International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2390349

Short Term Wind Forecasting Using Machine Learning Models with Noise Assisted Data Processing Method

Janbhi Sharma*[1], Heeba Faiyaz[2], Advin Manhar[3]
*[1] Student, B.Tech(CSE), Amity University Chhattisgarh, Raipur, Chhattisgarh, India
[3] Assistant Professor, B.Tech(CSE), Amity University Chhattisgarh, Raipur, Chhattisgarh, India

Article History:

Accepted: 01 June 2023
Published: 12 June 2023

Publication Issue
Volume 9, Issue 3
May-June-2023

Page Number
459-472

In order to address the complex and intermittent nature of wind, this thesis proposes an innovative approach to enhance the accuracy of short-term wind forecasting. By leveraging the strengths of different methods while mitigating their weaknesses, a robust hybrid model is developed. The methodology incorporates Empirical Mode Decomposition (EMD), a data-adaptive denoising technique, to break down the signals into meaningful components called Intrinsic Mode Functions (IMFs), along with a residue. However, EMD is known to suffer from the mode mixing problem, where different scales of signals are erroneously mixed within IMFs, causing signal intermittency. To overcome this challenge, Ensemble Empirical Mode Decomposition is introduced, utilizing an ensemble of white noise to establish a uniform reference frame in the time-frequency space. By doing so, the added noise effectively collates signal portions with similar scales into a single IMF. Subsequently, the IMFs and residue obtained from both EMD and Ensemble Empirical Mode Decomposition are fed into a Convolutional Neural Network (CNN). The performance of this hybrid model is then compared against benchmark models such as Bi LSTM and LSTM. Evaluation is conducted based on two critical factors: performance metrics including MSE, MAE, RMSE, MAPE, and loss, as well as the time required for testing. Through a comprehensive analysis of these factors, the superior performance of the hybrid model is determined, thereby enhancing the prospects of reliable wind forecasting.

Keywords : Empirical Mode Decomposition; Ensemble Empirical Mode Decomposition; Intrinsic Mode Function; Convolutional Nueral Network

I. INTRODUCTION

OBJECTIVE

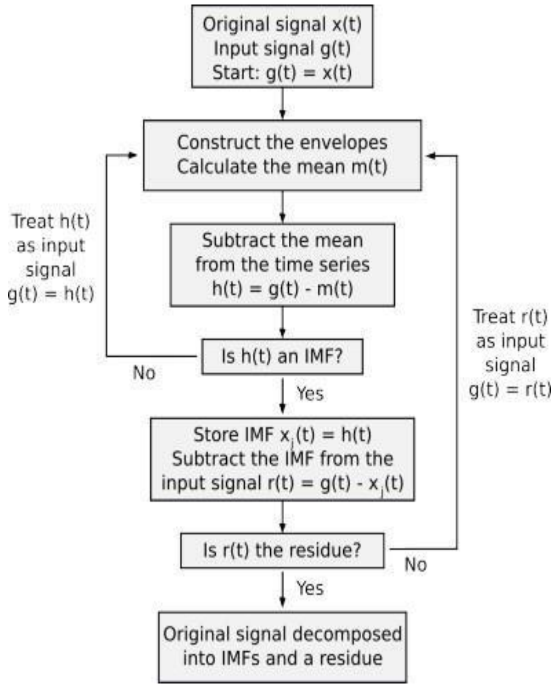
In order to address the challenge of finding the optimal model for accurate wind forecasting, we have developed a hybrid approach that combines the strengths of Convolutional Neural Networks (CNNs) and Ensemble Empirical Mode Decomposition. CNNs are neural networks that employ convolutions, allowing them to efficiently analyze matrices or tensors through a sliding window technique. By utilizing kernels to capture local invariant features and weight sharing, the number of trainable parameters in

CNNs is significantly reduced [3]. This research focuses on exploring machine learning algorithms for wind data forecasting. To enhance the accuracy of the forecasts, the study incorporates data denoising techniques such as Empirical Mode Decomposition and Ensemble Empirical Mode Decomposition, which are based on the Hilbert Huang Transform for nonlinear non-stationary data. The Hilbert Huang Transform combines the Hilbert Spectrum and Empirical Mode Decomposition, enabling analysis in the time-frequency domain without the need for data conversion, while also adapting to the characteristics of the data. The research also evaluates the model's performance with and without the implementation of data denoising techniques to gain insights into their impact on forecasting accuracy. Finally, by comparing the performance of the proposed hybrid model with state-of-the-art benchmark models like LSTM and Bi-directional LSTMs, this work provides evidence supporting the superiority of the hybrid approach in wind forecasting.

VARIOUS MODELS USED IN THE RESEARCH

A. EMD

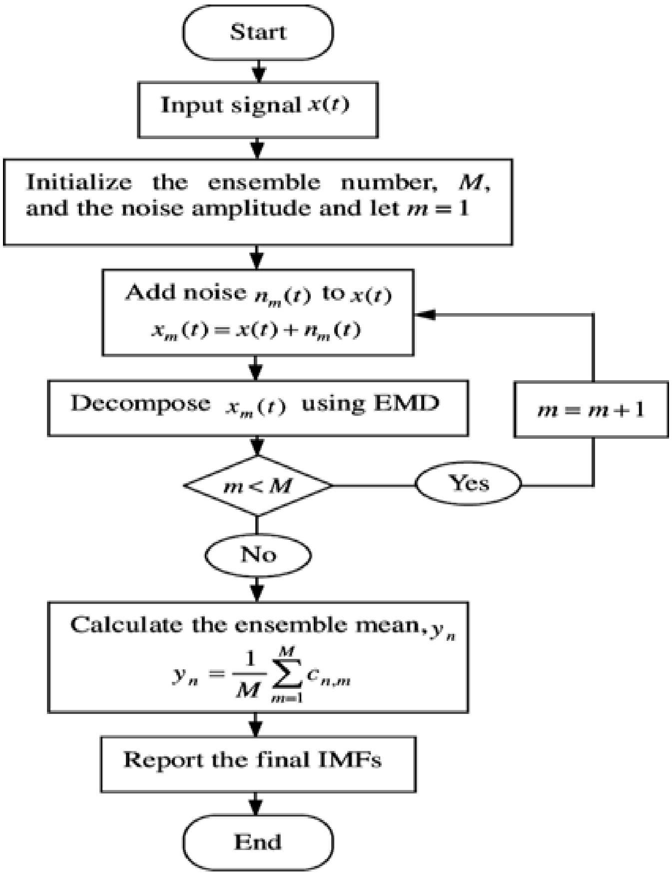
Empirical Mode Decomposition (EMD) serves as a valuable technique for unraveling intricate data into more manageable components. It proves particularly advantageous when grappling with time-varying data that lacks a regular pattern. Rather than assuming the data comprises simple waves, EMD diligently seeks distinct segments within the data that possess their own discernible patterns. By partitioning the data into smaller fragments known as Intrinsic Mode Functions (IMFs), each IMF encapsulates a specific portion of the data, complete with its own dynamic patterns. The beauty of EMD lies in its flexibility, as it imposes no rigid guidelines on the input data. Instead, it adapts adeptly to diverse data types, efficiently deciphering the optimal approach to simplifying the data into its constituent components. EMD boasts an array of practical applications across various domains. It aids in fault detection within machinery, such as bearings, facilitates the analysis of medical data, enables comprehensive examination of power signals, and supports the investigation of seismic signals. In these fields, EMD empowers researchers to comprehend and analyze complex data more effectively by disentangling it into its meaningful constituents.



