the EEMD. While the results of both articles are excellent, the main limitation is that model performance is dependent on the seasonal patterns present in the training and testing data.

In [39], the authors suggested a way to predict very-short-term wind power by fixing Numerical Weather Prediction (NWP) wind speeds and separating changing weather events using double clustering. This method can improve the accuracy of forecasts by integrating wind speed correction and employing double clustering. Although the NWP correction mechanism and multiple clustering approach can be adapted to certain datasets, this approach may not be applicable to other wind power forecasting scenarios or regions. In [40], phase space reconstruction and a broad learning system were introduced to estimate the wind speed. After reconstructing phase spaces under various delay dimensions and phase scales, this approach uses the natural neighbor spectrum without parameter setting to identify the ideal phase space and employs elastic–net regularization to reduce overfitting. However, the proposed model's performance is sensitive to parameter selection, and potential variations in the effectiveness of elastic–net regularization for preventing overfitting may pose limitations, especially for diverse datasets. A novel spatiotemporal wind speed prediction model using an optimally weighted graph convolutional network (GCN) and GRU was presented in [41]. Through innovative preprocessing of the data, the proposed model was able to outperform existing models in multi-step short-term forecasting. However, GCN and GRU may not extract spatiotemporal features accurately; additionally, the ability of the Huber loss to mitigate the negative impacts of outliers on prediction performance remains unclear.

The above review of the literature on wind power prediction methods that use decomposition to create features is summarized in Table 1, from which it is clear that each method has certain drawbacks. Many studies are limited in their applicability to other turbines or locations [28,32,37,38], as they rely on data from individual wind turbines or the atmosphere. The effectiveness of prediction models is heavily dependent on the availability and accuracy of these factors. If meteorological data are unavailable or inaccurate, the model results may have large differences from the real values, which could make their forecasts less accurate. Furthermore, a number of approaches frequently performed other difficult and computationally demanding processes without taking into account all available attributes, instead focusing on only one or two [31,33,34,36].

Table 1. Key information from recent wind generation forecasting publications on decomposition and model development, emphasizing limitations.

Ref.	Decomposition Approach	Prediction Model	No. of Features	Limitations
[28]	RFE and ETC	LSTM and GRNN	Not specified	• Limited to specific turbine data. • Not universally applicable.
[29]	CEEMD	Stacking-ensemble Learning	Five IMFs + residuals	• Challenges in input value accuracy • Reliability concerns
[30]	EMD, EEMD, WT, EWT	Bi-LSTM	Not specified	• Single decomposition strategy • Lack of diverse feature analysis
[31]	VMD	Echo State Network	Nine IMFs	• No rationale for selecting IMFs • VMD optimization not explained
[32]	EEMD + WOA	ENN	Nine IMFs	• Limited to specific turbines • Not applicable to larger firms
[33]	EMD	QRNN	Two IMFs + residual	• Limited feature selection without explanation
[34]	VMD, CoST	SVR	Not specified	• Computational inefficiency • Scalability issues
[35]	VMD	CNN-GRU	Four IMFs	• Dependence on meteorological factors • Absence leads to failure
[36]	EEMD, WT	Ensemble Forecasting	Multiple modes	• Random selection of IMFs • Diverse time series may affect real-world engine forecasting performance.
[37]	EEMD-PE	LSSVM-GSA	Twelve features in four groups	• Unclear rationale for the number of groups • Seasonal data dependency
[38]	EEMD	LASSO–QRNN	Not specified	• Seasonal data pattern dependency

The above analysis raises issues about comprehensiveness of the analysis provided in previous studies and highlights missed opportunities for more precise predictions. Exploring a broader range of decomposition approaches may yield new features and improve the predictive power of future models. This study seeks to overcome these constraints by employing a cutting-edge feature engineering approach. Rather than depending on specific data about atmospheric conditions or wind turbines, in this research we make use of comprehensive historical data on wind power. To provide a more versatile and generally applicable forecasting model, this study emphasizes the selection of highly correlated datasets using different feature engineering combinations. To improve the accuracy and dependability of the model, a thorough analysis is carried out to optimize the input parameters. The main contributions of the proposed research are listed below:

- Extensive analysis of multiple single and hybrid decomposition methods and their combinations is performed to efficiently break down the wind generation data, resulting in important insights into seasonal and internal patterns.
- By using a feature selection approach to optimize the model inputs, we ensure that only highly correlated features are incorporated into the forecasting model.
- Proficient adjustment of hyperparameters refines the performance and precision of the forecasting model.

- A comprehensive comparison with cutting-edge deep learning models is carried out to verify the superiority of the proposed model in predicting wind energy power generation.

The following is an arrangement of the paper's further discussion. Section 2 presents a thorough investigation of the proposed methodology, including a complete explanation of feature engineering (FE), associated mathematical expressions, and a deep analysis of the forecasting model. The heart of our findings is explored in Section 3, which analyzes model performance to determine the optimal FE technique along with hyperparameter adjustments and a comparison with other models. Finally, Section 4 serves as a summary of our findings and presents potential areas for future research, providing a coherent conclusion to this study.

2. Methodology and Model Description

The comprehensive framework of this research is illustrated in Figure 1, which presents the overall structure encompassing data gathering and the forecasting of wind energy generation in Australia. Australia has made substantial contributions to the generation of renewable energy in recent years, and is widely acknowledged for its abundance of wind energy. More than a third of Australia's renewable energy production in 2021 came from wind energy, representing 10% of the country's total energy generation. This study focuses specifically on wind power firms located in the New South Wales region of Australia. The proposed strategy began with the collection of historical data from the national databank. The main part starts with the Feature Engineering (FE) Unit. Here, the most important step is to analyze and create an extensive number of features. This phase uses FE algorithms to break down the data into multiple pieces in order to extract wind generation seasonality, patterns, and insights. The approaches utilized for feature generation are separated into two categories: pattern extraction and decomposition. Seasonality analysis (SA) and singularity spectrum analysis (SSA) are used for pattern recognition and data insights, whereas VMD, EMD, and EEMD algorithms are used for decomposition.

Afterwards, the Feature Selection and Input Processing Unit refines the previously created features. Statistical methods are used to identify those features that are highly correlated with wind generation prediction. After selecting the final input parameters, the data are normalized to ensure compatibility with the prediction model. Following this stage, the data are divided into training, validation, and testing phases. The Model Training, Optimisation, and Testing phase is the final level in the proposed approach. In this stage, the model is trained using advanced deep learning techniques, which are then iteratively modified and fine tuned to improve performance. In this stage, hyperparameter tuning is used to modify the model parameters to identify the best configuration. After training, the model is rigorously tested using new test data. At last, the performance of the model is assessed using a range of statistical metrics.

2.1. Feature Engineering Unit

This section covers the operational principles and mathematical foundations of the pattern extraction and generation methodologies as well as the decomposition algorithm employed in this research.

2.1.1. Seasonal Decomposition (SD)

Seasonal Decomposition (SD) is a method employed in time series analysis to derive informative and meaningful properties by decomposing a series into multiple components. This approach is especially effective for series with a repeating pattern over time. Consider $y(t)$ as raw time series data with t as the time index. Using Fourier analysis, SD breaks down time series into seasonal ($S(t)$), trend ($T(t)$), and residual ($R(t)$) components (Equation (1)) [42,43]. The trend component $T(t)$ represents the long-term trajectory of the time series. The moving average method is commonly used to estimate $T(t)$, as demonstrated by Equation (2), which describes the basic pattern or trend in the data:

$$y(t) = T(t) + S(t) + R(t) \tag{1}$$

$$T(t) = \frac{1}{n}\sum_{i=-k}^{k} y(t+i) \tag{2}$$

where n signifies the total number of observations in the moving window and k denotes the half-window size.

The seasonal component $S(t)$ shows the regular trend over a set period, and can be determined using Equation (3). After taking out the trend and seasonal components, the residual component $R(t)$ describes the anomalies or "noise" that remains in the series.

$$S(t) = Y(t) - T(t) \tag{3}$$

Seasonal decomposition produces a composite of these three components that shows the fundamental patterns in the time series. Seasonality analysis has the potential to offer significant insights regarding the seasonal variations of wind speed within a particular geographic area. This analysis detects patterns in wind speed and direction influenced by weather, terrain, and location by evaluating historical wind data over months or seasons, which helps the prediction model to perform better.

