Regular Issue

Stacked LSTM for Short-Term Wind Power Forecasting Using Multivariate Time Series Data

Manisha Galphade[1,2]*, V. B. Nikam[1], Biplab Banerjee[3], Arvind W. Kiwelekar[4], Priyanka Sharma[5]

[1] Department of Computer Engineering & IT, Veermata Jijabai Technological Institute, Mumbai (India)
[2] School of Computing, MIT Art Design & Technology University, Pune (India)
[3] Center of Studies in Resources Engineering, IIT Bombay Mumbai, (India)
[4] Department of Information Technology, Dr Babasaheb Ambedkar Technological University, Lonere, Raigad (India)
[5] Director of Software Engineering (HPC AI R&D Lab), Fujitsu Research (FRIPL),AI Thought Leader, Bengaluru, Karnataka (India)

* Corresponding author: galphademanisha@gmail.com

Received 20 May 2023 | Accepted 10 May 2024 | Published 15 July 2024

Abstract

Currently, wind power is the fast growing area in the domain of renewable energy generation. Accurate prediction of wind power output in wind farms is crucial for addressing the challenges associated the power grid. This precise forecasting enables grid operators to enhance safety and optimize grid operations by effectively managing fluctuations in power generation, ensuring a reliable and stable energy supply. In recent years, there has been a significant rise in research and investigations conducted in this field. This study aims to develop a multivariate short-term wind power forecasting (WPF) model with the objective of enhancing forecasting precision. Among the various prediction models, deep learning models such as Long Short-Term Memory (LSTM) have demonstrated outstanding performance in the field of WPF. By adding multiple layers of LSTM networks, the model can capture more complex patterns. To improve the performance, data pre-processing is carried out using two techniques such as removal of missing values and imputing missing values using Random Forest Regressor (RFR). The comparison between the proposed Stacked LSTM model and other methods including vector autoregressive (VAR), Multiple Linear Regression, Gated Recurrent Unit (GRU) and Bidirectional LSTM (BiLSTM) has been experimented on two datasets. The experimental results show that after imputing missing values using RFR, the Stacked LSTM is optimized model for better performance than above mentioned reference models.

Keywords

Deep Learning, Long Short Term Memory, Multivariate, Renewable Energy , Time Series Data, Wind Power Forecasting.

DOI: 10.9781/ijimai.2024.07.002

I. Introduction

Renewable energy sources are developed and adopted since there is a decline in the cost of renewable technology, globally. This is for the reason that using conventional energy causes the environment irreparable harm [1]. Among them, wind power is an attractive option because of its low operating costs, low environmental impact, and high availability and sustainability. As a result, to assure the stability and safety of the power system, efficient and reliable wind power forecasting (WPF) methodologies are required. The three categories of WPF tasks are: (i) Short-Term Prediction: time range from several minutes to several hours; (ii) Medium-Term Prediction: time range from hours to a week; and (iii) Long-Term Prediction: time range from week to year or more.Short-term WPF allows the power industry to prepare for fluctuations in wind farm output. This preparation reduces operating expenses, the need for backup power, and reduce the strain on the electrical grid. The relevant methods mainly classified into physical, statistical, and hybrid methods [2].

The physical method [3] requires lots of data, because the establishment of a predictive model will be done with the help of many variables, and the amount of data has a direct relationship with forecast accuracy. As a result, the predictive model's calculation process and mathematical structure are complex, resulting in a relatively longer computation time. The most extensively utilized method is numerical weather prediction (NWP) [4]. The NWP involves extensive calculations and is better suited for Long-Term forecasting rather than Medium-Term and Short-Term forecasting. Statistical methods involve linear models and non-linear models. Autoregressive (AR), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) [5], vector autoregressive (VAR), Smooth-Transition Auto-Regressive (STAR) are some examples of linear

Please cite this article in press as:
M. Galphade, V. B. Nikam, B. Banerjee, A. W. Kiwelekar, P. Sharma. Stacked LSTM for Short-Term Wind Power Forecasting Using Multivariate Time Series Data, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 3, pp. 71-81, 2025, http://dx.doi.org/10.9781/ijimai.2024.07.002

model. ARIMA is unable to reliably predict wind power due to the accumulation of error hence many strategies have been developed to solve this limitation. The Non-linear model, includes strategies such as Autoregressive Conditional Heteroskedasticity (ARCH) [6], Generalized Autoregressive Conditional Heteroskedasticity Process (GARCH) and Deep learning algorithms [7], [8]. Artificial intelligence is a recent area of statistical methods, which is successfully applied for forecasting wind power, and the results of the forecasts gaining people approval [9]–[14]. Techniques used here involve artificial neural network (ANN) [15], support vector machine (SVM) [16], and Long Short-Term Memory (LSTM) [17]. Artificial intelligence is best suited for general purposes, but its disadvantage is that it cannot accurately explain the relationship between model elements. A study conducted by Kia Qu et al. investigated the use of the transformer architecture for short-term WPF [18]. This research covered the examination of wind power output data from various wind farms and included a comparative assessment with the LSTM model serving as a benchmark model. Where as a novel deep learning model that combines transformers and wavelet transforms uses weather data to predict wind speed and power generation six hours in advance, outperforming LSTM models [19]. Transformers offer several advantages for WPF, they come with computational requirements and may require large datasets for training. Additionally, model architecture, hyperparameter tuning, and data preprocessing are critical factors in achieving accurate forecasts.