Flow chart of EMD

B. EEMD

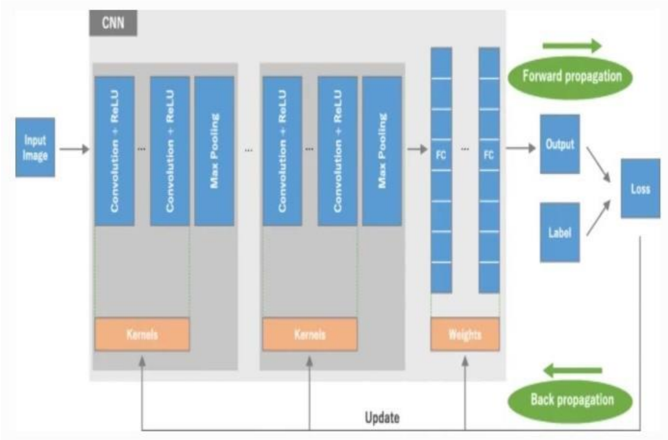
Ensemble Empirical Mode Decomposition (EEMD) emerges as a refined iteration of Empirical Mode Decomposition (EMD), designed to elevate the precision of the decomposition process by leveraging the power of noise. Consider a scenario where you aim to dissect a signal into distinct components. EEMD comes to the rescue by injecting random noise into the signal, effectively establishing a reference frame that facilitates the separation of its various constituents. In the EEMD procedure, the noisy signal undergoes multiple analyses. With each iteration, the introduced noise gradually dissipates through averaging, leaving behind the true signal component that remains consistent throughout the process. The brilliance of incorporating noise lies in its ability to combat the conundrum of "mode mixing." Mode mixing arises when different signal parts overlap, impeding their distinction. By introducing noise, EEMD ensures a more effective separation of each signal component. Picture the challenge of isolating distinct musical instruments playing simultaneously. The noise integrated into EEMD creates a transparent reference frame, simplifying the identification and isolation of each instrument's unique sound. In essence, EEMD serves as an enhanced rendition of EMD, employing noise to refine the accuracy of signal decomposition. The noise acts as a reliable reference frame, bolstering the efficiency of separating different signal parts and effectively circumventing mode mixing.



Flow chart of EEMD

C. CNN

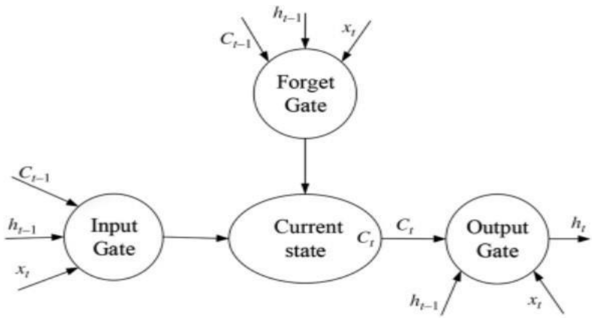
A groundbreaking form of artificial intelligence, the Convolutional Neural Network (CNN) draws inspiration from the extraordinary visual processing capabilities of the human brain. Its remarkable aptitude lies in the profound analysis and comprehension of images or data exhibiting a grid-like structure, such as time series data. Picture this: you possess an image depicting a dog, and your objective is for the computer to accurately identify it as a dog. The CNN accomplishes this by systematically dissecting the image into smaller constituents known as "filters" or "feature detectors." These remarkable filters adeptly recognize specific patterns crucial for object identification, encompassing edges, textures, and shapes. The CNN proceeds to scan the entire image, applying these filters in a systematic manner while diligently focusing on diverse regions in search of the acquired patterns. Each filter provides a "feature map" that illuminates the precise locations where these patterns manifest within the image. As the CNN delves deeper through multiple layers, it assimilates more intricate patterns by synthesizing the simpler patterns acquired from previous layers. Gradually, it constructs a hierarchical framework of features, progressing from rudimentary aspects such as edges towards increasingly sophisticated attributes like eyes, ears, or the overall shape of a dog. Ultimately, the CNN incorporates these well-learned features into fully connected layers, where it effectively formulates predictions based on the knowledge acquired. In scenarios involving image classification, the CNN can discern the likelihood of the image encompassing a dog or any other object under consideration. The CNN's distinct advantage resides in its innate ability to autonomously learn and extract vital features from the data, bypassing the necessity for explicit feature engineering. This remarkable capability allows the CNN to capture intricate patterns and establish relationships, making it an ideal solution for diverse tasks encompassing image recognition, object detection, and even time series analysis.



CNN internal structure

D. LSTM

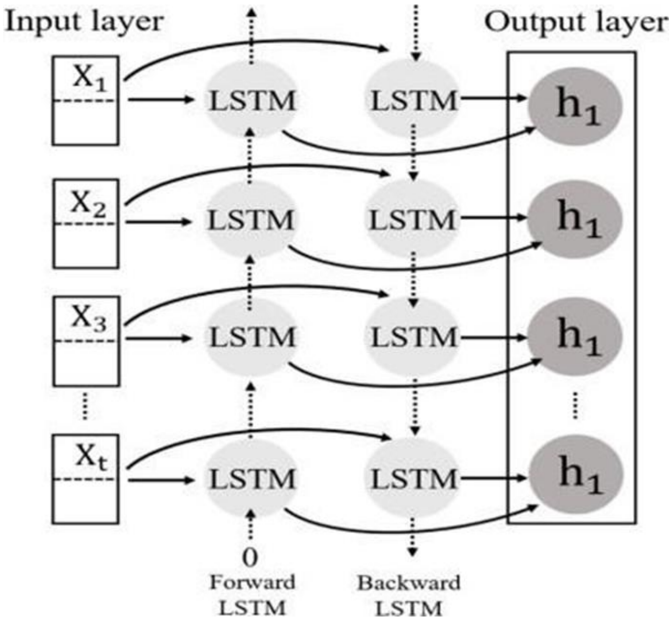
Long Short-Term Memory (LSTM) is an artificial neural network that excels in processing and comprehending sequential data, including text and time series. To illustrate its functionality, imagine immersing yourself in a captivating storybook where understanding the current chapter necessitates recalling the events from previous chapters. LSTM operates similarly by incorporating a memory component that adeptly stores and retrieves information from prior steps or time points within the sequence. Its architecture comprises interconnected nodes known as "cells," which facilitate seamless information exchange. These cells consist of three fundamental components: an input gate, a forget gate, and an output gate. The input gate controls the assimilation of new information into the memory, while the forget gate determines which information is irrelevant and should be discarded. On the other hand, the output gate regulates the extent to which the stored memory influences the output at a given step. By harnessing the power of memory, LSTM effectively captures long-term dependencies within the sequence, enabling it to grasp intricate patterns and relationships in the data.