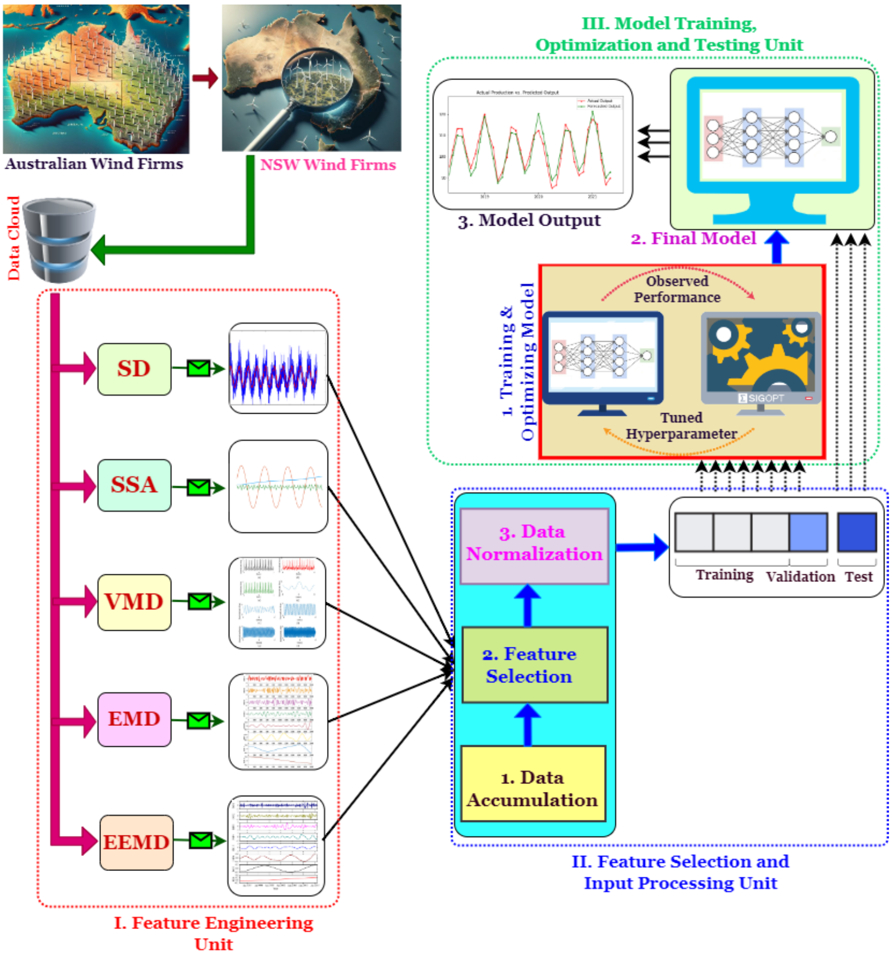


Figure 1. Schematic diagram of the proposed framework for FE-assisted wind energy forecasting.

2.1.2. Singular Spectrum Analysis (SSA)

An effective method for decomposing and analysing time series data is Singular Spectrum Analysis (SSA). It is especially helpful for extracting and splitting significant components from time series data, such as trends, oscillations, and noise. The SSA process consists of four basic steps: Embedding, Decomposition, Grouping, and Diagonal Averaging [44]. First, the time series is transformed into a trajectory matrix. A trajectory matrix C of dimensions $L \times K$, where $K = N - L + 1$ is created following given a time series $y(t) = \{y_1, y_2, \ldots, y_N\}$ and selected window length L $(1 < L < N)$ [44,45].

$$C = \begin{pmatrix} y_1 & y_2 & \cdots & y_K \\ y_2 & y_3 & \cdots & y_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & \cdots & y_N \end{pmatrix} \tag{4}$$

The decomposition and analysis of time series data relies on the matrix C generated by arranging delayed vectors.

The trajectory matrix undergoes Singular Value Decomposition (SVD) in the subsequent stage known as decomposition, which is carried out by applying the following expression:

$$B = U\Sigma V^*. \tag{5}$$

Here, U and V are orthogonal matrices and Σ is a diagonal matrix containing singular values. In the grouping stage, SVD-derived decomposed components are categorized into various groups (B_i):

$$B_i = U_i \Sigma_i V_i^* \tag{6}$$

where each B_i captures a unique dynamic aspect of the original time series.

During the last stage, the grouped components are reassembled into a time series using diagonal averaging. To ensure accurate reconstruction from the deconstructed components, $\hat{y}_i(t)$ is computed for each i-th component using the following procedure:

$$\hat{y}_i(t) = \begin{cases} \frac{1}{t}\sum_{j=1}^{t} B_i(j, t-j+1), & \text{for } 1 \leq t < L, \\ \frac{1}{L}\sum_{j=1}^{L} B_i(j, t-j+1), & \text{for } L \leq t \leq T-L+1, \\ \frac{1}{T-t+1}\sum_{j=t-T+L}^{L} B_i(j, t-j+1), & \text{for } T-L+1 < t \leq T. \end{cases} \tag{7}$$

This is followed by shifting the spectrum of the analytical signal to baseband using a complex exponential which aligns the center frequencies of all modes to zero, facilitating the isolation of individual modes. The SSA method improves the consistency and dependability of pattern recognition methodologies by breaking a signal down into its components, which reveals patterns and structures that might not be identified by other methods.

2.1.3. Variational Mode Decomposition (VMD)

The VMD algorithm breaks down a complicated signal into a group of Intrinsic Mode Functions (IMFs) with certain frequency domain sparsity qualities. This technique is a non-recursive, adaptive, and quasi-orthogonal decomposition approach used to address variational data challenges. The method uses a variational model to identify solutions inside a constraint variational model, leading to advanced signal decomposition [31,34]. The entire VMD process consists of a series of interconnected steps, which are detailed in the following paragraph. Initially, the actual signal $y(t)$ is represented as a sum of amplitude and frequency modulation characteristics in Equation (8) in order to capture the modulatory behavior of the signal [34,35]. Subsequently, the Hilbert transform is applied to each mode in order to obtain an analytical signal, facilitating the determination of the mode's instantaneous frequency (Equation (9)).

$$y_i(t) = A_i(t)\cos(\phi_i(t)) \tag{8}$$

$$\left[\delta(t) + \frac{j}{\pi t}\right] * y_i(t) \tag{9}$$

Following that, the spectrum of the analytical signal is shifted to the baseband through the use of the complex exponential shown in Equation (10). This operation sets all modes' center frequencies to zero, making it easier to isolate each mode individually. Then, the constraint provided in Equation (11) guarantees that the original signal is reconstructed using the sum of all decomposed modes. This balancing act distinguishes the modes without any spectral overlap, which is essential for complex signal analysis.

$$\left[\delta(t) + \frac{j}{\pi t}\right] * y_i(t)\left(e^{-j\omega_i t}\right) \tag{10}$$

$$\min_{\{u_i\},\{\omega_i\}} \quad \sum_{i=1}^{I}\left\| \left[\delta(t) + \frac{j}{\pi t}\right] * u_i(t) \cdot e^{-j\omega_i t}\right\|_2^2 \quad \text{such that,} \quad \sum_{i=1}^{I} p_i = y(t) \tag{11}$$

In the above equations, $p_i = \{p_1, p_2, p_3, \ldots, p_l\}$ represents the IMFs generated from the origial signal and $\omega_i = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_l\}$ represents the central frequency of the IMFs.

Equation (12) represents the Lagrangian within the VMD framework, which is essential for solving the optimization problem. The regularisation parameter α regulates the bandwidth of the extracted modes.

$$L(\{p_i\},\{\omega_i\},\lambda(t)) = \alpha\sum_{i=1}^{I}\left\| \left[\delta(t) + \frac{j}{\pi t}\right] * p_i(t) \cdot e^{-j\omega_i t}\right\|_2^2 \tag{12}$$

Equation (13) iteratively updates the modes p_i and their center frequencies ω_i to ensure that the decomposition matches the signal's instantaneous frequency. According to the energy distribution of the modes, Equation (14) finds the new ω_i values. Lastly, the Lagrange multiplier λ is adjusted in Equation (15) such that the original signal can be reconstructed by adding all the modes.

$$p_i^{m+1}(\omega) = \frac{\hat{y}(\omega) - \sum_{m=1}^{i-1}\hat{p}_m^{m+1}(\omega) + \frac{1}{2}\lambda^m(\omega)}{1 + 2\alpha(\omega - \omega_i^m)^2} \tag{13}$$

$$\omega_i^{m+1} = \frac{\int_0^\infty \omega\left|\hat{p}_i^{m+1}(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{p}_i^{m+1}(\omega)\right|^2 d\omega} \tag{14}$$

$$\lambda^{m+1}(t) = \lambda^m(t) + \tau\left(y(t) - \sum_{i=1}^{I} p_i^{m+1}(t)\right) \tag{15}$$

Equation (16) represents the convergence criterion, which ensures steady mode updates by comparing the relative changes between iterations. The algorithm stops to prevent overfitting when the changes fall below γ.

$$\sum_{i=1}^{I}\frac{\left\|u_i^{m+1} - u_i^m\right\|_2^2}{\left\|u_i^m\right\|_2^2} < \gamma \tag{16}$$

2.1.4. Emperical Mode Decomposition (EMD)

The Empirical Mode Decomposition (EMD) algorithm decomposes a signal into intrinsic oscillatory modes known as Intrinsic Mode Functions (IMFs). This iterative process involves several key steps [30,33]:

1. Identify the extrema (local maxima and minima) in the signal.
2. Create upper and lower envelopes by connecting the extrema.
3. Calculate the mean of the envelopes as a function of time.
4. Subtract the mean from the original signal to obtain the first IMF.
5. Iteratively sift the IMF to refine it.