A hybrid model [20] is a combination of different forecasting models. These combined models are expected to avoid the shortcomings of one model in forecasting wind power. Statistical methods are best in achieving high accuracy for short-term forecasts but tend to accumulate errors when used for long-term forecasts. On the other hand, physical methods, owing to their extensive scope and scale, are better suited for long-term forecasting rather than short-term predictions. Therefore, there are many errors in the existing methods, and studies are currently underway to improve the forecasting methods. Hybrid methods are generally classified into four types: hybrid approach based on data pre-processing technique, weight-based hybrid approach, hybrid approach based on data post-processing technique and parameter selection and optimization technique [21]. Table I provides a brief assessment of each class of hybrid techniques. This research takes first type of hybrid approach. Recent studies have suggested large amount of hybrid techniques in order to decrease the prediction errors by incorporating the merits of different methods. The author E. Lopez et al. utilized Principal Component Analysis (PCA) to reduce the input variables of the LSTM model for NWP data prediction [22]. The proposed model has demonstrated superior accuracy in comparison to SVM and Backpropagation Neural Network (BPNN). LSTM has higher prediction accuracy; therefore, it is used in wider range of applications. The author F. Shahid et al. merged wavenets with LSTM for diminishing wave and gradient transformation for nonlinear mapping [23]. WN-LSTM is applied on seven wind farms in Europe for Short-Term WPF. The performance is assessed based on metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results indicate a notable improvement, with performance showing a significant increase of 30%. In the Advanced LSTM [24] technique, the author fine-tuned the neural network's parameters based on new insights gained over time, instead of retraining the model using the entire dataset. The results of a case study in Belgium show that recalibration of advanced models will improve the consistency of predictions while reducing the cost of evaluating the system. Z. Sun et al. introduced a hybrid model that combines variational mode decomposition (VMD), Convolutional Long Short-Term Memory network (ConvLSTM), and error analysis [25]. The VMD approach divides the input wind energy into a set of different frequency components and then obtains a new structure that incorporates the convolutional layer into the LSTM network. This structure can extract the spatio-temporal features of each sub-series as an initial prediction engine.

Taking into account these issues, this study presents stacked LSTM for Short-Term Wind Power Forecasting using Multivariate Time Series Data. The major contributions of our paper are summarized as follows: (a) The presentation of a wind power forecasting model, employing the deep recurrent neural network LSTM, to assess potential improvements in predictions through the incorporation of additional training layers within the framework of time series analysis. (b) The proposed model uses data pre-processing technique and hyperparameter optimization to improve the LSTM network structure and proposed model is evaluated by comparing predicted values with actual values based on MAE & RMSE statistical error measures. (c) The performance of twelve deep LSTM models are assessed using various architectures for WPF. (d) The effectiveness of the presented research work is compared with three methods used in the literature. The subsequent sections of the paper are structured as follows: Section II describes overviews of LSTM. Section III provides an overview of the data pre-processing method. Section IV explains proposed model and performance metrics. Section V outlines experimental analysis on two open-access datasets. Finally, Section VI presents concluding remarks and future work.

II. Theoretical Framework

In this section LSTM is illustrated in detail. LSTM [26], a refined version of RNN, is designed for working with time series data. It's like a unique neural network that excels at making decisions based on data

TABLE I. A Brief Evaluation of the Hybrid Approaches

Hybrid WPF Approach	Advantages	Disadvantages
Hybrid approach based data pre-processing technique [27], [28]	• Higher performance compared to other approaches	• Mathematical knowledge of decomposition is required • slow response to new data
Weight-based hybrid approach [29]	• Adaptive to new data • Easy to implement • Adaptable to a variety of situations	• To determining the weights, an additional model is required • The best forecast within the forecast range is not guaranteed
Hybrid approach based on data post-processing technique [11]	• Effective in reducing systematic error • High accuracy	• Depends on designer's understanding of the optimization problems • Harder to code • Computationally complex
Hybrid approach based on parameter selection and optimization technique [30], [31]	• Relatively basic structure	• Computational time inefficient

over extended periods. Instead of RNN's conventional hidden layers, LSTM employs memory cells, as shown in the Fig. 1, to effectively manage and retain information.

