

Enhanced Wind Energy Forecasting using CEEMDAN and Transformer-BiLSTM with Frequency-based Analysis

Manisha Galphade, Nilkamal More, VB Nikam, Biplab Banerjee, Arvind W. Kiwelekar and Priyanka Sharma

Abstract—Renewable energy is sourced from continually available natural resources. Accurate wind energy forecasting is crucial for effective integration of renewable energy into power grids. In addition, it is essential to ensure the safety of wind turbines by anticipating extreme weather conditions and implementing the necessary precautions. This paper presents a novel approach that combines Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) decomposition with a hybrid neural transformer-BiLSTM network to make wind energy forecasts more accurate. The proposed method first applies CEEMDAN to decompose the original wind energy time series into several sub-series, then divide them into high- and low-frequency components. A transformer model is used to handle the complexity of the high-frequency data, while a BiLSTM network is used for the simpler low-frequency sub-series. Sample entropy is introduced as a criterion for feature selection to ensure that the model focuses on the most informative components of the data. The results of the Transformer and BiLSTM models are aggregated to produce the final wind power prediction. Extensive experiments demonstrate that the suggested methodology significantly overcomes the existing approaches to prediction accuracy and offers an efficient method for wind energy systems to enhance efficiency and stability. This research contributes to this topic by providing a robust and scalable frequency-based wind energy forecasting analysis.

Index Terms—Decomposition, Renewable energy, BiLSTM, Transformer, Sample entropy.

I. INTRODUCTION

HUMAN life comfort is tremendously changing because of advancements in science and technology. However, it generates energy crises and environmental risks [1], [2] such as reducing energy resources and increasing environmental pollutants, which threaten human life. Energy is an

essential material basis for human economic development and social progress that affects all aspects of social life [3]. Wind power is considered one of the fastest-growing sources of electricity generation due to its cost-effective harnessing of kinetic energy. Wind power [4], [5], [6] is a renewable and environmentally friendly form of electricity. As a result, several countries are focusing their efforts on wind power generation [7]. Wind power electricity generation is very volatile due to wind's chaotic and intermittent nature. This could potentially result in massive losses in the energy sector. As a result, providers need to have accurate forecasting skills for wind power [8]. The wind energy market requires technical and economic advancement [9], [10]. For technical advancement, the efforts are focused on how to use the wind at a maximum level, and for economic issues [7], the main objective is to reduce the penalty to gain maximum profit from available resources. India is world's third-largest electricity producer in the 2016, as per the report on 29 Feb 2024 [11].

When wind power prediction often relies on estimating wind speed using proper methodologies. Some literature has employed indirect approaches to estimate wind speed and thus predict wind power [12]. The indirect technique of wind power forecasting involves directly predicting wind power output without first forecasting wind speed. Several researchers have dedicated their efforts to advancing reliable wind power forecasting models. Researchers employ several models for their studies [13], [14] are mainly classified into physical [15], statistical [16] and hybrid [17] classes. The physical model such as, Navier-Stokes equations, Numerical Weather Forecast (NWP) models are widely used, but they also have certain drawbacks. Customers depend on weather forecast service providers for weather services. Forecasts are only accessible at particular intervals, and the offered time frames are always fixed. It is quite difficult to provide accurate forecasts using a physical model because the atmosphere is so chaotic. Therefore, alternative methods like statistical learning are used for short-term prediction. The Persistence or Naive forecasting is a statistical method that uses the value at a given instance to forecast for that same instance the following day. This model assumes the next value will be precisely the same as the last value seen. Therefore, Equation (1) can be used to express the Naive method.

$$Y_{t+1} = Y_t \quad (1)$$

where

Y_{t+1} is predicted value, Y_t is real value at time $t + 1$ and t respectively.

Manuscript received August 25, 2024; revised March 27, 2025.

Manisha Galphade is an Assistant Professor at the School of Computing, MIT Art Design and Technology University, Pune, India (phone: 9421259200 ; e-mail: galphademanisha@gmail.com).

Nilkamal More is an Associate Professor in the Department of Information Technology, K.J.Somaiya School of Engineering, Somaiya Vidyavihar University, Vidyavihar, Mumbai, India (e-mail: neelkamal-surve@somaiya.edu).

V. B. Nikam is an Associate Professor in the Department of Computer Engineering & IT, Veermata Jijabai Technological Institute, Mumbai, India (e-mail: vbnikam@gmail.com).

Biplab Banerjee is an Associate Professor in the Center of Studies in Resources Engineering, IIT Bombay Mumbai, INDIA (e-mail: getbiplab@gmail.com).

Arvind W. Kiwelekar is an Associate Professor in the Department of Information Technology, Dr Babasaheb Ambedkar Technological University, Lonere, Raigad, INDIA (e-mail: awk@dbatu.ac.in).

Priyanka Sharma is a Director of Software Engineering (HPC AI R&D Lab), Fujitsu Research (FRIPL), AI Thought Leader, Bengaluru, Karnataka, INDIA.(e-mail: drpriyankasharma.ai@gmail.com).

Conventional statistical techniques for forecasting wind speed or power utilize time series models, like moving average (MA), autoregressive (AR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA), which are easy to create and offer timely forecasts. Traditional statistical approaches are often used as reference models. Deep learning approaches provide the benefit of learning from data and generalizing, which sets them apart from typical learning methods. Different deep learning methods are employed based on the type of application, including convolutional neural network (CNN) [18], recursive neural networks (RNN), and long short term memory (LSTM) [19]. Based on our analysis of the literature survey [20], [21], [22], we have discovered that recent studies indicate that statistical methods are not suitable for non-linear wind data, struggle with handling large datasets, and are unable to predict long-term periods accurately. Therefore, statistical methods should not be considered as the primary recommended approach for prediction.

The hybrid models' objective is to leverage each model's strengths to get the most effective predicted performance. Hybrid techniques [14] combine different approaches, such as short-term and medium-term models, combining physical and statistical approaches, and so on. Hybrid approaches may be broadly categorized into four categories [23]: hybrid approaches based on weight, data post-processing, data pre-processing, and parameter selection and optimization techniques.

In our research, we propose a hybrid model based on data preprocessing. Conventional techniques, such as removing observations with missing values, replacing missing values with the mean, filling in missing values using previous or subsequent values, and using basic regression, are not able to handle large amounts of data, making it challenging to handle missing data in the big data era [24]. A data preprocessing model is provided below to address the existing gap in the literature. This approach utilizes Random Forest to identify and fill missing values in multivariate data. These approaches do not make any assumptions about the normal distribution or require the development of parametric models [25]. In addition, it has the benefit of effectively managing nonlinearities. Along with this, decomposition methods have emerged as powerful tools for improving forecasting accuracy. It works by breaking down complex time series data into simpler, more manageable components, each of which can be analyzed and predicted separately. Most wind power forecasting methods usually employ data decomposition techniques for processing, but often overlook the issue of high-frequency components. For instance, the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) decomposition approach produces a series with high frequency, which is volatile and contains noise [26]. The series with the high frequency is the most difficult to predict; because of this, the accuracy of the prediction might be significantly impacted [27]. Despite the significant advancements made by integrating the decomposition approach with artificial intelligence, additional improvements are necessary to enhance the accuracy of wind power forecasting further.

Over the last few years, numerous researchers have employed signal decomposition algorithms for ultra-ultra-short-term wind speed forecasting in wind farms, introducing

a category of forecasting methods centered on decomposition combinations [28]. In the literature, research on decomposition-based models is often classified into two classes: the direct approach [29] and the multi-component approach [30].

Wind power forecasting based on decomposition is an emerging approach that enhances predictions' accuracy by breaking down complex wind data into manageable components. Research indicates that this method effectively captures wind patterns' non-linear and time-varying nature. For instance, Hu et al. highlight the utilizations of empirical mode decomposition (EMD) to separate wind speed signals into intrinsic mode functions (IMFs), which improves forecasting precision by addressing noise and irregularities in data [31]. Further, integrating decomposition techniques with machine learning models demonstrates significant improvements in forecasting accuracy compared to traditional methods [32]. Additionally, wavelet decomposition allows for multi-resolution analysis, thus enhancing the model's ability to adapt to varying wind conditions [33]. However, limitations such as computational complexity and extensive historical data need to be addressed in implementing these techniques effectively [34]. A novel wind power forecasting method using improved variational mode decomposition (IVMD), mixture correntropy, and LSTM is proposed and optimized by particle swarm optimization [35]. Combining IVMD and sample entropy for data preprocessing, the hybrid IVMD-SE-PMC-LSTM model performs better with complex wind farm data.