LSTM unit structure

E. Bi-directional LSTM

The Bi-LSTM, an impressive advancement of the LSTM model, introduces a paradigm shift in the realm of sequential data analysis and prediction. Let's immerse ourselves in a captivating book, where we yearn to unravel the full context of a specific sentence. Naturally, we scan the sentences preceding and succeeding it to gain profound comprehension. Similarly, the Bi-LSTM operates with the same insightful approach, seamlessly assimilating both past and future information. While a conventional LSTM progresses forward, meticulously scrutinizing past data to forecast future patterns, the Bi-LSTM takes a remarkable leap forward. By seamlessly incorporating an additional layer that delves into the sequence in reverse, it unveils vital insights from the future. Merging the outputs from both the forward and backward LSTM layers, the Bi-LSTM expertly captures intricate dependencies and patterns, seamlessly intertwining the past and future contexts of the input sequence. This unparalleled capability makes it an optimal choice for tasks that demand a holistic understanding of the entire sequence, spanning sentiment analysis, speech recognition, and machine translation. Picture the Bi-LSTM as a dynamic duo of LSTM models, working in perfect synergy—one embarking on an insightful journey from the sequence's beginning to its end, while the other astutely explores the path in reverse. By harnessing the power of information from both directions, the Bi-LSTM unlocks a deeper level of comprehension and analysis, empowering us to unravel the enigmatic intricacies of sequential data.



Bi-LSTM process structure

II. METHODOLOGY

DATA PREPROCESSING AND DENOISING

A. Fetching dataset: The dataset is taken from the NREL site. The parameters, which have to be taken in the dataset, are fed in the URL. The specific parameter is fed in the URL. The GET URL brings up the data and loads up the CSV file into Pandas dataframe.

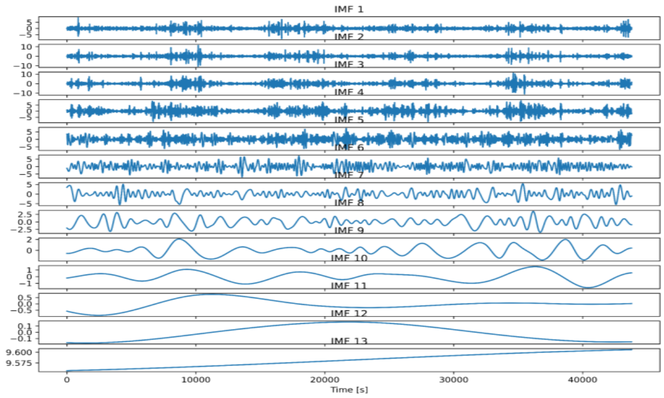
Request Parameters

Parameter	Required	Value	Description
api_key	Yes	**Type:** string **Default:** None	Your developer API key. See API keys for more information.
wkt	Yes	**Type:** well-known text point string **Default:** None	A well-known text (WKT) representation of the geometry for which to extract data. May be a point or polygon geometry.
attributes	Yes	**Type:** comma delimited string array **Default:** None **Options:** *pressure_40m, temperature_100m, temperature_120m, temperature_40m, temperature_80m, winddirection_100m, winddirection_120m, winddirection_40m, winddirection_80m, windspeed_100m, windspeed_120m, windspeed_40m, windspeed_80m, pressure_100m*	Each specified attribute will be returned as a column in the resultant CSV download.
names	Yes	**Type:** comma delimited integer array **Default:** 2014 **Options:** *2014*	The year(s) for which data should be extracted.
interval	No	**Type:** integer **Default:** 60 **Options:** *15, 60*	Desired data timestep resolution in minutes.
utc	No	**Type:** true or false **Default:** true	Pass true to retrieve data with timestamps in UTC. Pass false to retrieve data with timestamps converted to local time of data point (without daylight savings time).
leap_day	No	**Type:** true or false **Default:** false	Pass true to retrieve data including leap day (where appropriate). Pass false to retrieve data excluding leap day.
full_name	No	**Type:** string **Default:** None	The full name of the user requesting data.
email	Yes	**Type:** email string **Default:** None	An active email for the user requesting data. This email will be used to deliver the extracted data.
affiliation	No	**Type:** string **Default:** None	The organization with which the user requesting the data is affiliated.
reason	No	**Type:** string **Default:** None	The reason that the user is requesting the data.

API request parameters from NREL

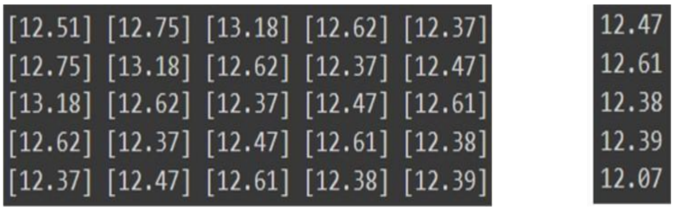
B. Data Cleaning and Preprocessing: The data obtained from API fetching in the form of a CSV file is then inspected, visualized, and cleaned to remove inconsistent and NaN values and is normalized using the Pandas library framework. The columns are also modified and the date is string formatted to obtain the values in the appropriate date-time format required for 13 our model. After removing the redundant data and setting the look back function, the cleaned data is ready for training and prediction.

C. Data Denoising (EMD / EEMD) : Utilizing the powerful PyEmd Library, the preprocessed data undergoes a transformative decomposition process, splitting it into two distinct components: the mesmerizing Intrinsic Mode Functions (IMFs) and the intriguing residue. To ensure seamless integration with the machine learning model, a carefully crafted while loop efficiently feeds the IMFs into the network. Simultaneously, employing the ingenious Ensemble Empirical Mode Decomposition (EEMD) technique, the same preprocessed data embarks on an alternative decomposition journey. Within this transformative process, the data gracefully unfolds into its constituent IMFs and a captivating residue. EEMD employs an ensemble of signals, expertly enhanced with white noise, sifting through each iteration to converge on the ultimate truth: the majestic mean. By harnessing the statistical characteristics of white noise, EEMD brings forth a remarkable evolution beyond its predecessor, the original EMD. This revolutionary approach eliminates the need for subjective criterion selection, granting the freedom to naturally separate scales without any a priori intermittence test. The fusion of PyEmd's EMD and the enchanting EEMD elevates the study's methodology to new heights, unraveling the intricate tapestry of the data's hidden secrets.



IMFs of the wind speed data obtained using EMD

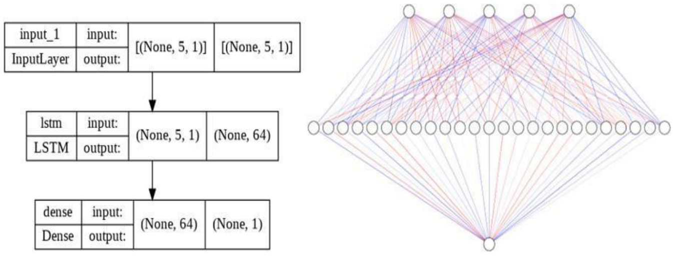
D. Feeding the Data into the Network: The dynamic nature of time series data lends itself to a fascinating transformation, akin to the art of supervised learning. Through the clever application of the window shifting technique, the given time series dataset undergoes a restructuring that mirrors a supervised-learning framework. This intricate process entails leveraging the power of previous time step data as the input, while the subsequent single time data assumes the role of the output variable. Referred to as the sliding window method, this approach harnesses the predictive capabilities of prior time steps to anticipate the future with precision. In the realm of time series analysis and statistical methods, this technique is commonly known as the lag method, while its succinct alias, the window method, is equally popular. The window width, or the number of previous time steps, plays a crucial role in shaping the predictive landscape, serving as the foundation for the sliding window's effectiveness. By adopting this fundamental methodology, any time series dataset can seamlessly transition into a supervised learning problem, unlocking new possibilities for analysis and insights. In the case of the wind speed dataset, a window length of five has been selected, signifying the input dimension of the neural network. As the window length shapes the input dimension, the neural network becomes equipped to unravel the intricacies and patterns inherent in the time series data.