6. Extract subsequent IMFs from the residue.
7. Repeat the process until no further IMFs can be found.

Let us assume that a given signal $y(t)$ is decomposed sequentially into its constituent IMFs. First, local maxima and minima are identified to form upper and lower envelopes. These envelopes are averaged to obtain the local mean $m_1(t)$. The residual $r_1(t)$ is calculated by subtracting m_1^a from the cumulative total $c_0(t)$ of previous proto-IMFs, as shown in Equation (17). Equation (18) demonstrates the process of subtracting $r_1(t)$ from $c_0(t)$ to obtain $c_1(t)$, which is used for the next iteration. For the first iteration,

$$R_1(t) = \frac{(c_0(t) + m_1^a)}{2} = \frac{(m_a + m_b)}{2} - \frac{m_1^a}{2}, \tag{17}$$

$$c_1(t) = c_0(t) - R_1(t) = m_a - \frac{(m_a + m_b)}{2}. \tag{18}$$

The second iteration uses the updated signal $r_1(t)$ from the previous iteration as the input signal. The upper and lower envelopes are updated as a result of recalculating the local maxima and minima. In Equation (19), the residual $r_2(t)$ is calculated using m_2^a and $c_1(t)$ for this iteration. The derivation of $c_2(t)$ from $c_1(t)$ for the following iteration is shown in Equation (20).

$$R_2(t) = \frac{(c_1(t) + m_2^a)}{2} = \frac{(m_a - m_b)}{2} + \frac{m_2^a}{2} \tag{19}$$

$$c_2(t) = c_1(t) - R_2(t) \tag{20}$$

This iterative improvement process is continued until the n-th iteration. Equations (21) and (22) respectively yield the residual $r_n(t)$ and cumulative sum $c_n(t)$ produced in the same way. This iterative modification continues until the residual $r_n(t)$ converges to zero, which indicates that the proto-IMF is complete and meets the IMF criteria. Convergence is indicated by the limit as n approaches infinity ($\lim_{n\to\infty}$).

$$R_n(t) = \frac{(C_{n-1}(t) + M_n^a)}{2} \tag{21}$$

$$C_n(t) = C_{n-1}(t) - R_n(t) \tag{22}$$

$$\lim_{n\to\infty} r_n(t) = 0 \tag{23}$$

$$\lim_{n\to\infty} c_n(t) = \lim_{n\to\infty} [c_{n-1}(t) - r_n(t)] \tag{24}$$

The first step in determining the IMF is to compute each proto-IMF using the set of equations in Equation (25). To obtain the final IMF ($IMF(t)$), the proto-IMFs are added together to form a single function that represents the inherent oscillatory mode of the original signal $y(t)$.

$$\left.\begin{aligned} I_1(t) &= c_1(t) - c_0(t) \\ I_2(t) &= c_2(t) - c_1(t) \\ &\vdots \\ I_n(t) &= c_n(t) - c_{n-1}(t) \end{aligned}\right\} \tag{25}$$

$$IMF(t) = I_1(t) + I_2(t) + \ldots + I_n(t) \tag{26}$$

2.1.5. Ensemble Empirical Mode Decomposition (EEMD)

The Ensemble Empirical Mode Decomposition (EEMD) is an enhanced version of the EMD that has been specifically developed to address the inherent limitations of the original EMD. In the EMD, the mode-mixing problem arises when signals of similar magnitudes are distributed among different IMFs or when signals of significantly different scales are stored in a single IMF, potentially causing major decomposition errors. The EEMD is differentiated from the EMD by the deliberate introduction of white noise into the signal. This noise aids in the decomposition process by providing a uniform reference scale across the signal, reducing the possibility of mode mixing by identifying the true scales of the data. Furthermore, the EEMD procedure is distinguished from VMD by employing white noise to better identify inherent signal scales, while VMD uses an optimization problem to extract modes with specific bandwidths. The following steps outline the method for resolving the EMD mode-mixing issue [30,32,36,37]:

1. Introduce a white noise series

$$y_{new}(t) = y(t) + w(t) \tag{27}$$

   to the original power series, where $y(t)$ represents the original time series data and $w(t)$ is the white noise series, then determine the EMD related components.
2. Determine the local maxima and minima of $y_{new}(t)$.
3. Construct the upper envelope $y_{max}(t)$ and lower envelope $y_{min}(t)$.
4. Compute the mean of the two envelopes using Equation (28) and the difference between the new data point and the average of the envelopes using Equation (29).

$$m(t) = \frac{y_{max}(t) + y_{min}(t)}{2} \tag{28}$$

$$d(t) = y_{new}(t) - m(t) \tag{29}$$

5. Repeat Steps 1–3 until $m(t)$ is less than or equal to a preset threshold δ indicating the allowable error by substituting $d(t)$ for $y_{new}(t)$. Assigning $c_1(t) = d(t)$ as the initial EMD component of $y_{new}(t)$, the residual is

$$r_1 = y_{new}(t) - c_1(t). \tag{30}$$

6. The time series y(t) can be expressed as a sum of its IMFs and a residue, as follows:

$$y(t) = \sum_{i=1}^{n} c_i^m + r^x \tag{31}$$

   where c_i^m denotes the IMFs and r^x is the final residue.

2.2. Feature Selection and Input Processing Unit

After the creation of the features using the method described in Section 2.1, the next step is to choose the associated features for the prediction model. The number of features significantly affects the prediction model's performance. An increase in the number of features enables the model to make more accurate predictions for the output based on the inputs. However, as additional features can cause overfitting and increase compute time, determining the ideal number of features is important to the output of the deep learning model. Numerous feature selection techniques, such as RF, RFE, DT, and others, are popular with data scientists. In this research, we evaluated the linear relationship between each pair of attributes using the efficient Pearson's correlation coefficient (PCC). Thanks to its simplicity and ease of understanding, the PCC is a useful tool in data analysis and feature selection that is frequently used to measure linear connections between variables. The formula for the PCC is as follows [46]:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{32}$$

where x_i and y_i indicate the variables' values, $\overline{x}$ and $\overline{y}$ represent their mean values, and n denotes the number of data points. The model uses items with high Pearson correlation coefficients, indicating strong linear correlations.

After choosing the most correlated features, normalizing the data is important for aligning the input features' scales, which helps the deep learning model to converge. In our proposed framework, the MinMaxScaler function is used to convert each feature to a specified range, typically [0, 1]. The transformation is defined by the mathematical expression [47]

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}, \tag{33}$$

where the parameter X denotes the initial value, X_{min} and X_{max} represent the minimum and maximum values, respectively, and X_{norm} signifies the normalised value. The MinMaxScaler function adjusts features into a consistent range to ensure that the deep learning model can learn patterns from normalised data without bypassing it as the scale increases. Next, the normalized dataset is separated into training, validation, and testing sets for model input.

2.3. Model Training, Optimization, and Testing Unit

The diagram in Figure 2 illustrates the proposed architecture of the Sequential Memory Unit (SMU) utilized for wind power generation prediction. This network's major benefit is its built-in memory unit; this allows it to understand long-term dependencies in sequential data, which is difficult for regular Recurrent Neural Networks [28,48]. These cells' special gating mechanism makes it possible for them to selectively store or forget information, making them very useful for complex tasks. Similar to other NNs, the proposed SMU works the same in training. One phase of data propagation includes both a forward pass in which input data are processed to generate output, and a backward pass in which gradients are generated and propagated back through the network for parameter adjustments. Through this dual process, the network learns from data and adjusts weights and biases to reduce error. The proposed model's forward and backward propagation mechanisms are mathematically detailed below.

The SMU cell is an essential component of the proposed data-driven network. It consists of two states, the cell state and hidden state, and three gates, the forget gate, input gate, and output gate. Each gate has a specific function in the cell's operation and overall prediction performance. By analyzing both the prior hidden state (h_{t-1}) and the current input (x_t), the forget gate (f_t) can select which information from the cell state (C_{t-1}) should be kept and which should be discarded. It uses Equation (34) to generate values ranging from 0 (showing complete forgetfulness) to 1 (representing complete retention) for each component of the cell state. Next, using Equation (35), the input gate (i_t) determines the amount by which each component of the cell state should be changed, then outputs values that fall between 0 and 1. According to Equation (36), the output gate (o_t) is responsible for determining the next hidden state (h_t) by taking into account the current state of the cell and the input.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{34}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{35}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{36}$$

Then, the SMU uses the current input and the prior hidden state to compute a candidate cell state ($\tilde{C}_t$) employing Equation (37). This Candidate Cell State is utilized to update the cell state (C_t) through the interaction of the forget and input gates via Equation (38). After passing the cell state through the hyperbolic tangent (tanh) function and modulating it with the output gate, the hidden state (h_t) is finally calculated using Equation (39). These gates and states allow the SMU to efficiently capture and handle information during forward data propagation.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{37}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{38}$$

$$h_t = o_t * \tanh(C_t) \tag{39}$$

In the above, σ represents the sigmoid function, W and b are the weights and biases, respectively, and $*$ denotes element-wise multiplication.