Given the concerns mentioned above, this research suggests a hybrid CEEMDAN-SampEn model. The initial phase decomposes the original wind power series into multiple subseries using CEEMDAN. The next stage is to divide these subseries into two distinct groups, namely low and high frequency components. Due to the complexity and predictive challenges posed by the high frequency subseries, a Transformer model is employed, while the simpler low-frequency subseries are addressed using a BiLSTM model. Finally, the generated predictions by the Transformer and BiLSTM models are aggregated through summation. This work's key innovations and contributions to the existing state of the art presented as: 1) This work introduces an innovative wind energy prediction system that utilizes CEEMDAN decomposition to effectively separate the wind energy data into high and low frequency components to improve the prediction accuracy. 2) By combining Transformer and BiLSTM networks, the proposed method efficiently handles the complexity of high frequency data with Transformer and utilizes BiLSTM for simpler low-frequency data, providing a robust solution for time series prediction. 3) The integration of sample entropy as a feature selection technique enables a refined analysis of frequency components. It ensures that the model focuses on the most significant features, improving predictive performance. 4) The paper contains a thorough comparative analysis with existing methods, demonstrating the superior accuracy and reliability of the suggested approach in real wind energy forecasting scenarios.

The subsequent sections of the paper are organized as follows. The theoretical framework of the proposed work is explained in Section 2. In section 3, the proposed model is elaborated. Section 4 describes how the experimental analysis

is carried out. Section 5 concludes with a discussion of the research that will be done in the future.

II. THEORETICAL BACKGROUND

This section thoroughly examines the theoretical foundation of frameworks that support our approach.

1) **CEEMDAN**: The CEEMDAN is a signal processing technique used for decomposing non-stationary and nonlinear data into multiple IMFs [36]. It is an extension of the EMD and EEMD algorithms. The computational complexity of an algorithm serves as a critical performance metric. EMD and EEMD may produce unnecessary IMFs, adversely affecting their efficiency. Hence, minimizing the generation of such redundant IMFs becomes essential to enhance the computational efficiency of these methods. The author, Colominas, proposed an enhanced version of CEEMDAN [37]. These algorithms aim to reduce the number of IMFs generated while effectively separating distinct signal components. By leveraging these approaches, the computational burden can be reduced without compromising the accuracy of the decomposition process. The CEEMDAN method adds adaptive noise into the residual acquired from the previous step instead of directly introducing noise to the original signal, as done in EEMD. This adaptive noise is derived from the noise's mode corresponding to the iteration obtained with EMD. By incorporating adaptive noise in this manner, CEEMDAN aims to improve decomposition accuracy and mitigate mode mixing issues commonly encountered in EMD and EEMD. The key steps of the CEEMDAN algorithm involve iteratively decomposing the input signal into IMFs and a residual component, adding adaptive noise to the residual at each iteration, and repeating this process until convergence is achieved. The resulting IMFs can then be analyzed or used for further processing tasks such as signal denoising, feature extraction, or time series forecasting. Figure 1 shows that the original wind power sequence is separated into ten IMFs and one residual component.

- 1) **Ensemble Generation**: Generate ensemble members by adding original signal with white noise as in Equation (2):

$$x_i(t) = x(t) + \epsilon_i(t) \quad (2)$$

where $x_i(t)$ is the i^{th} ensemble member, $x(t)$ is the original signal, $\epsilon_i(t)$ is white noise.

- 2) **Apply EMD**: Apply the EMD process independently to each ensemble member to obtain IMFs and a residual component. For each ensemble member $x_i(t)$, decompose it into IMFs $c_{i,k}(t)$ and a residual $r_i(t)$ as in Equation (3).

$$x_i(t) = \sum_{k=1}^{N_i} c_{i,k}(t) + r_i(t) \quad (3)$$

- 3) **Adaptive Noise Injection**: Add adaptive noise to the residual obtained from the previous iteration:

$$r_i^n(t) = r_i^{n-1}(t) + \epsilon_i(t) \quad (4)$$

where $r_i^n(t)$ is the residual at iteration n , and $\epsilon_i(t)$ is the noise added to the residual.

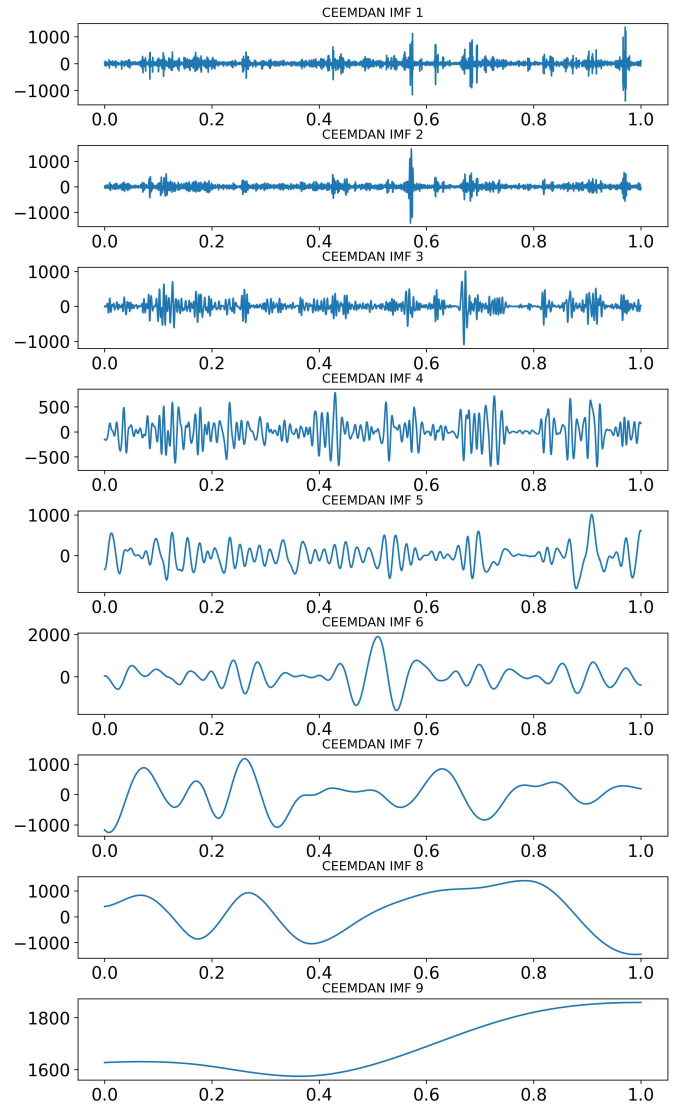


Fig. 1. IMF generated by CEEMDAN

- 4) **IMF Averaging**: Average the IMFs obtained from all ensemble members to reduce noise and improve stability:

$$C_k^n = \frac{1}{M} \sum_{i=1}^M C_{i,k}^n(t) \quad (5)$$

where C_k^n is the averaged IMF at iteration n , M is the total number of ensemble members and $C_{i,k}^n(t)$ is the k^{th} IMF of the i^{th} ensemble member at iteration n .

- 5) **Residual Calculation**: Compute the residual component by averaging the residuals from all ensemble members:

$$r^n(t) = \frac{1}{M} \sum_{i=1}^M r_i^n(t) \quad (6)$$

- 6) **Iteration**: Repeat steps 2-5 iteratively until a convergence criterion is met.

2) **Sample Entropy**: This is a measure which helps to assess the complexity and pattern of time series data [38]. It calculates the likelihood of the patterns in the data that are similar and will remain similar in the future. This metric provides valuable information about the pattern of the time

series, helping to understand its behavior over time.

Step 1: Given a time series of length N , $\{x_i, x_{i+1}, x_{i+2} \dots x_N\}$ vectors of length m (the embedding dimension) are created as $X_i = \{x_i, x_{i+1}, x_{i+2} \dots x_N\}$ for $i = 1, 2, 3, \dots, N - m + 1$.