[12.51]	[12.75]	[13.18]	[12.62]	[12.37]	12.47
[12.75]	[13.18]	[12.62]	[12.37]	[12.47]	12.61
[13.18]	[12.62]	[12.37]	[12.47]	[12.61]	12.38
[12.62]	[12.37]	[12.47]	[12.61]	[12.38]	12.39
[12.37]	[12.47]	[12.61]	[12.38]	[12.39]	12.07

Input and Output data for Supervised learning

The window length is the input dimension of the Neural network which is 5.



Structure of the Neural Network used for all models

III. ML MODELS - TRAINING AND TESTING

The wind speed dataset having the number of rows equal to 43798 is split into Train/validate/test as 70/10/20. For all the basic models the original data is directly given to the Neural Networks (here LSTM, Bi_LSTM, CNN).

The structure of all the NN blocks is the same as mentioned in the previous session. For EMD models:

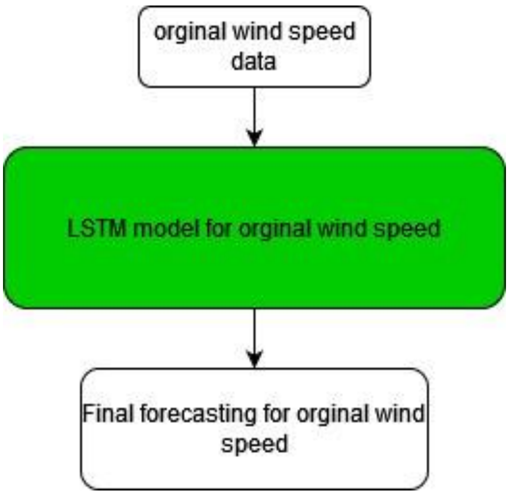
1. Start from the original wind speed data and do the preprocessing
2. Give the obtained data to EMD (Empirical Mode Decomposition). It will create various IMFs (Intrinsic Mode Functions).
3. Push each IMFs into the Neural Network of a similar structure.
4. Each model creates forecasting according to the IMFs.
5. Now just merge/ aggregate all the results obtained separately.
6. After merging compare the result with test data to obtain various error metrics.

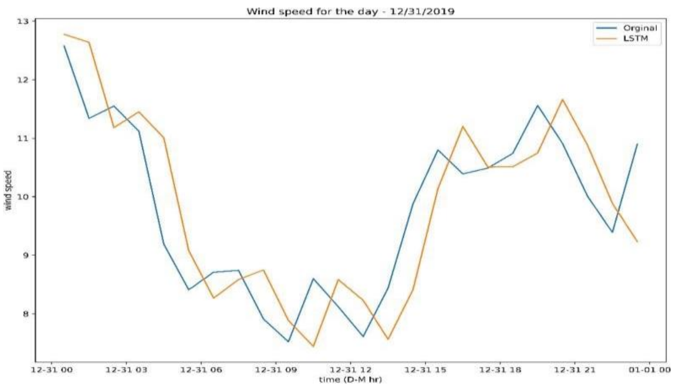
For EEMD models:

1. Start from the original wind speed data and do the preprocessing
2. Give the obtained data to EEMD (Ensemble Empirical Mode Decomposition). It will create various IMFs (Intrinsic Mode Functions).
3. Using IMFs and wind speed finds the noise created by EEMD
4. Push each IMFs and noise into the Neural Network of a similar structure.
5. Each model creates forecasting according to the IMFs, noise.
6. Now just merge/ aggregate all the results of IMFs obtained separately.
7. Remove forecasted noise from the aggregate to obtain the final result.
8. Use the final result with test data to obtain various error metrics. The flowchart and results (for day 12/31/2019) for all the nine models in the paper are given below. The input data is hourly, therefore the results for one day contain 24 points.