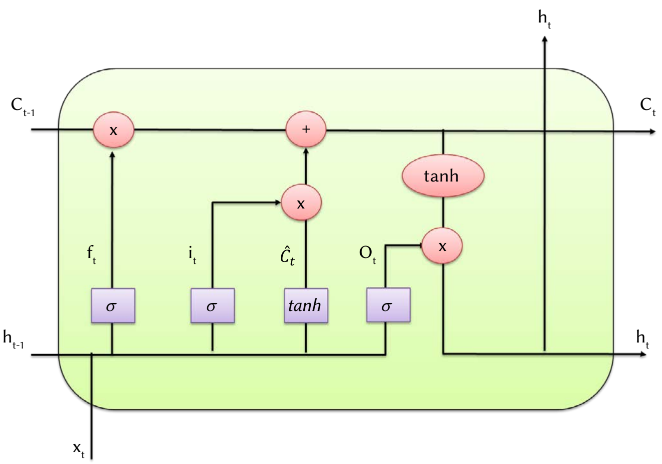


Fig. 1. Internal structure of LSTM cell.

LSTMs are experts at grasping temporal relationships by employing several gate units like the input gate (i_t), the forget gate (f_t), and the output gate (o_t), all accompanied by activation functions like tanh and sigmoid. By utilizing the sigmoid function each gate produces a binary output, either 1 or 0, which satisfies the input data flow. The input gate is a critical element responsible for identifying and selecting relevant information from the current input. It takes both the present input and the previous hidden state into account, using them to compute a candidate cell state. The decision related to the kind of information to be maintained and discarded is the responsibility of the forget gate. It achieves this by considering the current input and the previous hidden state, ultimately calculating a forget factor. The process of updating the cell state involves combining the previous cell state with the candidate cell state, with the assistance of the forget gate and input gate. This step ensures that the cell state is updated to integrate new information while preserving essential past information. The output gate plays a pivotal role in deciding which information from the updated cell state should be forwarded as the output. Additionally, it is instrumental in guiding subsequent cells in a sequence.

The mathematical computation is explained as follows:

Equation (1) compute the output vector h_t

$$\text{output: } h_t = o_t * \tanh(C_t) \tag{1}$$

where o_t is output gate vector, C_t is cell state, expressed using Equation (2) and Equation (3).

$$\text{OutputGate}: o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{2}$$

$$\text{CellUpdate}: C_t = f_t * C_{t-1} + i_t * \hat{C}_t)) \tag{3}$$

To adjust the memory cell, at each step the hyperbolic tangent (tanh) function is utilized so as to improve the training performance which is shown in Equation (4).

$$\text{ProcessInput}: \hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

The values f_t and i_t correspond to the forget gate and input gate, respectively. These values are computed using Equation (5) and Equation (6).

$$\text{ForgetGate}: f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

$$\text{InputGate}: i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{6}$$

Where W represents the weight matrix. Here subscript *o*, *c*, *i* and *f* denotes weight of each gate. LSTM networks have established themselves as exceptionally effective technique across an array of applications which includes sequential data and time series analysis. Their versatility is well-demonstrated in various domains. LSTM plays a crucial role in the domain of Natural Language Processing (NLP) which includes machine translation using sequence-to-sequence analysis and Google Neural Machine Translation (GNMT) models [32]. Moreover, LSTMs are also used for text generation tasks, for instance OpenAI's GPT-2 model to craft human-like text [33]. LSTM (Long Short-Term Memory) networks are versatile in time series forecasting, proving effective in applications such as predicting stock prices [34], [35], oil production [36], weather forecasting [37], and Automatic Speech Recognition systems [38] ,Sea Surface Temperature Prediction [39], photovoltaic Generation [40] by analyzing historical data. They proficiently translate spoken language into text, enabling voice-controlled personal assistants like Siri, Google Assistant, and Amazon Alexa, accurately. In Healthcare, LSTMs are crucial for analyzing and decision making in medical time series data such as electrocardiogram (ECG) and electroencephalogram (EEG) [41]. In autonomous driving, LSTM networks are effectively employed for tasks like vehicle trajectory prediction and object detection [42]. Human Activity Recognition (HAR) utilizes LSTM networks in order to recognize and classify human activities emerging from sensor data located in wearable devices [43]. These examples collectively highlight the adaptability of LSTM networks in handling sequential data across a diverse range of domains.

III. Data Pre-Processing

Time series information is generally obtained from real world environment or data generated from sensors. These data are usually affected by instances like noise. A proper method for collecting data and the removal of any deficient information should be followed before performing any kind of analysis. This is crucial as it will assist in making the analysis less challenging and to avoid incorrect analytical conclusions. Typically, data obtained from wind farms contains a lot of missing values due to sensors malfunctioning. Detecting outliers, eliminating noise, and filling in missing values are the three most common data pre-processing techniques.