Step 2: The distance between two vectors X_i and X_j as the maximum absolute difference between their corresponding components as given in Equation (7):

$$d(X_i, X_j) = \max_{k=0, \dots, m-1} |x_{i+k} - x_{j+k}| \quad (7)$$

Step 3: Define a tolerance level r . Two vectors X_i and X_j are considered similar if $d(X_i, X_j) \leq r$

Step 4: Count the number of vectors X_j that are similar to X_i (excluding self-matches) and normalize by the total number of comparisons using Equation (8):

$$C_i^m(r) = \frac{\text{Number of } X_j \text{ such that } d(X_i, X_j) \leq r}{N - m} \quad (8)$$

Step 5: Compute the average of $C_i^m(r)$ for all i using Equation (9).

$$C^m(r) = \frac{1}{N - m + 1} \sum_{i=1}^{N-m+1} C_i^m(r) \quad (9)$$

Step 6: Repeat the process for $m + 1$ to form X_i' vectors of length $m + 1$ and compute $C^{m+1}(r)$ using Equation (10).

$$C^{m+1}(r) = \frac{1}{N - m} \sum_{i=1}^{N-m} C_i^{m+1}(r) \quad (10)$$

Step 7: Finally, sample entropy is calculated using Equation (11)

$$\text{SampEn}(m, r, N) = -\ln \left(\frac{C^{m+1}(r)}{C^m(r)} \right) \quad (11)$$

Sample entropy provides a robust way to quantify the complexity of time-series data by measuring the likelihood that identical patterns of observations remain similar as the series progresses. This makes it a valuable tool in fields like physiology, financial analysis, and other domains where understanding the underlying dynamics of a time series is crucial.

3) *Transformer*: The Transformer architecture [39], initially implemented for natural language processing, is also gaining attention for its application in time series analysis. The original Transformer architecture has six stacked encoder and decoder layers; the number can vary depending on the application. Each layer comprises a multi-head self-attention mechanism and a feedforward network as its sub-layers. The architecture consists of three sub-layers in each layer:

- 1) A masked multi-head self-attention mechanism
- 2) A multi-head attention mechanism operated on the encoder's output
- 3) a feed-forward network

The masked self-attention guarantees that the forecast for a certain position is only determined by the known outputs

generated at the previous position. The Transformer architecture struggles to capture the sequence of input data. In order to solve this issue, positional encoding is used to find the position of each element in the input sequence. Every individual input sequence value is associated with a feature vector. At the start, it performs embedding operations on each input, expanding the data from a one-dimensional matrix to a two-dimensional matrix, as illustrated in Figure 3. Following this, sine and cosine function is used to encode the input sequence using Equation (12) and Equation (13), resulting in a fixed absolute position representation, known as position encoding. Ultimately, the position encoding is combined with the embedding sequence that was generated before.

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (12)$$

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (13)$$

where,

$PE_{pos, i}$ is the i^{th} dimension of the positional encoding vector for position pos ,

d_{model} is the dimensionality of the input embeddings.

The choice of sinusoidal functions ensures that the positional encoding vectors have a unique representation for each position and have a smooth, continuous nature. The frequencies and phases of the sinusoidal functions allow the model to learn meaningful positional representations, with lower frequencies capturing global positional information and higher frequencies capturing local positional information.

The self-attention mechanism enables the model to assign a weight to each element in the input sequence when making predictions for a specific element, as shown in Figure 2. It calculates the attention scores by considering pairwise association between all samples in the input sequence. Let's consider input sequence as $X = \{x_1, x_2, \dots, x_n\}$ where x_i represents the time series value at time step i . There are three learnable vectors associated with each element x_i as: Q_i in the form of a query vector, K_i as a key vector, and V_i a value vector as shown in Equation (14) to (16).

$$Q_i = W_Q \cdot x_i \quad (14)$$

$$K_i = W_K \cdot x_i \quad (15)$$

$$V_i = W_V \cdot x_i \quad (16)$$

where,

W is weight matrix,

Subscript K , Q , and V are learnable vectors.

The attention score α_{ij} between each pair of elements is computed by performing dot product of their query and key vectors using Equation (17).

$$\alpha_{ij} = \text{softmax} \left(\frac{Q_i \cdot K_j}{\sqrt{d_k}} \right) \quad (17)$$

where d_k is the key vectors dimensionality.

The attention scores are used to obtain the context vector C_i for each element x_i as shown in Equation (18).

$$C_i = \sum_{j=1}^n \alpha_{ij} \cdot V_j \quad (18)$$

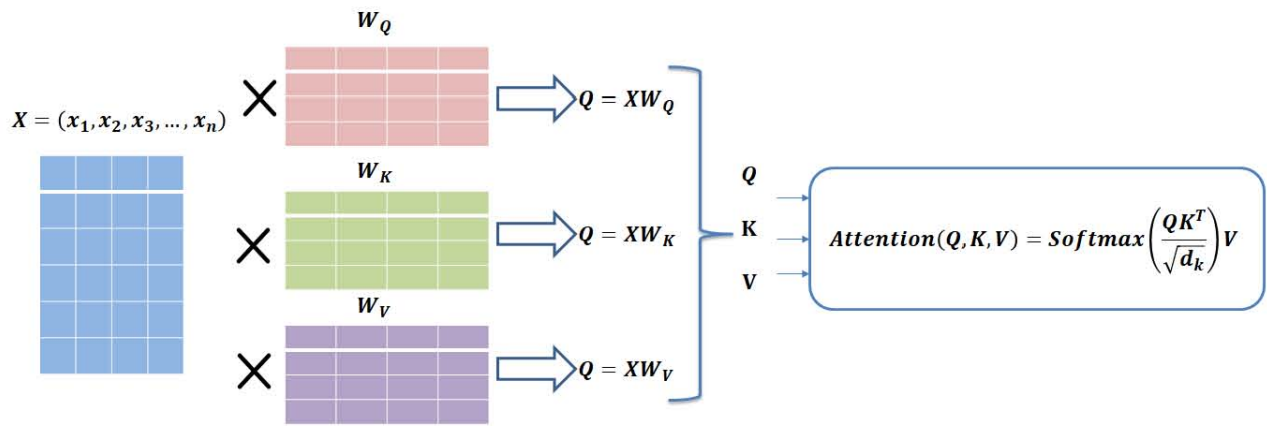


Fig. 2. Self-attention evaluate relationships within a sequence, helping the model capture dependencies between elements.

0	P_{00}	P_{01}	...	P_{0d}
1	P_{10}	P_{11}	...	P_{1d}
2	P_{20}	P_{21}	...	P_{2d}
3	P_{30}	P_{31}	...	P_{3d}
4	P_{40}	P_{41}	...	P_{4d}

Fig. 3. Position embeddings encode the order of sequence elements, enabling the model to recognize positional information.

The context vector C_i represents the aggregated information from the entire input sequence. It is calculated as a weighted sum of the value vectors of all elements in the sequence. The attention score is used to determine the weights of these elements. This allows the model to focus on the important aspects of the input sequence while making predictions for each element. In the subsequent stage of multi-head attention, the self-attention mechanism is implemented many times simultaneously, each with its own distinct set of trainable parameters, as seen in Figure 4. This feature allows the model to pay attention to many segments of the input sequence concurrently, therefore capturing a wide range of patterns and correlations. Mathematically, multi-head attention can be defined using Equation (19). A final output is obtained by concatenating and linearly transforming the context vectors from all heads.

$$MultiHead(X) = Concat(C_i^1, C_i^2, \dots, C_i^H).W_o \quad (19)$$

where,

H is the total number of heads,

$Concat$ denotes concatenation,

W_o is a learnable weight matrix for the output.

Multi-head attention helps concurrently record several elements of the input sequence, improving the ability to learn correlations and complicated patterns effectively.

Feed-forward neural networks (FFNNs) play a crucial role in the subsequent processing of the outputs generated by the

self-attention mechanism. Every layer in both the encoder and decoder of a Transformer contains FFNN that enhances the representations obtained from the attention layers. Once the self-attention mechanism captures the relationship among various tokens in a sequence, the resultant vectors are fed into the FFNN. This network has two linear transformations divided by a non-linear activation function, usually ReLU. The FFNN applies these transformations to each token independently, adding non-linearity and enabling the model to understand complex patterns. Beginning with the self-attention layer's output, which is combined with the original input by a residual link, the process advances to layer normalization.

This normalized output is fed into the FFNN, where the first linear transformation is applied, followed by the ReLU activation. The resulting vectors then undergo a second linear transformation. Finally, the output of the FFNN is merged with the input through an additional residual link and normalized, preparing it for the next layer. This process ensures the model maintains stability and accelerates training by preventing issues like vanishing or exploding gradients. The FFNNs in Transformers significantly improve the model's capacity to collect and depict complex relationships within the data, enhancing its efficacy in various applications like language modeling and translation.

A. BiLSTM

BiLSTM is an RNN that improves the capacity to discover relationships in sequential data by analyzing it in both the forward and backward directions [40], [41] as shown in Figure 5.

The outputs of these two LSTM layers are combined, allowing the network to learn from both the future and past context of each time step, improving its ability to understand and predict sequential data. The output of each LSTM cell can then be merged or combined to provide the overall output of the biLSTM as in Equation (20).

$$y_t = f(\vec{h}_t, \overleftarrow{h}_t) \quad (20)$$

where,

$\vec{h}_t$ output of forward layer,

$\overleftarrow{h}_t$ output of backward layer

f is a function such as sum, mul, concat or avg.