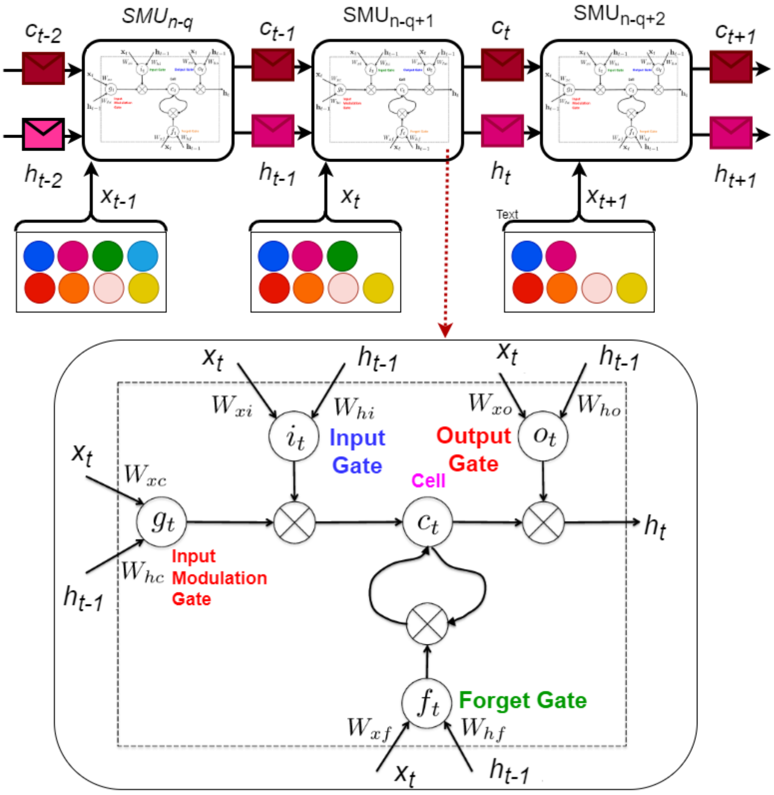


Figure 2. Overview of the proposed deep learning model architecture.

During the process of backward propagation, the gradients of the loss function are computed using the weights and biases. In order to achieve this, the derivatives of the loss function need to be computed for each parameter in the network. General gradients for the SMU cell parameters are

$$\frac{\partial L}{\partial W} = \sum_t \frac{\partial L_t}{\partial W}, \tag{40}$$

$$\frac{\partial L}{\partial b} = \sum_t \frac{\partial L_t}{\partial b}, \tag{41}$$

where L represents the loss function and the derivatives are calculated for each time step and then summed.

The weights and biases in the SMU are updated using optimization algorithms such as stochastic gradient descent (SGD). The updated equations are:

$$W_{new} = W_{old} - \eta \cdot \frac{\partial L}{\partial W}, \tag{42}$$

$$b_{new} = b_{old} - \eta \cdot \frac{\partial L}{\partial b}, \tag{43}$$

where η is the learning rate. In our proposed framework, the SMU cells are connected in series and data propagation and weight-bias updates are based on the previous formulas. However, the performance of the model during testing is dependent on the parameters of each cell as well as on the overall prediction model. Thus, optimal performance requires finding the best parameters. For this reason, hyperparameter calibration and optimization are essential for deep learning models; yet, they have frequently been neglected in previous research. In this study, the best model parameters were found using the GridSearchCV algorithm. Table 2 provides the search space. After determining the best model parameters, the training and validation data sets were used to train the model.

Table 2. Hyperparameter tuning search space for model optimization.

Parameters	Tuning Parameter Boundaries
No. of Units	32, 64, 96, 128, 160, 192
Batch Size	16, 32, 48, 64, 80
No. of Epochs	25, 50, 75, 100, 125
Learning Rate	0.0002, 0.0004, 0.0006, 0.0008, 0.001
Window Length	160, 192, 224, 256, 288, 320

The test data were used to validate the performance of the deep learning model after the model was trained and the final prediction ready for display. The performance of the proposed model was evaluated using performance metrics such as the MAE, MAPE, MSE, RMSE, and R-squared score. The mathematical expressions of the performance matrices are provided below [28–33]:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{44}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{45}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{46}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{47}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{48}$$

where y_i stands for the real values, $\hat{y}_i$ for the predicted values, and $\bar{y}_i$ for the mean of the actual values, while n indicates the total number of observations. To maintain accuracy and clarity in the formulation and computation of the error metrics, these symbols are applied uniformly across every metric.

3. Results and Discussion

An original dataset on wind energy generation from the Australian Energy Market Operator (AEMO) was the main source of data for the analysis part of this study. This dataset covers the period from 1 May 2018 to 31 December 2018, and provides a full picture of wind energy generation during this time [49]. To focus the investigation on a specific geographic location, we selectively chose data only for wind farms in New South Wales (NSW), Australia. To make sure that the time precision for these analyses was very fine, the dataset was carefully processed to add up the values of wind generation every five minutes. Figure 3 shows the aggregated wind generation patterns for the analysis and interpretation in this research. According to the graphic, the pattern of wind generation is very nonlinear and fluctuates between a range of 0 to 1000 megawatts (MW). Following this, the data were evaluated and checked for missing values and anomalies that do not exist in the dataset. The dataset was finally split into training, validation, and testing sets at a ratio of 80:10:10.

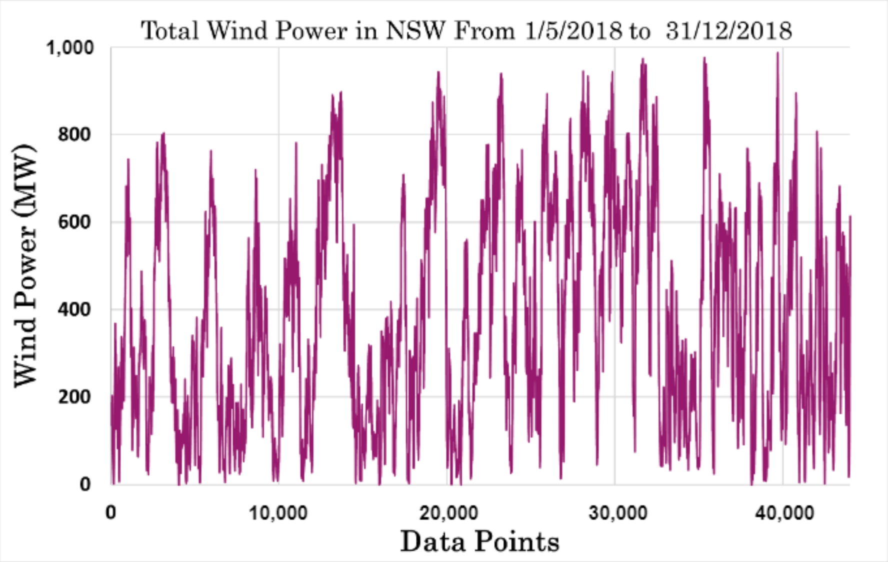


Figure 3. The wind power generation profile of New South Wales, Australia from May to December 2018.

An extensive search was conducted to assess the performance of the different combinations described in Section 2.1 as part of the examination of various FE approaches. We thoroughly examined numerous combinations with the algorithms and chose the best six FE combinations (F1–F6). These units decompose only the wind generation data to generate features: F1 uses the SD; F2 combines the SD and the SSA; F3 includes the VMD with F2; F4 combines the EMD with the SD, SSA, and VMD; F5 adds the EEMD with F4; and F6 removes the EMD from F5. After implementing the feature selection technique described in Section 2.2, the performance indicators showed that the F4 combination with SD + SSA + VMD + EMD yielded the best results. Figure 4 shows the performance metrics for the proposed forecasting model. For the aforementioned FE combination, the proposed model provides the lowest MAE of 8.76, MSE of 139.48, RMSE of 11.81, and MAPE of 4.85%, with an outstanding R-squared value of 0.997.

Despite F5 and F6 having more characteristics than F4, our findings showed that F4 was the best for this dataset; thus, the F4-prepared dataset was taken into consideration for additional analysis. Figure 5 shows the expected results when using the F1 through F6 combinations. This graph further supports the choice of F4 as the best FE algorithm.

The number of SMU units in the proposed model was systematically changed from 32 to 192 (with a step size of 32) in order to examine its impact on performance. Figure 6 depicts the related performance metrics; in particular, the figure shows that performance improves up to a certain point, beyond which further unit increases result in a progressive rise in errors. This observed phenomenon is consistent with the concept that an overly complicated model with a large number of units can be susceptible to overfitting [50–52]. The graph shows that 128 units optimize the model's complexity and predictive accuracy.

Next, we explored the impact of batch size and window length (WL) on the performance of the proposed SMU model. The batch size was varied from 16 to 80 with a step size of 16, while the window length (WL) was varied from 160 to 320 with a step size of 32. The findings suggest that the model attains its best performance when using a batch size of 32, as illustrated in Figure 7. The performance of the proposed model for various WLs is presented in Figure 8, from which it is clear that the ideal size for the data sample window in this forecasting model is 288.