A. LSTM

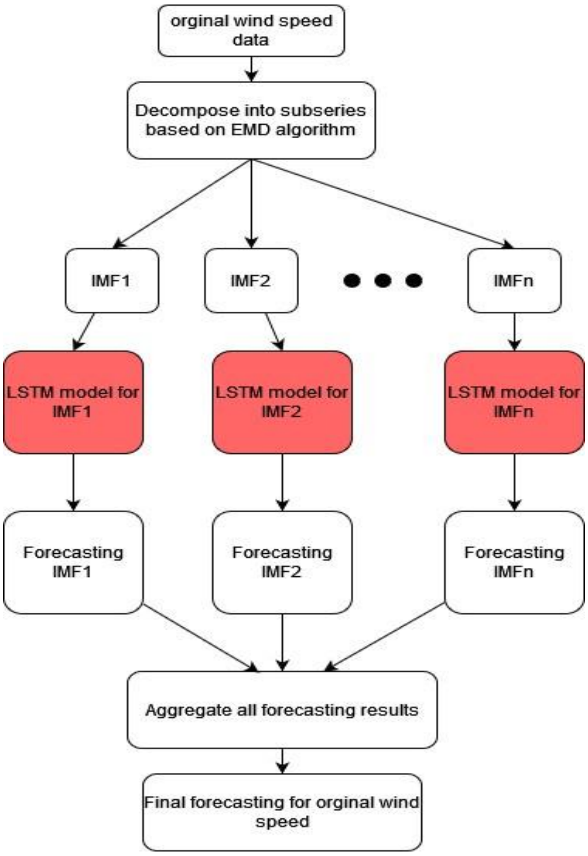
Flowchart of LSTM model



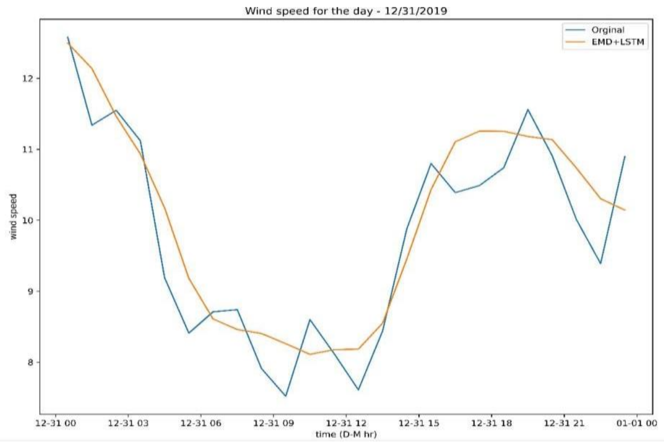


Wind speed Actual and Predicted using LSTM model

B. EMD LSTM

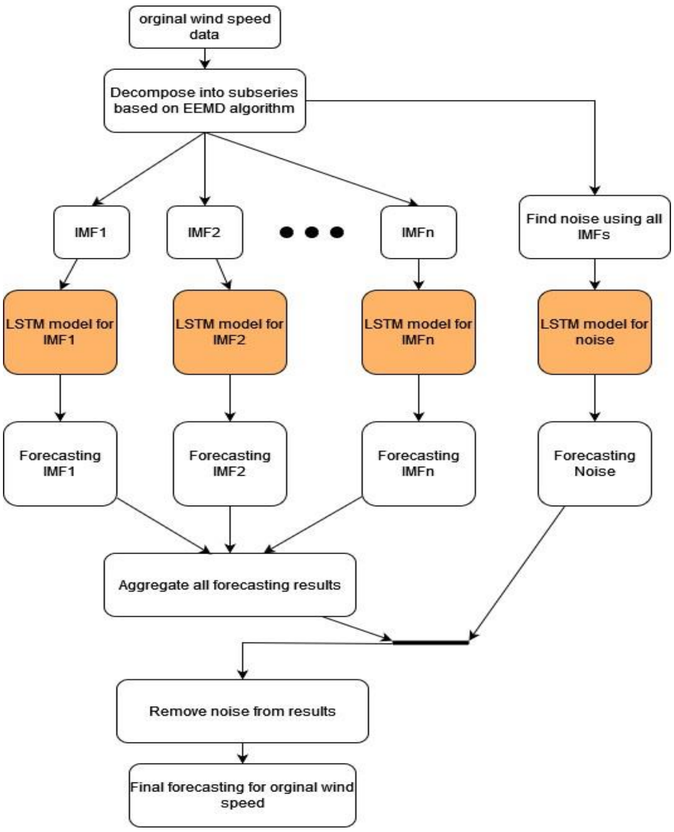


Flowchart of EMD LSTM model

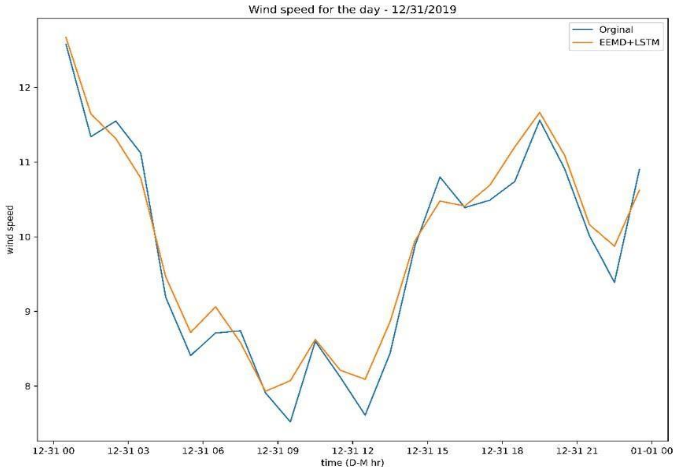


Wind speed Actual and Predicted using EMD LSTM model

C. EEMD LSTM

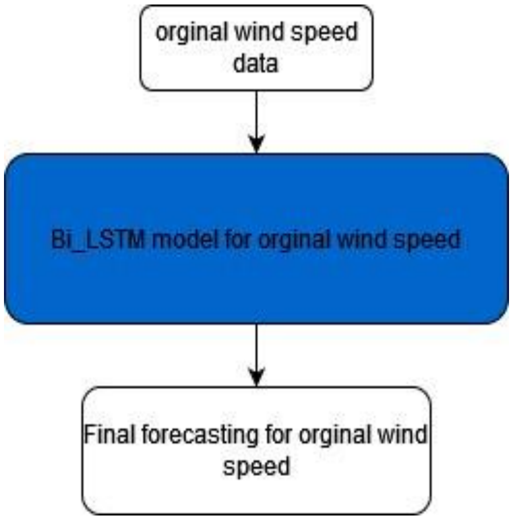


Flowchart of EEMD LSTM model

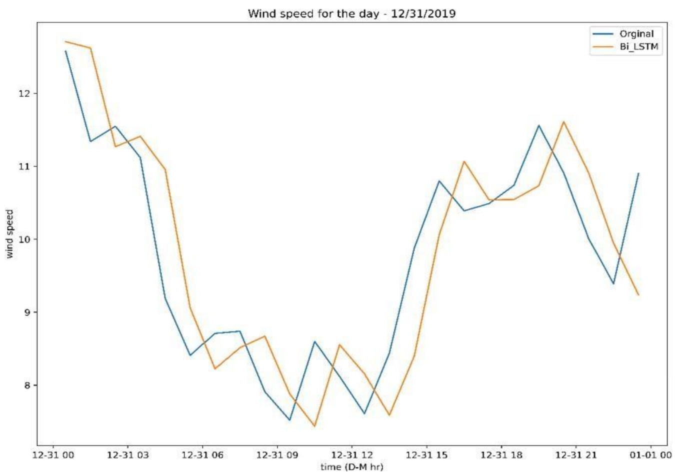


Wind speed Actual and Predicted using EEMD LSTM model.