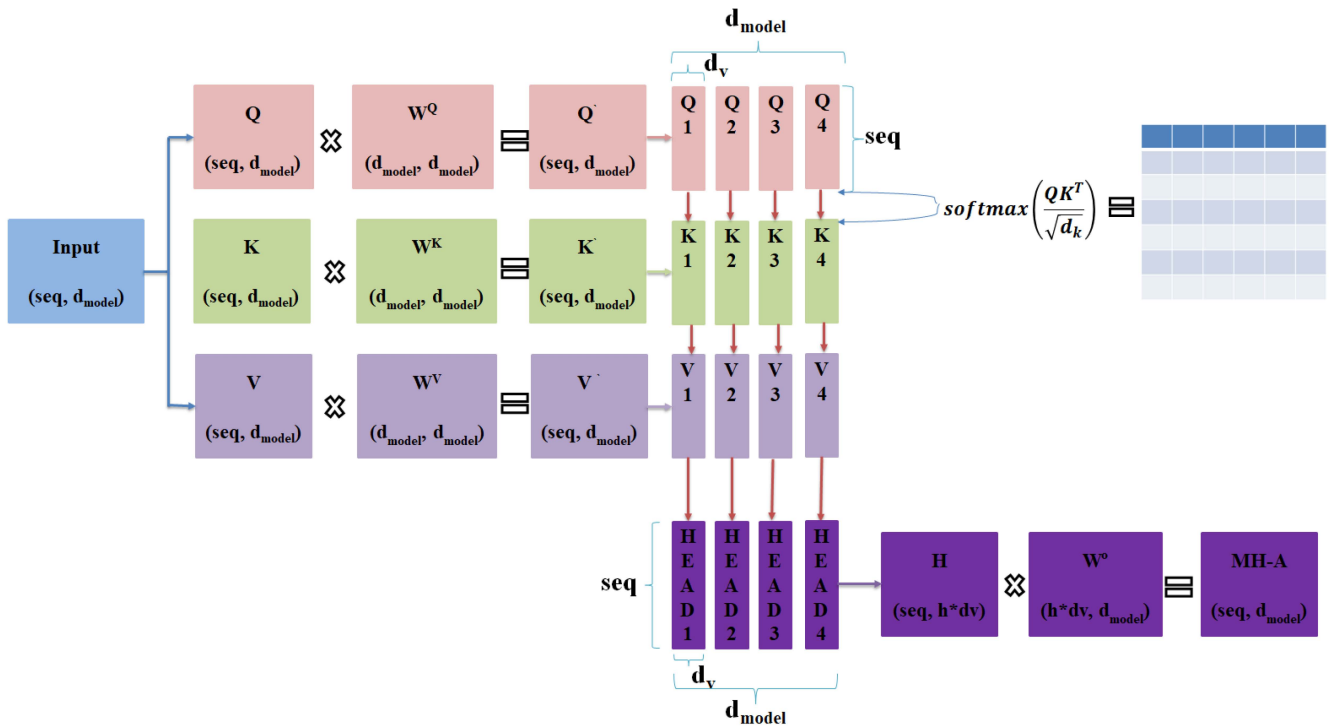


Fig. 4. Multi-head attention enables parallel focus on various sequence parts, enhancing contextual understanding.

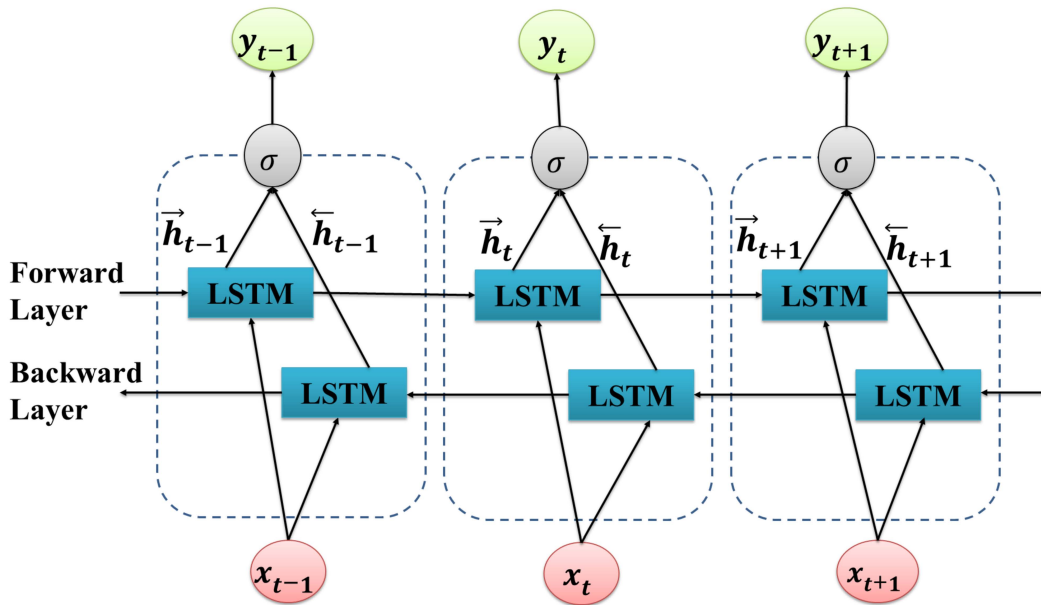


Fig. 5. Architecture of a BiLSTM network

III. DESIGN OF THE PROPOSED MODEL

A Hybrid wind power forecast model, now referred to as CEEMDAN-SampEn-TRAN-BiLSTM, which integrates CEEMDAN, sample entropy, Transformer, and BiLSTM, is proposed in this work. In this hybrid model, two prediction models were selected based on the properties of the sequences. The Transformer model, which has an encoder-decoder architecture, handles high-frequency sequences with complex structures. This model can offer higher attention to detail, effectively mining patterns and yielding more accurate prediction results. Conversely, for low-frequency sequences with stronger periodicity and lower complexity, the BiLSTM model is chosen. Its relatively simple structure

makes it well-suited for series prediction, as it reduces training time and accelerates prediction speed—an overview of the proposed model's essential structure. The general framework of the suggested model is depicted in Figure 6. The procedure is partitioned into six steps as outlined below:

Step 1: The original wind power series is separated IMFs and a residual component using CEEMDAN. These IMFs represent various frequency components of the original signal.

Step 2: The decomposed IMFs are analyzed for frequency characteristics using Sample entropy. The IMFs are

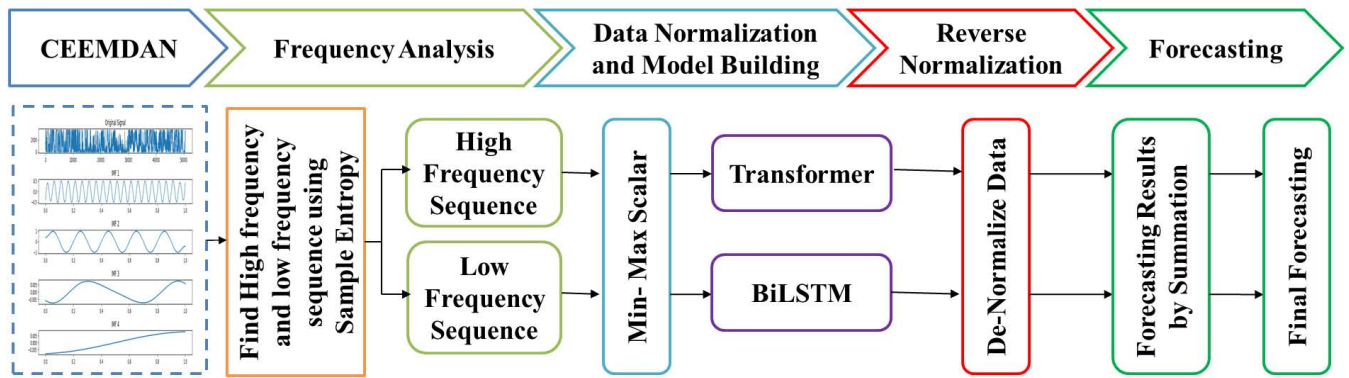


Fig. 6. Design of the Proposed Model

categorized into high and low frequency subseries, which will help to distinguish between different patterns and complexities in the data.

Step 3: The high and low frequency subseries are separately normalized using a Min-Max Scalar. This normalization scales the data to a specific range, which helps improve the subsequent models' performance.