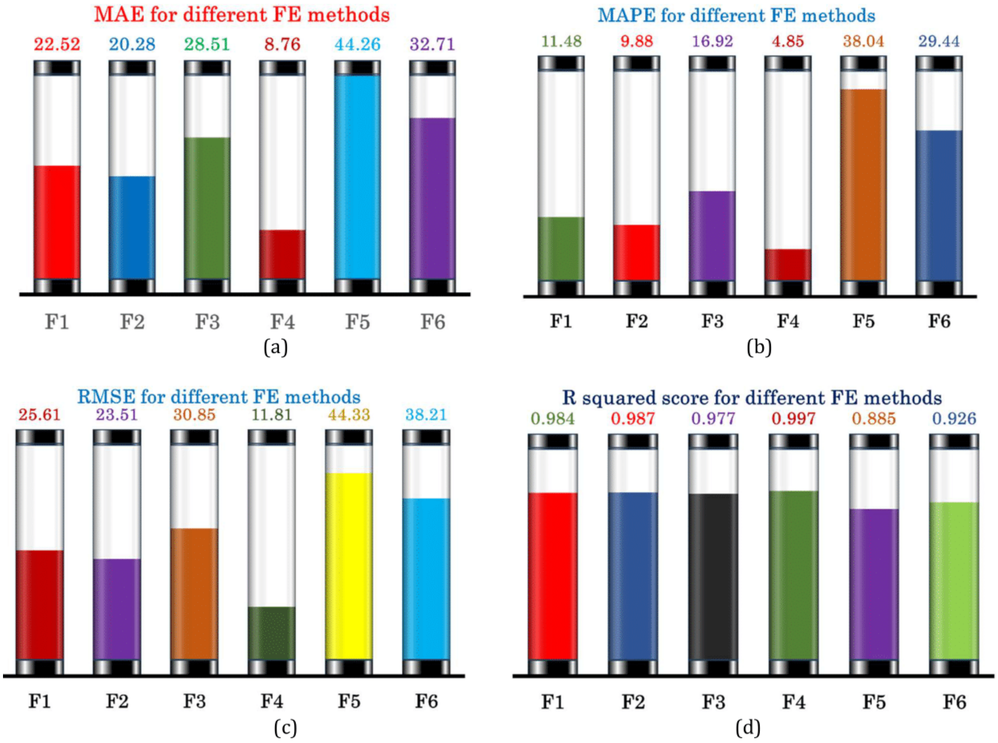


Figure 4. Performance metrics such as (**a**) MAE, (**b**) MAPE, (**c**) RMSE and (**d**) R squared score of the proposed data-driven model for different FE combinations.

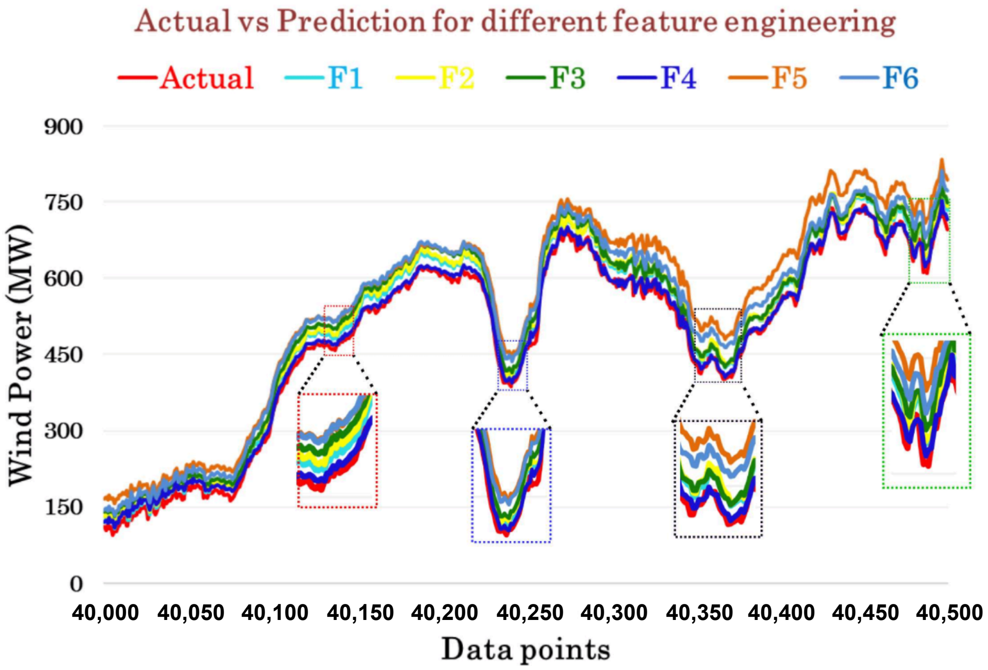


Figure 5. Actual and predicted wind generation for different FE combinations.

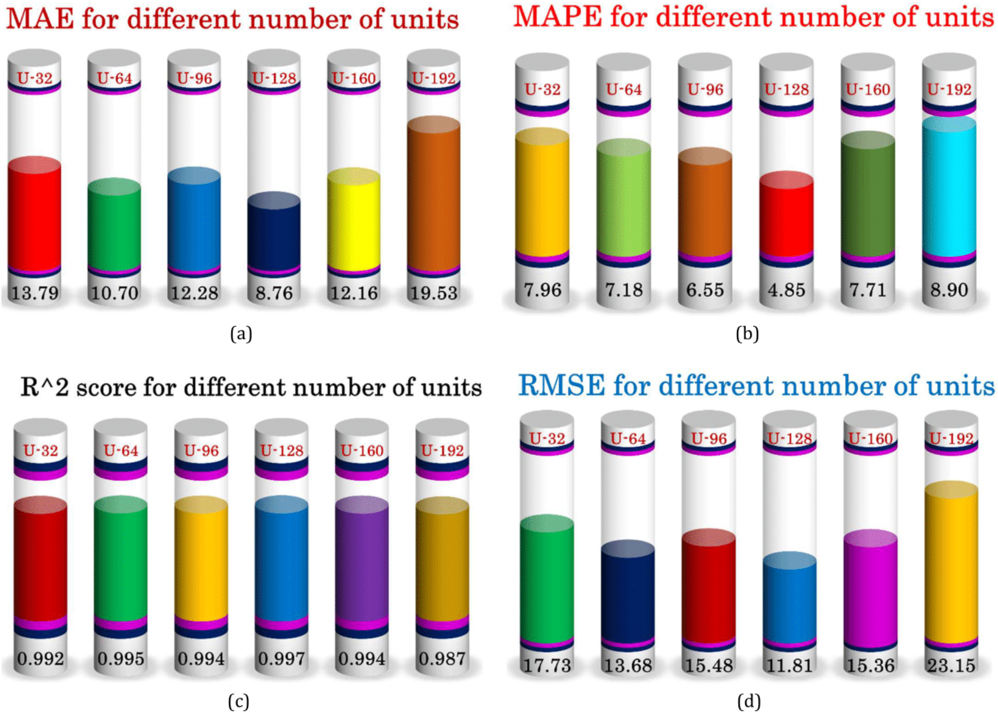


Figure 6. Performance metrics such as (**a**) MAE, (**b**) MAPE, (**c**) RMSE and (**d**) R squared score comparison for different numbers of units of the proposed SMU.

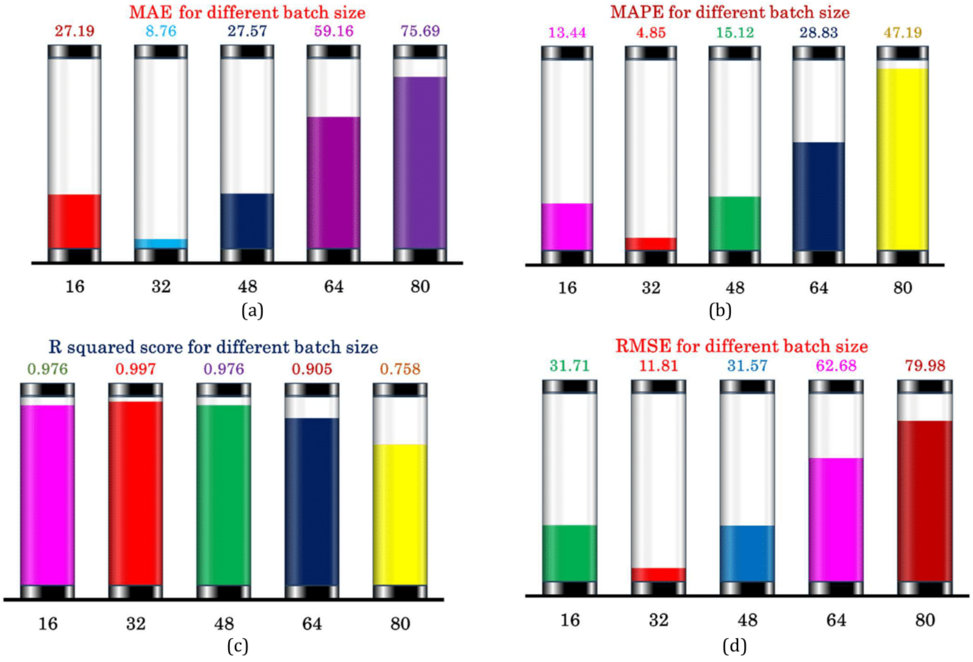


Figure 7. Error comparison for different batch sizes during training of the proposed model.