D. Bi – LSTM

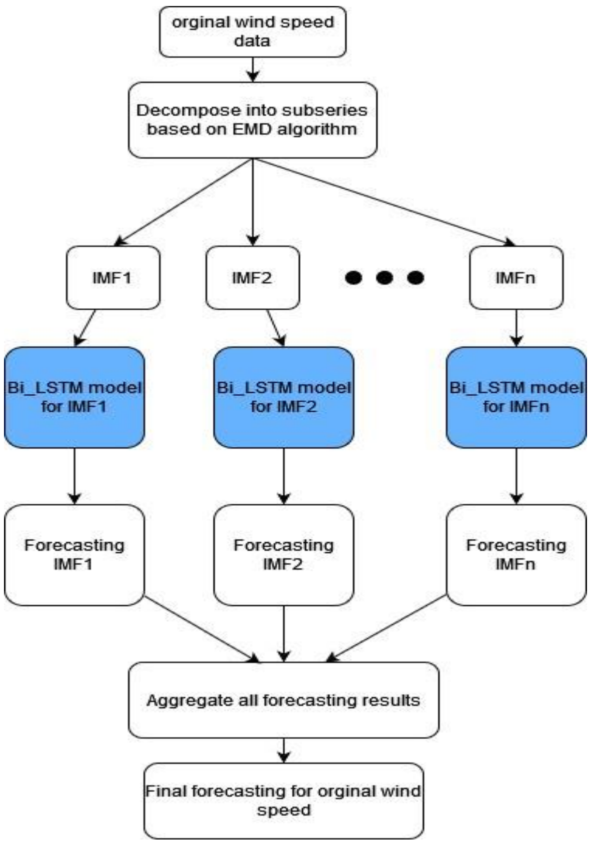


Flowchart of Bi-LSTM model

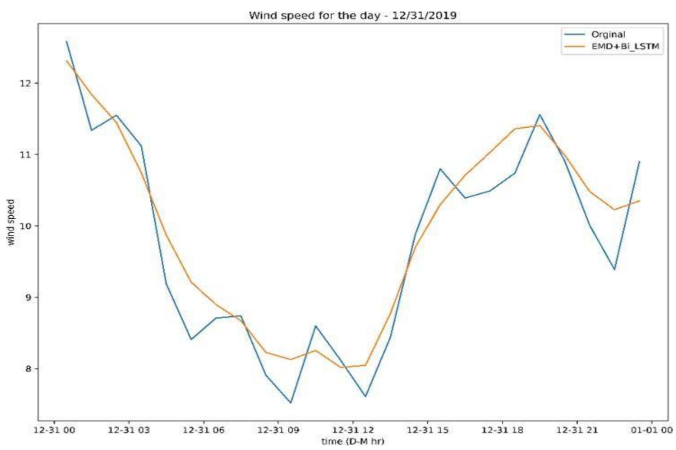


Wind speed Actual and Predicted using Bi-LSTM model

E. EMD Bi- LSTM

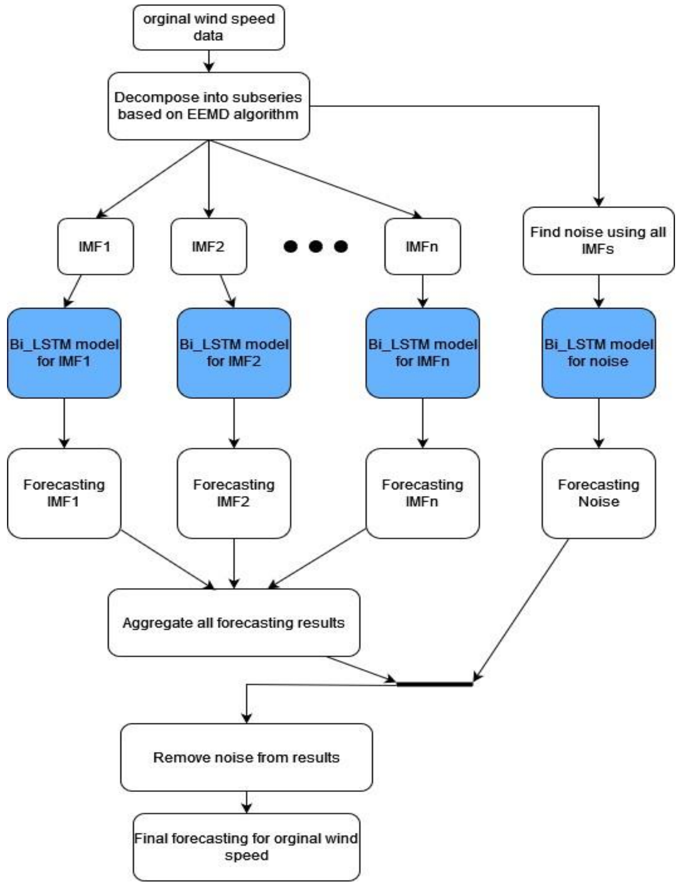


Flowchart of EMD Bi-LSTM model

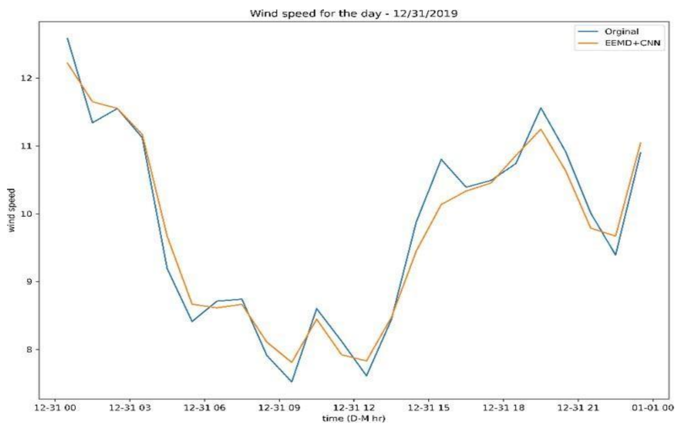


Wind speed Actual and Predicted using EMD Bi-LSTM model

F. EEMD Bi - LSTM



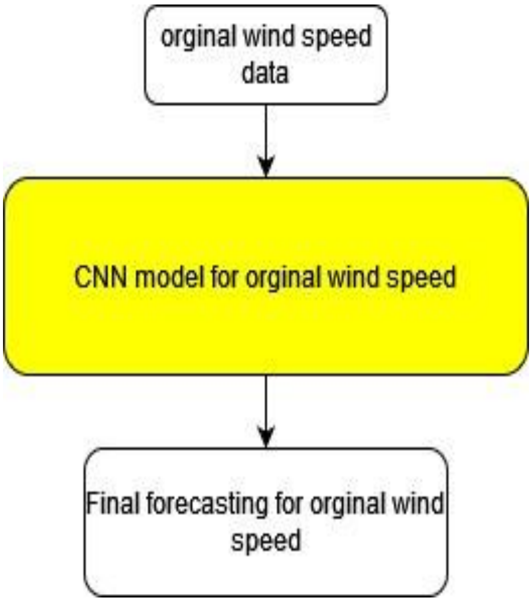
Flowchart of EEMD Bi-LSTM model



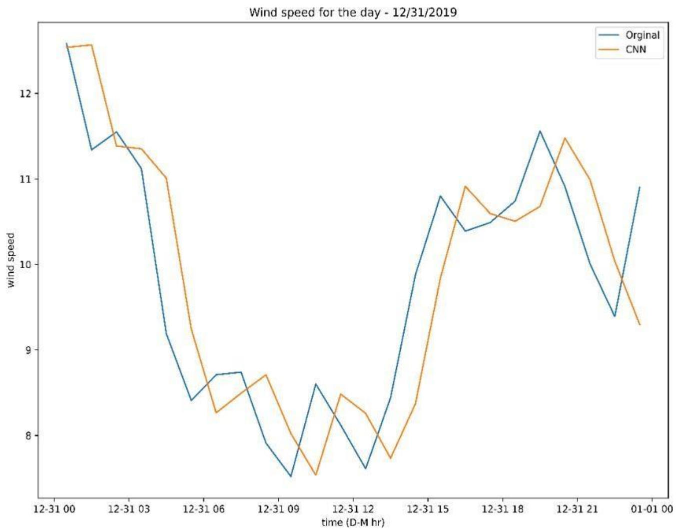
Wind speed Actual and Predicted using EEMD Bi-LSTM model

G. CNN

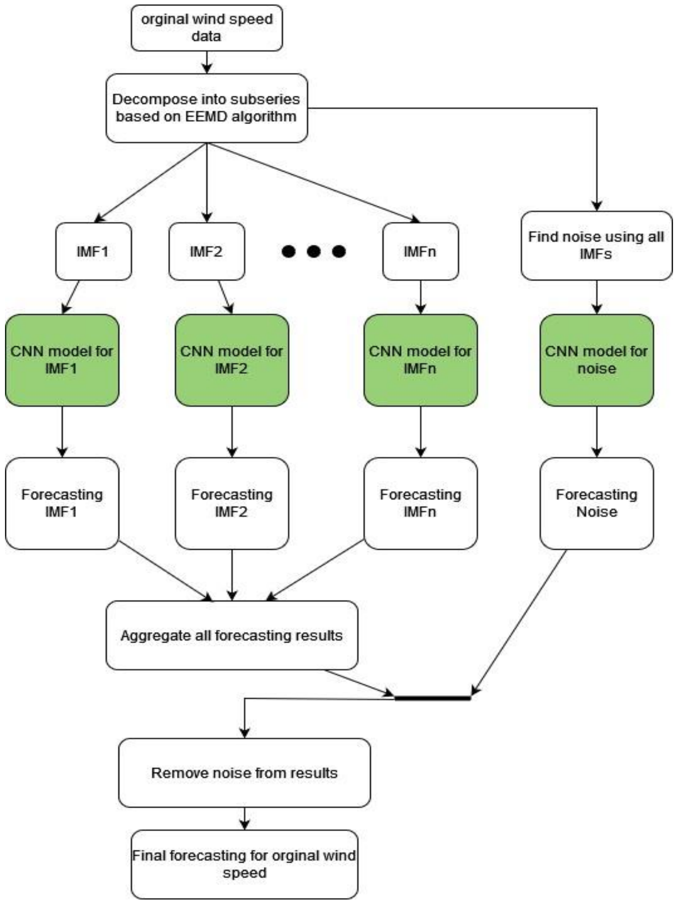
Flowchart of CNN model



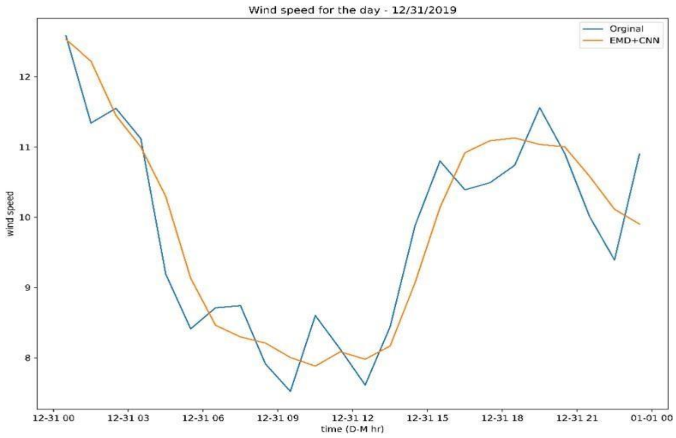
Wind speed Actual and Predicted using CNN model