Step 4: The normalized high-frequency sequences are fed into a Transformer model, while the low-frequency sequences are input into a BiLSTM model. These models are chosen for their ability to record several data points. Transformers effectively capture long-range dependencies, and BiLSTMs are efficient for sequence prediction tasks.

Step 5: The outputs are de-normalized after the prediction models (Transformer for high frequency and BiLSTM for low frequency) generate their respective forecasts. This step reverses the normalization process, returning the scaled predictions to their original range.

Step 6: The de-normalized predictions from both models are combined. The results are summed to generate a comprehensive forecast of wind power. The overall forecast is derived by combining the high and low frequency predictions, providing a complete wind power prediction.

A. Hyperparameter Tuning

Figure 7 illustrates the RMSE values corresponding to various lookback window sizes, ranging from 1 to 10. The RMSE exhibits a U-shaped pattern, with values decreasing initially and then increasing as the lookback size grows. The minimum RMSE is observed at a lookback value of 5, suggesting that this is the optimal window size for this dataset. Smaller lookback sizes, such as 1 or 2, may not capture sufficient historical information, resulting in higher errors. Conversely, larger lookback sizes beyond 5 lead to increased RMSE, potentially due to overfitting or the inclusion of redundant data. Therefore, a lookback value of 5 is recommended as it balances the trade-off between underfitting and overfitting, providing the most accurate predictions for this dataset. As presented in Figure 7, a look-back value of 5 demonstrates the best performance with the lowest RMSE,

indicating its suitability for capturing the temporal dependencies in the dataset. Therefore, this value has been adopted uniformly across all hybrid models that incorporate LSTM for prediction tasks. This consistent selection ensures that the models leverage the optimal look-back period for effective feature extraction and accurate forecasting. By maintaining this configuration, the comparative analysis across different hybrid architectures remains valid, and the predictive performance is enhanced due to the proven efficacy of this look-back value in capturing sequential patterns. The remaining

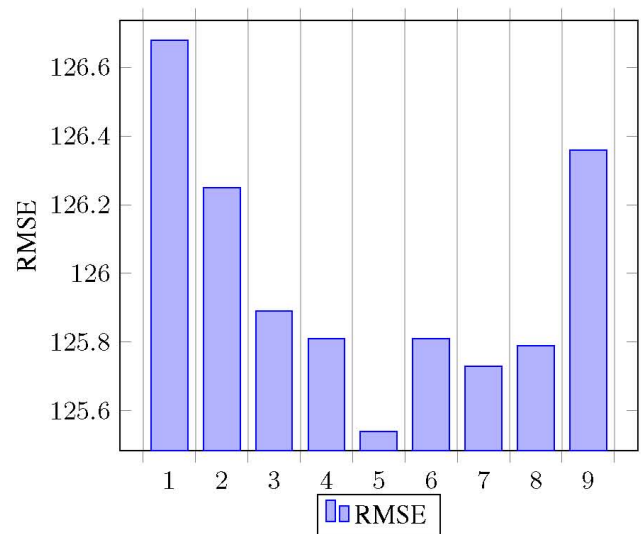


Fig. 7. Impact of Lookback on LSTM performance

hyperparameters for the proposed model are detailed in Table I, providing a comprehensive overview of the configurations used. The dataset employed in this study was split into training and testing sets following a 70-30 split ratio, where 70% of the data was used for training the model to learn underlying patterns, and the remaining 30% was reserved for testing to evaluate its predictive performance. This split ensures a balanced approach, providing sufficient data for both training and validation while minimizing the risk of overfitting.

IV. EXPERIMENTAL RESULT

This experiment uses the Sotavento wind farm dataset, located in Galicia, Spain. The wind farm has 24 onshore

TABLE I
HYPERPARAMETERS CONFIGURATION OF PROPOSED MODELS.

Model	Parameter	Parameter description	Value
CEEMDAN	Noise level	Value of noise that is added adaptively to the signal during the decomposition process.	0.05
BiLSTM	Hidden layer sizes	Refer to the number of neurons or nodes in each hidden layer of the neural network.	128
	Activation function	To decides whether a neuron should activate or not, helping the model learn complex patterns in data.	ReLU
	Optimizer	To adjusts a model's weights to make predictions more accurate by reducing the error step by step.	Adam
	Learning rate	The step size at which a model's weights are updated during optimization	0.001
	Epochs	The number of times a machine learning model goes through the entire training dataset to learn from it	100
	batch_size	The number of training examples processed together in a single forward and backward pass of a neural network.	64
Transformer	num_layers	Number of encoder and decoder layers	6
	num_heads	Number of attention heads in multi-head attention	12
	d_ff	Dimensionality of the feed-forward network	256
	dropout_rate	Probability of dropping neurons during training.	0.1
	max_seq_length	Maximum sequence length supported.	28
	batch_size	Number of samples processed in one iteration	64

TABLE II
THE CHARACTERISTICS OF THE SOTAVENTO WIND FARM DATASET

Data Source	Attribute Specifications	Resolution	Capacity	No. of records
Sotavento wind farm	Wind speed, wind direction and wind power	10-minute, 1-hour, 1-day, Data from 2014 to till date	17560kW	52568 records with 3 attributes

turbines and a total capacity of 17560 kW [42]. The characteristics of the dataset include location, capacity, attributes, record count and resolution, detailed in Table II.

The proposed model takes the wind power sequence as an input and applies CEEMDAN with a noise level of 0.05 to split it. The complexity of decomposed subseries is determined using sample entropy as listed in Table III. Greater sample entropy value signifies increased nonlinearity, complexity, and a more impact on the accuracy of prediction. The entropy curve is depicted in Figure 8. As the frequencies of the IMF and residual components fall, the entropy values also decrease, suggesting a reduction in the complexity of each component. IMF1 is having highest sample entropy among the other components, indicating that the first IMF exhibits the highest level of system complexity. We are

combining IMF1 to IMF5 into IMF1_part1 and using it as input for the first model, which is a Transformer. The remaining IMFs are combined into IMF_Part2 and used as input for the second model, which is a BiLSTM.

TABLE III
SAMPLE ENTROPY VALUE FOR EACH SUB-SERIES

IMF Number	Sample Entropy Value
IMF1	0.271
IMF2	0.191
IMF3	0.196
IMF4	0.165
IMF5	0.148
IMF6	0.091
IMF7	0.042
IMF8	0.021
IMF9	0.006
Residual	0.001

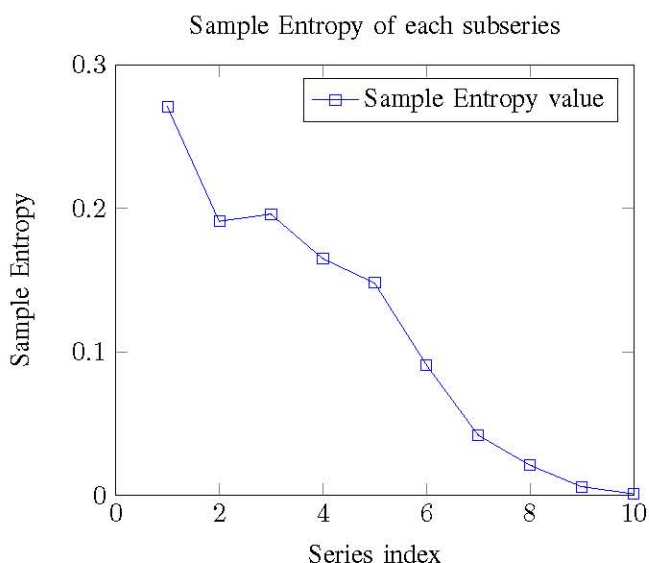


Fig. 8. Sample Entropy for each series

Figure 9 shows a comparative analysis of different decomposition methods applied to a signal. Each method decomposes the original signal into several IMFs, and for each IMF, two plots are displayed: the time-domain signal (left column) and the frequency spectrum (right column) through a Fourier Transform. In general, EMD, EEMD, and CEEMDAN show similar trends in terms of IMF structure. Lower-order IMFs contain higher-frequency oscillations, while higher-order IMFs capture lower-frequency, trend-like components. However, the decomposition quality varies significantly across methods, particularly in terms of frequency separation, noise handling, and IMF stability. EMD, the simplest of the three methods, produces a series of IMFs that are more prone to noise and mode mixing. Mode mixing occurs when a single IMF contains oscillations of multiple frequencies, which makes it challenging to interpret the results. The frequency spectra of EMD's IMFs show some overlapping frequencies, especially in lower-order IMFs, indicating that EMD struggles to isolate distinct frequency