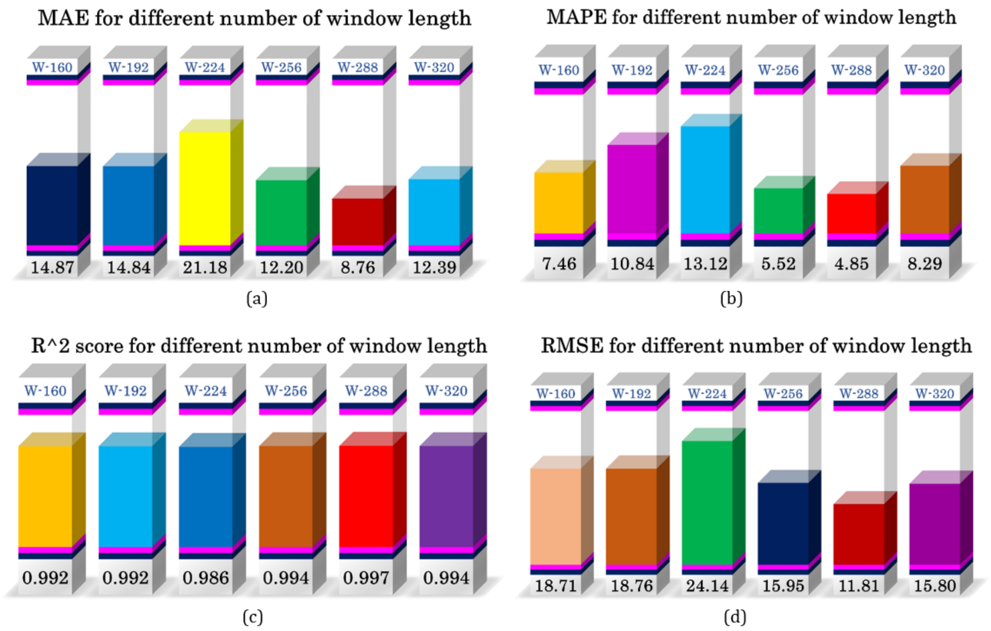


Figure 8. Analysis of the performance metrics for different window lengths.

The learning rate affects the training dynamics of a deep learning model. In machine learning and deep learning, the learning rate is a hyperparameter that controls the size of the optimization step, affecting how quickly a model converges and how well it navigates the loss landscape to find an optimal solution. Figure 9 illustrates the results for the different performance metrics when the learning rate was systematically varied from 0.0002 to 0.001 with a step size of 0.0002. It can be observed that the optimal learning rate for this model is 0.0008.

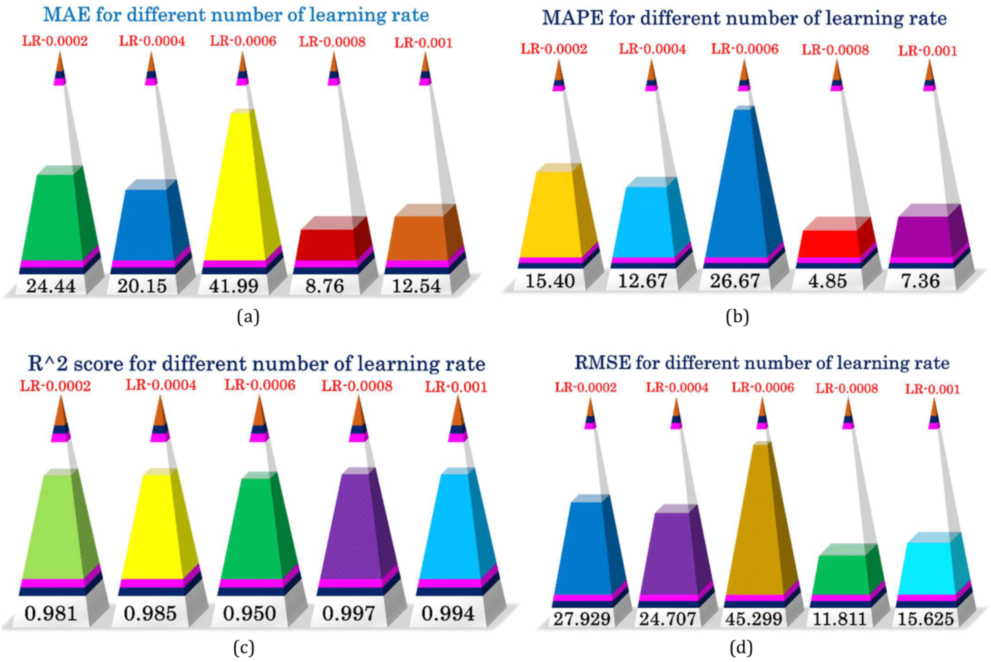
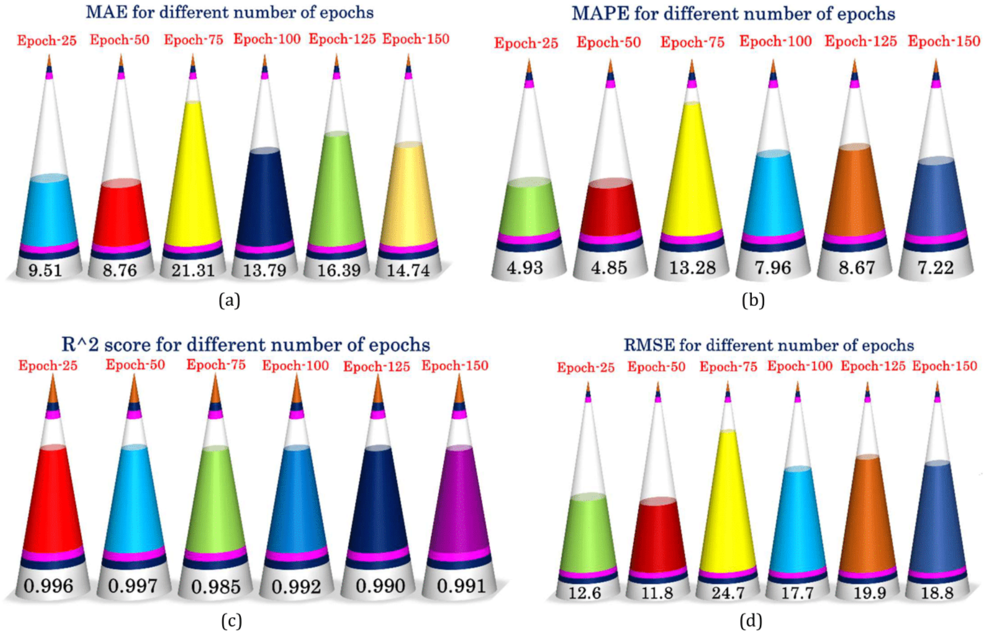


Figure 9. Analysis of the performance metrics for different learning rates.

The final step is to look at the number of epochs in order to find the best number of training sessions with the given training data. The MAE, MAPE, RMSE, and R squared scores were examined after changing the number of epochs from 25 to 150 with a step size of 25, and the results are presented in Figure 10. Figure 10 shows that 50 epochs is the ideal number for achieving the best performance metrics. According to these results, the model may be unable to identify the underlying patterns in the data if there are too many or too few training epochs in the process. In addition, larger epochs require more training time.



(a) (b) (c) (d)

Figure 10. Error comparison for different numbers of epochs during training of the proposed model.

The optimal hyperparameters and associated error of the model after individually investigating the hyperparameters are shown in Table 3; the results for the proposed model are comparable to those of the articles discussed in the literature review. Figures 11 and 12 provide a comprehensive comparison of the proposed model's performance against a wide range of deep learning frameworks, including CNN, TCN, Bi-TCN, ANN, and Bi-LSTM networks. These models were chosen for comparison in this study for multiple reasons: first, as these models are good at capturing temporal dependencies and patterns, they are well known and widely used in the field of time series data analysis and forecasting; second, their memory-enriched architectures allow them to efficiently memorize and learn from previous data sequences, which is quite similar to the proposed model. The metrics presented in Figure 11 demonstrate the proposed model's superiority across a range of performance factors, including improvements in accuracy and efficiency. Specifically, the CNN, ANN, and TCN models performed less than ideally, as indicated by their lower performance metrics in Figure 11 and in the prediction graph in Figure 12. The primary reason for this is that the complex and highly nonlinear patterns in the data (Figure 3) are not well captured by the aforementioned models. According to the following graphs, the proposed model outperforms the comparative models, indicating its robustness and potential to forecast accurately.