H. EMD CNN

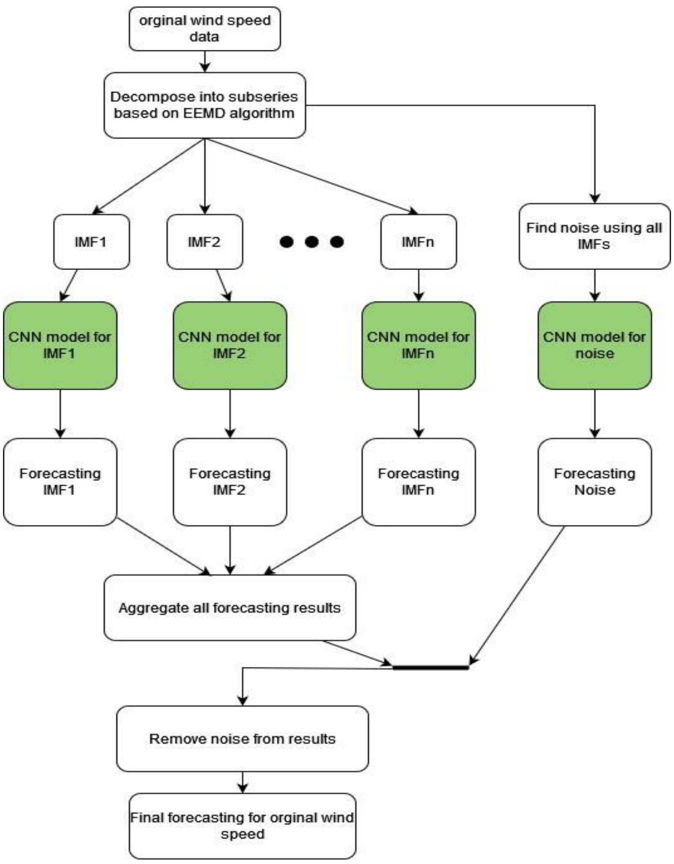


Flowchart of EMD CNN model

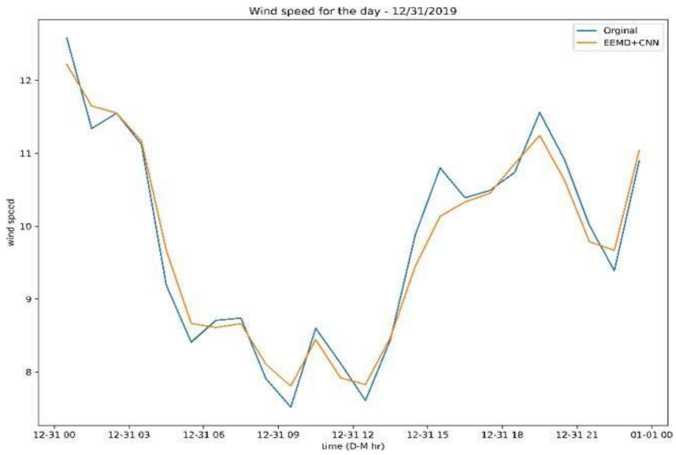


Wind speed Actual and Predicted using EMD CNN model

I. EEMD CNN



Flowchart of EEMD CNN model



Wind speed Actual and Predicted using EEMD CNN model

IV. RESULT

To understand the efficacy of our proposed hybrid model, comparisons at various levels were performed by contrasting its performance with other models. Firstly, every model was run for 15 iterations and the corresponding values for metrics of evaluations were noted. These models were run under google colab environment in a system with vIntel® Core™ i5-8250U CPU @ 1.60GHz × 8 processor and 7.7 GiB memory with a cloud-based GPU runtime environment. Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Validation Loss and Time takes for testing were chosen as the parameters for performance evaluation.

The average values obtained under each metric after 15 iterations were tabulated and is as follows:

Table 4.1: Analysis of various models based on metrics of evaluation

MODEL NAME	MSE	MAE	RMSE	MAPE	LOSS	TIME(s)
LSTM	1.2792	0.6997	1.1309	0.0957	1.0965	1.1878
EMD+LSTM	0.6462	0.5376	0.8035	0.0795	0.0720	15.0466
EEMD+LSTM	0.2600	0.3730	0.5097	0.0531	0.0172	16.5667
Bi LSTM	1.1451	0.6743	1.1101	0.0955	0.9551	1.6973
EMD+Bi LSTM	0.5926	0.4982	0.7697	0.1258	0.0640	21.9533
EEMD+Bi LSTM	0.2451	0.3609	0.4947	0.0515	0.0170	21.4600
CNN - 1D	1.1105	0.6750	1.0538	0.0990	1.0270	0.4514
EMD+CNN	0.6780	0.5287	0.8232	0.0805	0.0457	5.4613
EEMD+CNN	0.2068	0.3234	0.4543	0.0470	0.0163	6.1146

A good model is expected to have lower values in these metrics. It was observed that the EEMD +CNN model turned out to be the better performing hybrid model for the short-term hour ahead wind forecasting as it outperforms all the other models in five out of six metrics (all except time taken for testing). It can also be noted that the time taken is still optimal as it is more only with respect to basic CNN and EMD+CNN model and is far better than Bi LSTM and LSTM models. It is also expected the same as EEMD is a serial and computationally expensive process that requires more time than a model run without EEMD. It is observed that decomposition-based forecasting models outperform the individual models in all metrics of evaluation except time. Among the two Decomposition based models (EMD and EEMD), the EEMD-based models exhibit better performance accuracy with lower MSE, MAE, RMSE, MAPE, and loss values

V. CONCLUSION

The prominence of wind energy generation and integration with the Grid has encouraged reliable and most accurate forecasting approaches. Based on the experiments, the following conclusions can be drawn on wind forecasting. Wind speed forecast was successfully performed with three ML algorithms namely LSTM, 31 Bi-LSTM, and CNN with and without the data denoising techniques. Performance evaluation of Machine Learning Algorithms was done for comprehensive comparison based on two important factors: 1. Performance metrics (MSE, MAE, RMSE, MAPE, Loss) 2. Time was taken during testing. Convolutional Nueral Network proved to be a better performing model than the Recurrent Nueral Network Models namely LSTM and Bi-LSTM. Denoising time took more time as expected and outperformed the basic technique in terms of forecasting accuracy and losses. In the following section, some of the other explorations done during the course of this project and the avenues identified to expand the domain of this project work will be discussed. This will be followed by any roadblocks encountered in each one of these explorations, overcoming which, further exploration is possible in the future.