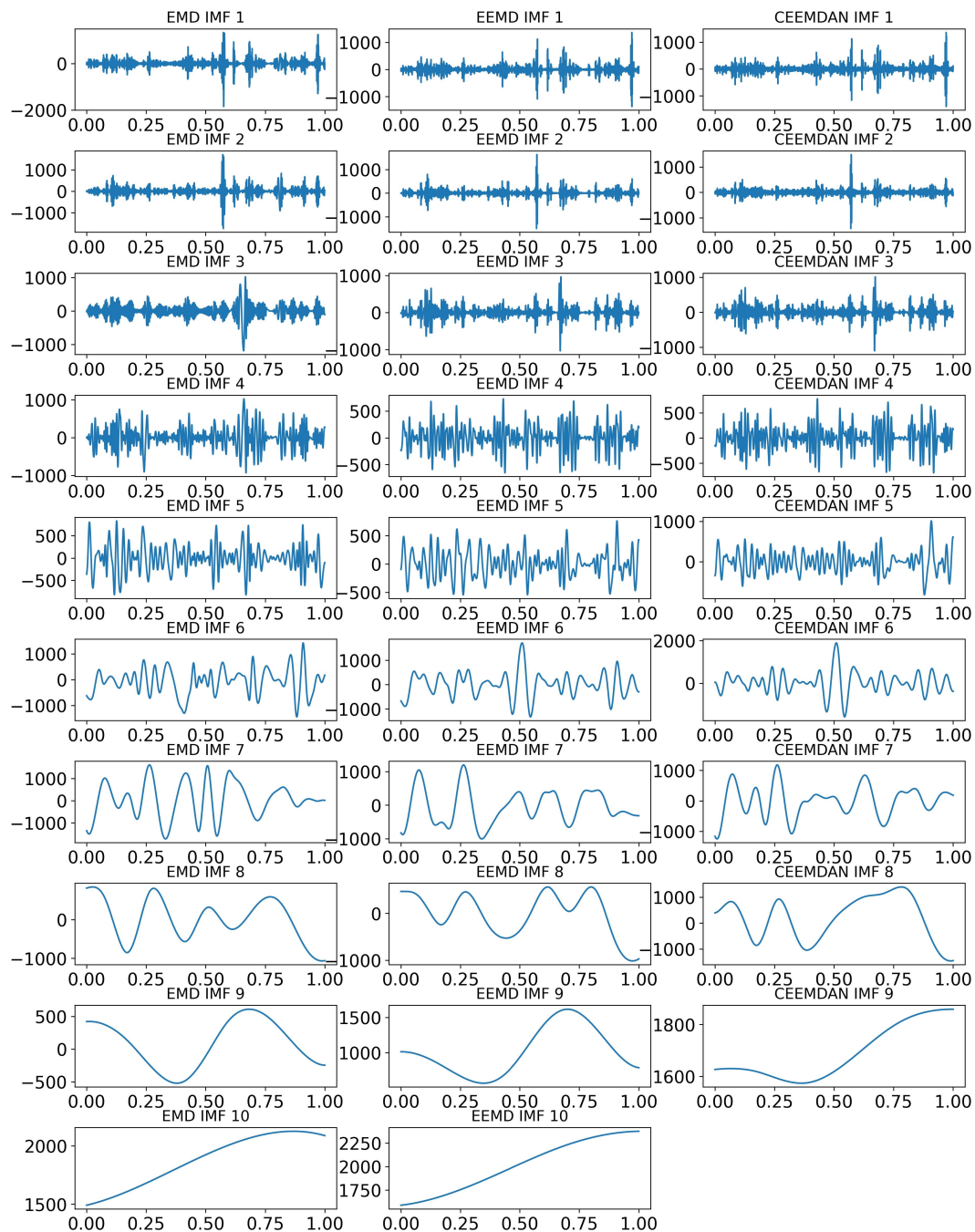


Fig. 9. Comparative analysis of different decomposition methods

components in noisy or complex signals. EEMD addresses some limitations of EMD by adding white noise to the signal and averaging the decompositions over multiple trials. This approach reduces mode mixing, resulting in IMFs with more stable frequency characteristics and less noise. The frequency spectra of EEMD's IMFs show clearer separation of frequencies, indicating improved frequency isolation. EEMD thus provides a more reliable decomposition, especially for signals that contain overlapping frequency components. CEEMDAN, an advanced extension of EEMD, offers even greater robustness by further mitigating mode mixing and enhancing frequency separation. The IMFs obtained through CEEMDAN are smooth, with minimal noise interference, and their frequency spectra exhibit clearly defined peaks with distinct separation between modes. This suggests that

CEEMDAN is particularly effective at isolating different frequency bands, making it the most accurate method for decomposing complex signals among the three. In summary, EMD is the least robust of the three methods, often resulting in noisy and mixed IMFs. EEMD improves upon EMD by stabilizing the decomposition and enhancing frequency isolation, while CEEMDAN provides the most reliable and clean decomposition, excelling at separating frequency components with minimal noise. For applications where accurate frequency isolation is crucial CEEMDAN would likely be the preferred method due to its superior ability to handle noise and avoid mode mixing.

The Figure 10 shows intermediate results from the decomposition of a time series using CEEMDAN. In the top rows, the IMFs represent high-frequency components with rapid