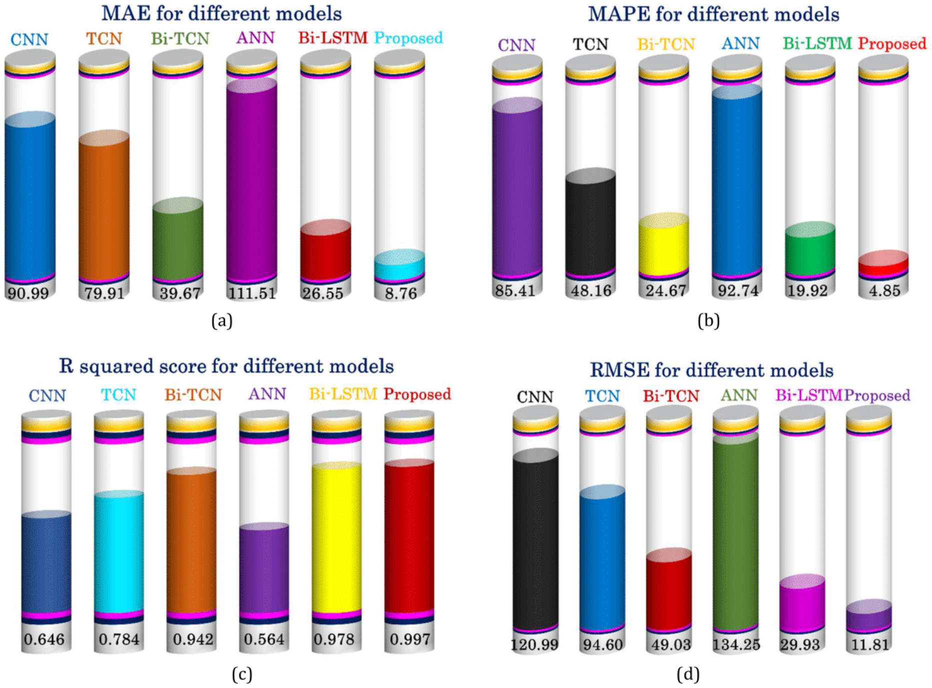


Figure 11. Analysis of the performance metrics for the proposed DL model in comparison with other models.

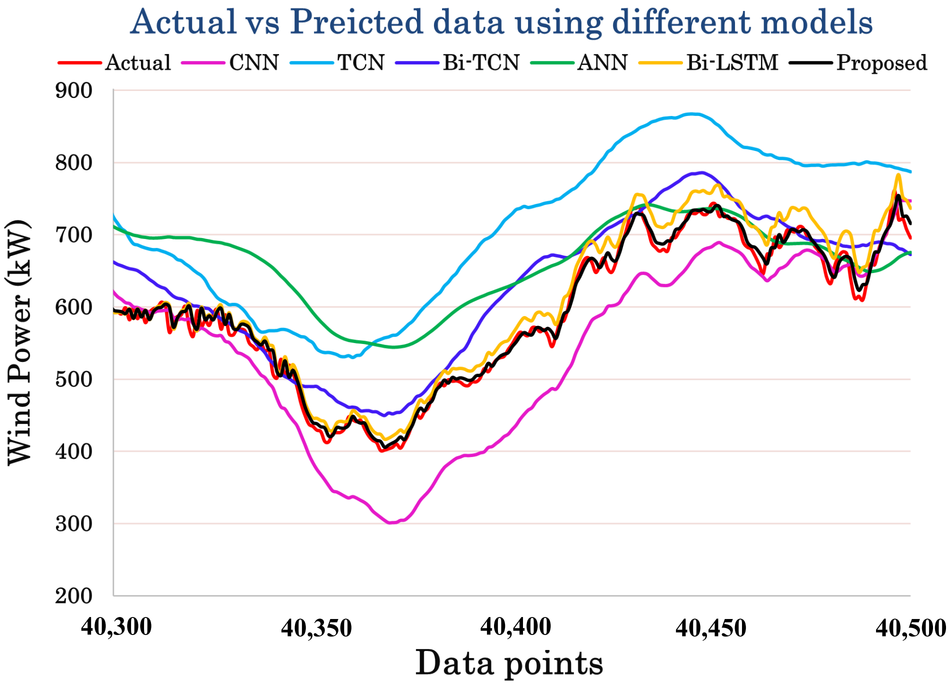


Figure 12. Actual and predicted wind generation for different comparable DL models and the proposed model.

For the better understanding, the compared model information and achieved performance indicators are listed in Table 3.

Table 3. Comparison of different models based on model attributes and performance metrics.

Section	Attributes/Metrics	CNN	TCN	Bi-TCN	ANN	Bi-LSTM	Proposed
Model	Units	128	64	128	128	64	128
	Learning rate	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
	Epochs	100	100	50	75	100	50
	Window length	288	288	288	288	288	288
	Batch size	32	32	32	32	32	32
Performance	MAE	90.98	79.90	39.67	111.51	26.55	**8.76**
	MAPE	85.41%	48.16%	24.67%	92.74%	19.92%	**4.85%**
	MSE	14638	8948.9	2403.5	18024	895.5	**139.49**
	RMSE	120.99	94.60	49.03	134.25	29.92	**11.81**
	R^2 score	0.645	0.783	0.942	0.564	0.978	**0.997**

4. Conclusions and Future Directions

This article has investigated the state of the art in wind energy forecasting while emphasizing the potential benefits of strategic feature engineering in conjunction with intelligent deep learning. By utilizing a cutting edge deep learning model and conducting extensive analysis, this investigation effectively exploits real-time wind generation data from New South Wales, Australia to generate short-term predictions. Traditional methods primarily rely on meteorological factors, whereas our unique approach uses only historical data and hybrid feature engineering. The developed model outperforms other contemporary deep learning and hybrid models. Its accuracy was confirmed through important metrics including the MAE, MSE, RMSE, R-squared score, and MAPE, demonstrating the significant role of advanced data analysis and feature development in improving forecasting accuracy.

With careful consideration, the discipline of wind power forecasting holds immense promise for additional improvements. Advanced hybrid feature engineering approaches could revolutionize feature synthesis and reveal complex data patterns. Additionally, integrating feature selection strategies within a modified/cascaded deep learning network with attention layers might improve outcomes and provide an innovative aspect to the current research trend. Integrating security measures against cyberattacks and fake data injection is another intriguing concept that may be used with this research. These developments could lead to more robust, dependable, and effective forecasting solutions by enhancing model prediction capabilities and discovering wind energy dynamics.