VI. SCOPE OF FUTURE WORK

From the research work carried out so far, EEMD turned out to be a computationally expensive data denoising technique. Other data denoising techniques like CEEMD and CEEMDAN need to be dwelled upon. EEMD also suffers from a problem, that if the no of trails is less, a small amount of noise does get mixed with the time-series signal. CEEMD generates a collection of independent Gaussian white noise and a complimentary pair for each white noise to perfectly cancel each other. EEMD also suffers from another problem, that is, when the no of trails increases, then the number of sifting processes also increases. In order to reduce the number of trails while retaining the ability to solve the mode mixing problem, we go with CEEMDAN. CEEMD is advantageous over EEMD and CEEMD in the following ways. First, It introduces an extra noise coefficient vector w to control the noise level at each decomposition stage. Secondly, the reconstruction is complete and noise-free and lastly, it requires fewer trails than EEMD and CEEMD. All this points out to go for a better data denoising method, that is, CEEMDAN. Some machine learning ensemble methods such as bagging and boosting can also be implemented along with this more noise-assisted data analysis method. Apart from these, the addition of wind parameters is one of the areas that can be explored since it is an important determinant for wind power production. Distributed computation for the process can also be carried out.

VII. REFERENCES

[1]. Wu, Zhaohua & Huang, Norden. (2009). Ensemble Empirical Mode Decomposition: a Noise-Assisted Data Analysis Method. Advances in Adaptive Data Analysis. 1. 1-41. 10.1142/S1793536909000047.

[2]. Soman, S.S., Zareipour, H., Malik, O., & Mandal, P. (2010). A review of wind power and wind speed forecasting methods with different time horizons. North American Power Symposium 2010, 1-8.

[3]. Trebing, Kevin & Mehrkanoon, Siamak. (2020). Wind speed prediction using multidimensional convolutional neural networks. 713-720. 10.1109/SSCI47803.2020.9308323.

[4]. Wang, Tong & Zhang, Mingcai & Yu, Qihao & Zhang, Huyuan. (2012). Comparing the

application of EMD and EEMD on time-frequency analysis of seismic signal. Journal of Applied Geophysics. 83. 29-34. 10.1016/j.jappgeo.2012.05.002.

[5]. J. K.U. and B. C. Kovoor, "EEMD-based Wind Speed Forecasting system using Bidirectional LSTM networks," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-9, doi: 10.1109/ICCCI50826.2021.9402648.

[6]. Madasthu, Santhosh & Venkaiah, Chintham & Kumar, D.M.. (2018). Ensemble empirical mode decomposition based adaptive wavelet neural network method for wind speed prediction. Energy Conversion and Management. 168. 482-493. 10.1016/j.enconman.2018.04.099.

[7]. Shao, Bilin & Song, Dan & Bian, Genqing & Zhao, Yu. (2021). Wind Speed Forecast Based on the LSTM Neural Network Optimized by the Firework Algorithm. Advances in Materials Science and Engineering. 2021. 1-13. 10.1155/2021/4874757.

[8]. Doucoure, Boubacar & Agbossou, K. & Cardenas, Alben. (2016). Time series prediction using artificial wavelet neural network and multiresolution analysis: Application to wind speed data. Renewable Energy. 92. 202-211. 10.1016/j.renene.2016.02.003.

[9]. M. E. Torres, M. A. Colominas, G. Schlotthauer and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 4144-4147, doi: 10.1109/ICASSP.2011.5947265.

[10]. Huang, Norden & Shen, Zheng & Long, Steven & Wu, M.L.C. & Shih, Hsing & Zheng, Quanan & Yen, Nai-Chyuan & Tung, Chi-Chao & Liu, Henry. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences. 454. 903-995. 10.1098/rspa.1998.0193.

[11]. Y. Su, J. Yu, M. Tan, Z. Wu, Z. Xiao and J. Hu, "A LSTM Based Wind Power Forecasting Method Considering Wind Frequency Components and the Wind Turbine States," 2019 22nd International Conference on Electrical Machines and Systems (ICEMS), 2019, pp. 1-6, doi: 10.1109/ICEMS.2019.8921671.

[12]. K. Moharm, M. Eltahan and E. Elsaadany, "Wind Speed Forecast using LSTM and Bi-LSTM Algorithms over Gabal El-Zayt Wind Farm," 2020 International Conference on Smart Grids and Energy Systems (SGES), 2020, pp. 922-927, doi: 10.1109/SGES51519.2020.00169.

[13]. X. -J. Liu, Z. -Q. Mi, B. Lu and W. Tao, "A Novel Approach for Wind Speed Forecasting Based on EMD and Time-Series Analysis," 2009 AsiaPacific Power and Energy Engineering Conference, 2009, pp. 1-4, doi: 10.1109/APPEEC.2009.4918088.

[14]. S. Jia, "A new method for the short-term wind speed forecasting," 2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 2011, pp. 1320-1324, doi: 10.1109/DRPT.2011.5994100.

[15]. Y. Ren, P. N. Suganthan and N. Srikanth, "A Comparative Study of Empirical Mode Decomposition-Based

[16]. Short-Term Wind Speed Forecasting Methods," in IEEE Transactions on Sustainable Energy, vol. 6, no. 1, pp. 236-244, Jan. 2015, doi: 10.1109/TSTE.2014.2365580.

[17]. Z. Xing, Y. Zhi, R. -h. Hao, H. -w. Yan and C. Qing, "Wind Speed Forecasting Model Based on Extreme Learning Machines and Complete Ensemble Empirical Mode Decomposition," 2020 5th Asia Conference on Power and Electrical Engineering (ACPEE), 2020, pp. 159-163, doi: 10.1109/ACPEE48638.2020.9136553.

[18]. C. M. Baby, K. Verma and R. Kumar, "Short term wind speed forecasting and wind energy estimation: A case study of Rajasthan," 2017 International Conference on Computer, Communications and Electronics (Comptelix), 2017, pp. 275-280, doi: 10.1109/COMPTELIX.2017.8003978.

[19]. A. J. Domingo, F. Carlo Garcia, M. L. Salvaña, N. J. C. Libatique and G. L. Tangonan, "Short Term Wind Speed Forecasting : A Machine Learning Based Predictive Analytics," TENCON 2018 - 2018 IEEE Region 10 Conference, 2018, pp. 1948-1953, doi: 10.1109/TENCON.2018.8650287.

[20]. H. Masrur, M. Nimol, M. Faisal and S. M. G. Mostafa, "Short term wind speed forecasting using Artificial Neural Network: A case study," 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2016, pp. 1-5, doi: 10.1109/ICISET.2016.7856485.

[21]. C. LI, X. KONG, X. WANG, F. ZHENG, Z. CHEN and Z. ZHOU, "Short-Term Wind Power Forecasting Method Based on Mode Decomposition and Feature Extraction," 2019 22nd International Conference on Electrical Machines and Systems (ICEMS), 2019

Cite this article as :

Janbhi Sharma, Heeba Faiyaz, Advin Manhar, "Short Term Wind Forecasting Using Machine Learning Models with Noise Assisted Data Processing Method", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 3, pp.459-472, May-June-2023. Available at doi : https://doi.org/10.32628/CSEIT2390349
Journal URL : https://ijsrcseit.com/CSEIT2390349