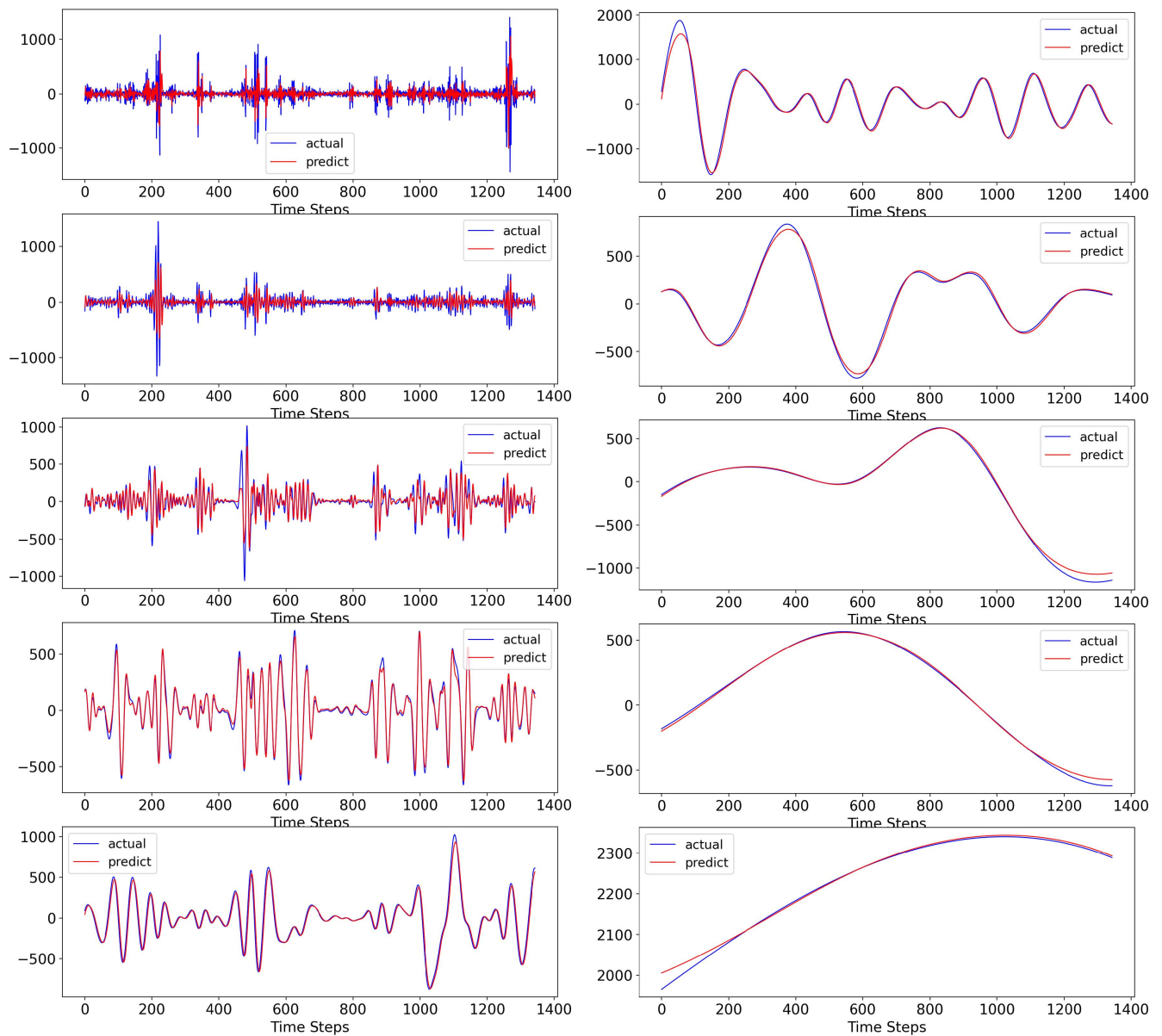


Fig. 10. Actual vs predicted result of IMFs

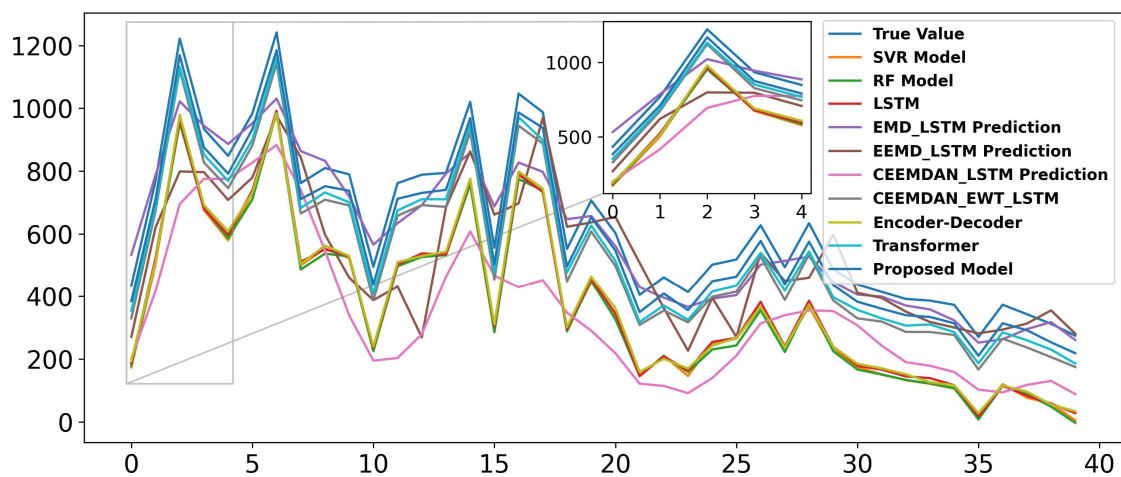


Fig. 11. Comparison of various model's predictions



Fig. 12. Radar chart for ten models

oscillations, capturing short-term fluctuations in the data. Here, the model closely follows the actual values but shows minor discrepancies, particularly at peaks and troughs, which is expected as high-frequency components are often noisy and challenging to predict accurately. Moving down the rows, the IMFs capture intermediate frequencies with smoother oscillations and moderate trends. In these middle rows, the model performs better, tracking the actual values with greater precision, which suggests that it is adept at handling medium-frequency components that represent key patterns in the data. Overall, this decomposition reveals that the model is well-suited to capturing both short-term and long-term dynamics within the data. It performs effectively across high, medium, and low-frequency IMFs, with the highest accuracy observed in the low-frequency components, which are crucial for understanding the general trend. Such a multi-frequency analysis is valuable in fields like machinery fault detection, wind turbine monitoring, and financial time series forecasting, where it's essential to differentiate between rapid fluctuations and underlying trends. This detailed breakdown

across IMFs provides insights into the model's strengths and potential areas for refinement, especially in handling noisy, high-frequency components.

Table IV presents a comparative analysis of several predictive model, focusing on their RMSE and MAE. The SVR model shows RMSE of 255.8 and MAE of 116.52, indicating lower prediction accuracy and precision. The RF model has a little superior performance, as indicated by RMSE of 253.64 and an MAE of 107.01, in comparison to the SVR model. LSTM model demonstrates a slight enhancement in accuracy and precision compared to SVR and RF, evidenced by its RMSE of 282.87 and MAE of 97.16. The combination of EMD with LSTM (EMD_LSTM) significantly reduces both RMSE and MAE to 170.53 and 84.71, respectively, highlighting the advantage of signal decomposition before prediction. Further improvement is seen with the EEMD combined with LSTM (EEMD_LSTM), which achieves an RMSE of 127.52 and an MAE of 84.24. The CEEMDAN_LSTM model, which incorporates CEEMDAN, performs even better with RMSE of 123.02 and an MAE of 83.99. Adding EWT

TABLE IV
COMPARISON OF PROPOSED MODEL WITH STATE-OF-ART APPROACHES

Sr.No.	Model	Train accuracy			Test accuracy		
		RMSE	MAE	R2	RMSE	MAE	R2
1	SVRModel	255.8	116.52	0.84	253.34	115.54	0.84
2	RFModel	253.64	107.01	0.83	252.23	106.89	0.83
3	LSTM	282.87	97.16	0.84	280.97	96.94	0.84
4	EMD_LSTM	170.53	84.71	0.85	169.09	83.34	0.92
5	EEMD_LSTM	127.52	84.24	0.96	177.55	86.47	0.95
6	CEEMDAN_LSTM	123.02	83.99	0.97	150.5	79.09	0.97
7	CEEMDANEWT_LSTM	108.26	53.03	0.97	105.32	52.02	0.97
8	Encoder-Decoder	254.44	162.02	0.97	255.89	162.02	0.97
9	Transformer	78.32	43.45	79.89	80.03	45.43	0.99
10	ProposedModel	57.45	18.01	0.99	55.57	19.03	0.99

to this model (CEEMDAN_EWT_LSTM) enhances performance further, reducing RMSE to 108.26 and MAE to 53.03. The Encoder-Decoder model, despite being advanced, shows high error values with an RMSE of 254.44 and an MAE of 162.02, indicating it may not be as effective for this particular prediction task. The Transformer model demonstrates greatly enhanced performance, achieving a much reduced RMSE of 78.32 and a MAE of 43.45. The proposed hybrid model, likely incorporating multiple advanced techniques, delivers the best performance with the lowest RMSE of 57.45 and MAE of 18.01, showcasing superior prediction accuracy and precision compared to all other models. Figure 11 presents a comparison of various models' predictions for wind power generation against the true values over a specified period. The x and y axis represent time steps and generated wind power, respectively. The main graph displays the actual versus predicted values for each model over time. Models that have prediction lines closely following the black line are considered more accurate. Notably, the proposed model (pink) demonstrates a strong performance, closely tracking the actual values. Other models like the Transformer (orange), LSTM (blue), and CEEMDAN_EWT_LSTM (magenta) also show commendable accuracy but with slight deviations in certain intervals. Additionally, an inset graph zooms in on a specific time range (from 0 to 5 on the x-axis), providing a detailed view of the models' predictions during that period. This detailed comparison aids in examining the models' performances more closely for a particular timeframe. In summary, the diagram clearly illustrates the accuracy of each predictive model, emphasizing the efficacy of the suggested model in reliably predicting wind power generation.

Figure 12 displays three radar charts, each of which compares the performance of different time series forecasting models based on key metrics: RMSE, MAE and R2. The top-left radar chart focuses on RMSE, which quantifies the average magnitude of prediction errors, with lower values indicating better performance. The top-right radar chart shows the MAE values for each model, another measure of prediction accuracy. The bottom radar chart presents R2 values, which indicate how well each model explains the variability in the data, with values closer to one representing better performance. Overall, these radar charts suggest that the "Proposed Model" and CEEMDAN-based models outperform traditional models (such as LSTM and Transformer) across RMSE, MAE, and R2 metrics, highlighting their effectiveness in time series forecasting. The charts visually reinforce the comparative advantages of these models in both

error minimization and explanatory power.

V. CONCLUSION

This work presents the advanced wind energy forecasting model that integrates CEEMDAN decomposition with a hybrid Transformer-BiLSTM neural network, enhanced by the use of sample entropy for frequency-based analysis. The approach effectively addresses the challenges posed by the varying complexities of high and low frequency components within wind power time series data. By leveraging the Transformer model's ability to capture intricate patterns in high-frequency subseries and the BiLSTM's strength in handling low-frequency data, the proposed method offers a substantial improvement in forecasting accuracy. The results indicate that this hybrid model not only outperforms traditional forecasting techniques but also provides a more reliable and robust framework for predicting wind energy output. This work contributes to the ongoing development of more accurate and efficient forecasting tools, essential for the incorporation of renewable energy sources into power systems. Future research could explore further optimization of model parameters and extend this approach to other forms of renewable energy forecasting.