Author Contributions: Conceptualization, M.A.H.; methodology, M.A.H.; software, M.A.H. and M.J.H.; validation, M.A.H.; formal analysis, M.A.H.; investigation, M.A.H.; resources, M.A.H.; data curation, M.A.H.; writing—original draft preparation, M.A.H.; writing—review and editing, M.A.H. and M.J.H.; visualization, M.A.H. and M.J.H.; supervision, M.J.H.; project administration, M.J.H.; funding acquisition, M.J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This research uses confidential data that can be provided upon request subject to approvals and restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Swain, R.B.; Karimu, A. Renewable electricity and sustainable development goals in the EU. *World Dev.* **2020**, *125*, 104693. [CrossRef]
2. Cantarero, M.M.V. Of renewable energy, energy democracy, and sustainable development: A roadmap to accelerate the energy transition in developing countries. *Energy Res. Soc. Sci.* **2020**, *70*, 101716. [CrossRef]
3. Ahmed, S.D.; Al-Ismail, F.S.; Shafiullah, M.; Al-Sulaiman, F.A.; El-Amin, I.M. Grid integration challenges of wind energy: A review. *IEEE Access* **2020**, *8*, 10857–10878. [CrossRef]
4. Georgilakis, P.S. Technical challenges associated with the integration of wind power into power systems. *Renew. Sustain. Energy Rev.* **2008**, *12*, 852–863. [CrossRef]
5. Pathak, A.; Sharma, M.; Bundele, M. A critical review of voltage and reactive power management of wind farms. *Renew. Sustain. Energy Rev.* **2015**, *51*, 460–471. [CrossRef]
6. Ourahou, M.; Ayrir, W.; Hassouni, B.E.; Haddi, A. Review on smart grid control and reliability in presence of renewable energies: Challenges and prospects. *Math. Comput. Simul.* **2020**, *167*, 19–31. [CrossRef]
7. Blaabjerg, F.; Yang, Y.; Ma, K.; Wang, X. Power electronics—The key technology for renewable energy system integration. In Proceedings of the 2015 International Conference on Renewable Energy Research and Applications (ICRERA), Palermo, Italy, 22–25 November 2015; pp. 1618–1626. [CrossRef]
8. Blaabjerg, F.; Ma, K. Future on Power Electronics for Wind Turbine Systems. *IEEE J. Emerg. Sel. Top. Power Electron.* **2013**, *1*, 139–152. [CrossRef]
9. Beaudin, M.; Zareipour, H.; Schellenberglabe, A.; Rosehart, W. Energy storage for mitigating the variability of renewable electricity sources: An updated review. *Energy Sustain. Dev.* **2010**, *14*, 302–314. [CrossRef]
10. Suberu, M.Y.; Mustafa, M.W.; Bashir, N. Energy storage systems for renewable energy power sector integration and mitigation of intermittency. *Renew. Sustain. Energy Rev.* **2014**, *35*, 499–514. [CrossRef]
11. Kashem, S.B.A.; Chowdhury, M.E.; Khandakar, A.; Ahmed, J.; Ashraf, A.; Shabrin, N. Wind power integration with smart grid and storage system: Prospects and limitations. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
12. Khalid, M. Smart grids and renewable energy systems: Perspectives and grid integration challenges. *Energy Strategy Rev.* **2024**, *51*, 101299. [CrossRef]
13. Banik, A.; Behera, C.; Sarathkumar, T.V.; Goswami, A.K. Uncertain wind power forecasting using LSTM-based prediction interval. *IET Renew. Power Gener.* **2020**, *14*, 2657–2667. [CrossRef]
14. Dong, W.; Sun, H.; Mei, C.; Li, Z.; Zhang, J.; Yang, H. Forecast-driven stochastic optimization scheduling of an energy management system for an isolated hydrogen microgrid. *Energy Convers. Manag.* **2023**, *277*, 116640. [CrossRef]
15. Tarek, Z.; Shams, M.Y.; Elshewey, A.M.; El-kenawy, E.S.M.; Ibrahim, A.; Abdelhamid, A.A.; El-dosuky, M.A. Wind Power Prediction Based on Machine Learning and Deep Learning Models. *Comput. Mater. Contin.* **2023**, *75*, 716–732. [CrossRef]
16. Wang, Y.; Zou, R.; Liu, F.; Zhang, L.; Liu, Q. A review of wind speed and wind power forecasting with deep neural networks. *Appl. Energy* **2021**, *304*, 117766. [CrossRef]
17. Zhao, L.; Nazir, M.S.; Nazir, H.M.J.; Abdalla, A.N. A review on proliferation of artificial intelligence in wind energy forecasting and instrumentation management. *Environ. Sci. Pollut. Res.* **2022**, *29*, 43690–43709. [CrossRef] [PubMed]
18. Mostafa, K.; Zisis, I.; Moustafa, M.A. Machine learning techniques in structural wind engineering: A State-of-the-Art Review. *Appl. Sci.* **2022**, *12*, 5232. [CrossRef]
19. Liu, Y.; Wang, Y.; Wang, Q.; Zhang, K.; Qiang, W.; Wen, Q.H. Recent advances in data-driven prediction for wind power. *Front. Energy Res.* **2023**, *11*, 1204343. [CrossRef]
20. Tang, M.; Zhao, Q.; Ding, S.X.; Wu, H.; Li, L.; Long, W.; Huang, B. An improved lightGBM algorithm for online fault detection of wind turbine gearboxes. *Energies* **2020**, *13*, 807. [CrossRef]
21. Malakouti, S.M. Use machine learning algorithms to predict turbine power generation to replace renewable energy with fossil fuels. *Energy Explor. Exploit.* **2023**, *41*, 836–857. [CrossRef]
22. Wu, Z.; Luo, G.; Yang, Z.; Guo, Y.; Li, K.; Xue, Y. A comprehensive review on deep learning approaches in wind forecasting applications. *CAAI Trans. Intell. Technol.* **2022**, *7*, 129–143. [CrossRef]
23. Gultepe, I.; Sharman, R.; Williams, P.D.; Zhou, B.; Ellrod, G.; Minnis, P.; Trier, S.; Griffin, S.; Yum, S.S.; Gharabaghi, B.; et al. A review of high impact weather for aviation meteorology. *Pure Appl. Geophys.* **2019**, *176*, 1869–1921. [CrossRef]
24. Li, J.; Li, Z.; Jiang, Y.; Tang, Y. Typhoon resistance analysis of offshore wind turbines: A review. *Atmosphere* **2022**, *13*, 451. [CrossRef]
25. Chen, Y.; Yu, S.; Islam, S.; Lim, C.P.; Muyeen, S. Decomposition-based wind power forecasting models and their boundary issue: An in-depth review and comprehensive discussion on potential solutions. *Energy Rep.* **2022**, *8*, 8805–8820. [CrossRef]
26. Das, S.; Prusty, B.R.; Bingi, K. Review of adaptive decomposition-based data preprocessing for renewable generation rich power system applications. *J. Renew. Sustain. Energy* **2021**, *13*, 062703. [CrossRef]
27. Shao, Z.; Han, J.; Zhao, W.; Zhou, K.; Yang, S. Hybrid model for short-term wind power forecasting based on singular spectrum analysis and a temporal convolutional attention network with an adaptive receptive field. *Energy Convers. Manag.* **2022**, *269*, 116138. [CrossRef]
28. Kisvari, A.; Lin, Z.; Liu, X. Wind power forecasting—A data-driven method along with gated recurrent neural network. *Renew. Energy* **2021**, *163*, 1895–1909. [CrossRef]

29. da Silva, R.G.; Ribeiro, M.H.D.M.; Moreno, S.R.; Mariani, V.C.; dos Santos Coelho, L. A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting. *Energy* **2021**, *216*, 119174. [CrossRef]
30. Jaseena, K.U.; Kovoor, B.C. Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks. *Energy Convers. Manag.* **2021**, *234*, 113944. [CrossRef]
31. Tian, Z.; Li, H.; Li, F. A combination forecasting model of wind speed based on decomposition. *Energy Rep.* **2021**, *7*, 1217–1233. [CrossRef]
32. Zhu, A.; Zhao, Q.; Wang, X.; Zhou, L. Ultra-short-term wind power combined prediction based on complementary ensemble empirical mode decomposition, whale optimisation algorithm, and elman network. *Energies* **2022**, *15*, 3055. [CrossRef]
33. Parri, S.; Teeparthi, K.; Kosana, V. A hybrid methodology using VMD and disentangled features for wind speed forecasting. *Energy* **2024**, *288*, 129824. [CrossRef]
34. Zhang, L.; Xie, L.; Han, Q.; Wang, Z.; Huang, C. Probability density forecasting of wind speed based on quantile regression and kernel density estimation. *Energies* **2020**, *13*, 6125. [CrossRef]
35. Zhao, Z.; Yun, S.; Jia, L.; Guo, J.; Meng, Y.; He, N.; Li, X.; Shi, J.; Yang, L. Hybrid VMD-CNN-GRU-based model for short-term forecasting of wind power considering spatio-temporal features. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105982. [CrossRef]
36. Sun, S.; Fu, J.; Li, A. A Compound Wind Power Forecasting Strategy Based on Clustering, Two-Stage Decomposition, Parameter Optimization, and Optimal Combination of Multiple Machine Learning Approaches. *Energies* **2019**, *12*, 3586. [CrossRef]
37. Lu, P.; Ye, L.; Sun, B.; Zhang, C.; Zhao, Y.; Teng, J. A new hybrid prediction method of ultra-short-term wind power forecasting based on EEMD-PE and LSSVM optimized by the GSA. *Energies* **2018**, *11*, 697. [CrossRef]
38. He, Y.; Wang, Y. Short-term wind power prediction based on EEMD–LASSO–QRNN model. *Appl. Soft Comput.* **2021**, *105*, 107288. [CrossRef]
39. Yang, M.; Guo, Y.; Huang, Y. Wind power ultra-short-term prediction method based on NWP wind speed correction and double clustering division of transitional weather process. *Energy* **2023**, *282*, 128947. [CrossRef]
40. Xu, X.; Hu, S.; Shi, P.; Shao, H.; Li, R.; Li, Z. Natural phase space reconstruction-based broad learning system for short-term wind speed prediction: Case studies of an offshore wind farm. *Energy* **2023**, *262*, 125342. [CrossRef]
41. Xu, X.; Hu, S.; Shao, H.; Shi, P.; Li, R.; Li, D. A spatio-temporal forecasting model using optimally weighted graph convolutional network and gated recurrent unit for wind speed of different sites distributed in an offshore wind farm. *Energy* **2023**, *284*, 128565. [CrossRef]
42. Prema, V.; Rao, K.U. Time series decomposition model for accurate wind speed forecast. *Renew. Wind Water Sol.* **2015**, *2*, 1–11. [CrossRef]
43. Xu, L.; Ou, Y.; Cai, J.; Wang, J.; Fu, Y.; Bian, X. Offshore wind speed assessment with statistical and attention-based neural network methods based on STL decomposition. *Renew. Energy* **2023**, *216*, 119097. [CrossRef]
44. Wang, C.; Zhang, H.; Ma, P. Wind power forecasting based on singular spectrum analysis and a new hybrid Laguerre neural network. *Appl. Energy* **2020**, *259*, 114139. [CrossRef]
45. Zhao, Y.; Jia, L. A short-term hybrid wind power prediction model based on singular spectrum analysis and temporal convolutional networks. *J. Renew. Sustain. Energy* **2020**, *12*, 056101. [CrossRef]
46. Chen, H.; Chang, X. Photovoltaic power prediction of LSTM model based on Pearson feature selection. *Energy Rep.* **2021**, *7*, 1047–1054. [CrossRef]
47. Zaman, U.; Teimourzadeh, H.; Sangani, E.H.; Liang, X.; Chung, C.Y. Wind speed forecasting using ARMA and neural network models. In Proceedings of the 2021 IEEE Electrical Power and Energy Conference (EPEC), Virtual, 22–31 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 243–248.
48. Ren, L.; Dong, J.; Wang, X.; Meng, Z.; Zhao, L.; Deen, M.J. A data-driven auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3478–3487. [CrossRef]
49. Australian Energy Market Operator. Data Dashboards. 2023. Available online: https://www.aemo.com.au/energy-systems/data-dashboards (accessed on 30 November 2023).
50. Weigend, A.S. On overfitting and the effective number of hidden units. In *Proceedings of the 1993 Connectionist Models Summer School*; Psychology Press: London, UK, 2014; pp. 335–342.
51. Salman, S.; Liu, X. Overfitting mechanism and avoidance in deep neural networks. *arXiv* **2019**, arXiv:1901.06566.
52. Rice, L.; Wong, E.; Kolter, Z. Overfitting in adversarially robust deep learning. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2020; pp. 8093–8104.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.