REFERENCES

- [1] S. Singh, "Energy crisis and climate change: Global concerns and their solutions," *Energy: crises, challenges and solutions*, pp. 1–17, 2021.
- [2] E. Hille, "Europe's energy crisis: Are geopolitical risks in source countries of fossil fuels accelerating the transition to renewable energy?" *Energy Economics*, vol. 127, p. 107061, 2023.
- [3] Y. Xu and F. Zhao, "Impact of energy depletion, human development, and income distribution on natural resource sustainability," *Resources Policy*, vol. 83, p. 103531, 2023.
- [4] M. A'zam, O. Shoxruxmirzo, and O. Serobiddin, "Wind generator: A clean and sustainable source of energy," 2023.
- [5] M. Hannan, A. Q. Al-Shetwi, M. Mollik, P. J. Ker, M. Mannan, M. Mansor, H. M. Al-Masri, and T. I. Mahlia, "Wind energy conversions, controls, and applications: a review for sustainable technologies and directions," *Sustainability*, vol. 15, no. 5, p. 3986, 2023.
- [6] S. Kumar and K. Rathore, "Renewable energy for sustainable development goal of clean and affordable energy," *International Journal of Materials Manufacturing and Sustainable Technologies*, vol. 2, no. 1, pp. 1–15, 2023.
- [7] O. Summerfield-Ryan and S. Park, "The power of wind: The global wind energy industry's successes and failures," *Ecological Economics*, vol. 210, p. 107841, 2023.
- [8] R. Pérez-Chacón, G. Asencio-Cortés, F. Martínez-Álvarez, and A. Troncoso, "Big data time series forecasting based on pattern sequence similarity and its application to the electricity demand," *Information Sciences*, vol. 540, pp. 160–174, 2020.
- [9] J. K. Kirkegaard, D. P. Rudolph, S. Nyborg, H. Solman, E. Gill, T. Cronin, and M. Hallisey, "Tackling grand challenges in wind energy through a socio-technical perspective," *Nature Energy*, vol. 8, no. 7, pp. 655–664, 2023.

- [10] R. Yasmeen, X. Zhang, A. Sharif, W. U. H. Shah, and M. S. Dincă, "The role of wind energy towards sustainable development in top-16 wind energy consumer countries: Evidence from stirpat model," *Gondwana Research*, vol. 121, pp. 56–71, 2023.
- [11] M. of Statistics and P. Implementation, "Energy statistics india," Year Published, 2024. [Online]. Available: https://www.mospi.gov.in/sites/default/files/publication_reports/EnergyStatistics_India_publication_2024N.pdf
- [12] X. Zhao, S. Wang, and T. Li, "Review of evaluation criteria and main methods of wind power forecasting," *Energy Procedia*, vol. 12, pp. 761–769, 2011.
- [13] J. Jung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 762–777, 2014.
- [14] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," in *North American power symposium 2010*. IEEE, 2010, pp. 1–8.
- [15] A. Lahouar and J. B. H. Slama, "Hour-ahead wind power forecast based on random forests," *Renewable energy*, vol. 109, pp. 529–541, 2017.
- [16] N. S. Pearre and L. G. Swan, "Statistical approach for improved wind speed forecasting for wind power production," *Sustainable Energy Technologies and Assessments*, vol. 27, pp. 180–191, 2018.
- [17] Q. He, J. Wang, and H. Lu, "A hybrid system for short-term wind speed forecasting," *Applied energy*, vol. 226, pp. 756–771, 2018.
- [18] M. Solas, N. Cepeda, and J. L. Viegas, "Convolutional neural network for short-term wind power forecasting," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. IEEE, 2019, pp. 1–5.
- [19] T. Srivastava, Vedanshu, and M. Tripathi, "Predictive analysis of rnn, gbm and lstm network for short-term wind power forecasting," *Journal of Statistics and Management Systems*, vol. 23, no. 1, pp. 33–47, 2020.
- [20] R. Yu, J. Gao, M. Yu, W. Lu, T. Xu, M. Zhao, J. Zhang, R. Zhang, and Z. Zhang, "Lstm-efg for wind power forecasting based on sequential correlation features," *Future Generation Computer Systems*, vol. 93, pp. 33–42, 2019.
- [21] T. Peng, J. Zhou, C. Zhang, and Y. Zheng, "Multi-step ahead wind speed forecasting using a hybrid model based on two-stage decomposition technique and adaboost-extreme learning machine," *Energy Conversion and Management*, vol. 153, pp. 589–602, 2017.
- [22] K. Wang, X. Qi, H. Liu, and J. Song, "Deep belief network based k-means cluster approach for short-term wind power forecasting," *Energy*, vol. 165, pp. 840–852, 2018.
- [23] A. Tascikaraoglu and M. Uzunoglu, "A review of combined approaches for prediction of short-term wind speed and power," *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 243–254, 2014.
- [24] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, pp. 1–16, 2016.
- [25] M. Galphade, V. B. Nikam, B. Banerjee, A. W. Kiwelekar, and P. Sharma, "Stacked lstm for short-term wind power forecasting using multivariate time series data," *International Journal of Interactive Multimedia and Artificial Intelligence*, 2024. [Online]. Available: <http://dx.doi.org/10.9781/ijimai.2024.07.002>
- [26] P. Lv, Y. Shu, J. Xu, and Q. Wu, "Modal decomposition-based hybrid model for stock index prediction," *Expert Systems with Applications*, vol. 202, p. 117252, 2022.
- [27] Y. Lin, Z. Lin, Y. Liao, Y. Li, J. Xu, and Y. Yan, "Forecasting the realized volatility of stock price index: A hybrid model integrating ceemdan and lstm," *Expert Systems with Applications*, vol. 206, p. 117736, 2022.
- [28] S.-X. Lv and L. Wang, "Deep learning combined wind speed forecasting with hybrid time series decomposition and multi-objective parameter optimization," *Applied Energy*, vol. 311, p. 118674, 2022.
- [29] H. T. Nguyen and I. T. Nabney, "Short-term electricity demand and gas price forecasts using wavelet transforms and adaptive models," *Energy*, vol. 35, no. 9, pp. 3674–3685, 2010.
- [30] Z. Qian, Y. Pei, H. Zareipour, and N. Chen, "A review and discussion of decomposition-based hybrid models for wind energy forecasting applications," *Applied energy*, vol. 235, pp. 939–953, 2019.
- [31] M. Hu, G. Zheng, Z. Su, L. Kong, and G. Wang, "Short-term wind power prediction based on improved variational modal decomposition, least absolute shrinkage and selection operator, and bigru networks," *Energy*, p. 131951, 2024.
- [32] S. Yang and X. Qian, "A combined model based on poa-vmd secondary decomposition and lstm for ultra-short-term wind power forecasting," *Journal of Renewable and Sustainable Energy*, vol. 16, no. 3, 2024.
- [33] L. Jiang and Y. Wang, "A wind power forecasting model based on data decomposition and cross-attention mechanism with cosine similarity," *Electric Power Systems Research*, vol. 229, p. 110156, 2024.
- [34] J. Wang, X. Tang, and W. Jiang, "A deterministic and probabilistic hybrid model for wind power forecasting based improved feature screening and optimal gaussian mixed kernel function," *Expert Systems with Applications*, vol. 251, p. 123965, 2024.
- [35] W. Lu, J. Duan, P. Wang, W. Ma, and S. Fang, "Short-term wind power forecasting using the hybrid model of improved variational mode decomposition and maximum mixture correntropy long short-term memory neural network," *International Journal of Electrical Power & Energy Systems*, vol. 144, p. 108552, 2023.
- [36] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 4144–4147.
- [37] M. A. Colominas, G. Schlotthauer, and M. E. Torres, "Improved complete ensemble emd: A suitable tool for biomedical signal processing," *Biomedical Signal Processing and Control*, vol. 14, pp. 19–29, 2014.
- [38] J. S. Richman, D. E. Lake, and J. R. Moorman, "Sample entropy," in *Methods in enzymology*. Elsevier, 2004, vol. 384, pp. 172–184.
- [39] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.
- [40] M. Galphade, V. Nikam, B. Banerjee, and A. W. Kiwelekar, "Comparative analysis of wind power forecasting using lstm, bilstm, and gru," in *International Conference on Frontiers of Intelligent Computing: Theory and Applications*. Springer, 2022, pp. 483–493.
- [41] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2018, pp. 1394–1401.
- [42] Sotavento, "Sotavento wind farm," Accessed, 2024. [Online]. Available: <https://www.sotaventogalicia.com/en/technical-area/real-time-data/real-time-data-instantaneous-wild-farm/>