- **Detecting outliers**: Statistical-based outlier detection methods leverage statistical distributions and the relationship of data points to these distributions. Parametric methods [44], [45] rely on predefined models, while non-parametric methods take a more flexible and data-driven approach to identify outliers. Whereas density-based [46] outlier detection methods are founded on a fundamental principle: outliers tend to reside in regions of low data density. Conversely, non-outliers, often referred to as inliners or genuine data points, are expected to cluster in dense neighborhoods. These methods assess the densities of data points within their local contexts, comparing them to the densities of their nearby neighbors. Distance-based methods identify outliers by calculating distances between data points. Outliers are often defined as data points that are significantly distant from their nearest neighbors.
- **Eliminating noise**: One of the dominant challenges encountered in wind farm datasets is the presence of noisy data. This issue can be effectively addressed by employing a filter method to extract the accurate signal estimation. Among the frequently employed data filtering techniques, the most common strategies include frequency domain, time-domain and time-frequency domain filtering.
- **Missing values**: Missing data are common in statistical analyses. The imputation function is widely used to identify the missing values in a wind farm based on the variables and their relationship. Do not use metaphorical expressions.

The data was normalized before using it in the forecasting system. A dataset can be normalized using a variety of methods. The method used in this study is known as min-max normalization.

A. Sliding Window

In a multivariate time series, each variable represents an aspect of a complex system, and the temporal evolution of these variables signifies changes in the underlying system state. To capture the interrelationships and temporal dependencies among these variables and derive the system's state over specific periods, we employed a sliding window approach with a size of K to segment the multivariate time series data. This partitioning technique allows us to analyze the data within discrete intervals, facilitating the extraction of meaningful insights about the system's behavior. As illustrated in Fig. 2, for a given point in time t, we can establish the sliding window data $W_t \in R^{MXK}$, where $W_t = \{x_{t-K}, ..., x_{t-1}, x_t\}$, M is the dimension of the variable and K represent the sliding window size.

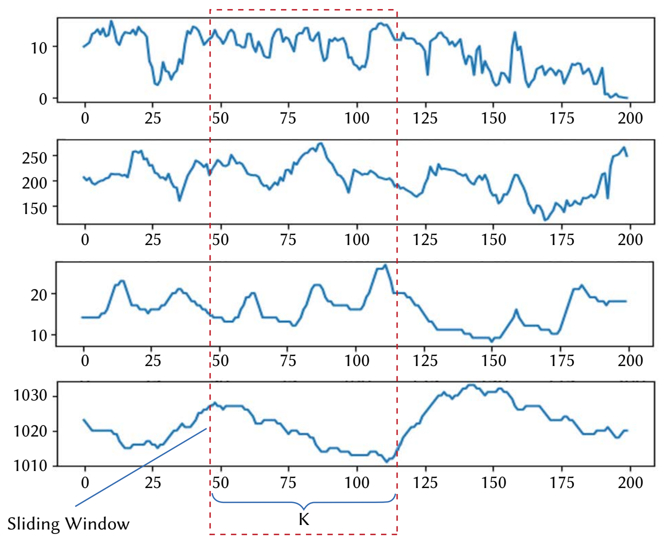


Fig. 2. Sliding Window.

B. RFR Based Imputation Method

Random forest [47] imputation is a machine learning technique that offers the advantage of handling nonlinearities and interactions within data without necessitating the specification of a specific regression model. These methods do not assume normality or require specification of parametric models. In this paper random forest method has been used for filling missing value. It extends from classification and regression trees (CART) [48], which are predictive models that iteratively partition the data based on predictor variable values. Notably, random forests don't depend on distributional assumptions and can effectively manage nonlinear relationships and interactions in the data. The schematic of the Random Forest Imputation framework is shown in Fig. 3 The steps involved in the proposed imputation approach is explained in Algorithm 1:

Algorithm 1: Random Forest Imputation Algorithm

Data: Data set $\mathcal{D}$ containing $|\mathcal{R}|$ records and $|\mathcal{A}|$ attributes.

Result: An imputed dataset $\mathcal{D}_{Final}$ containing $|\mathcal{R}|$ records and $|\mathcal{A}|$ attributes.

for *each record* $R_i \in \mathcal{D}$ **do**
  if *any attribute* A_k *is missing* **then**
    $\mathcal{D}_{miss} = \mathcal{D}_{miss} \cup R_i$;
  end
  else
    $\mathcal{D}_{comp} = \mathcal{D}_{comp} \cup R_i$;
  end
end
rf_{model} = TrainRandomForest($\mathcal{D}_{comp}$);
while $\mathcal{D}_{miss} \neq \emptyset$ **do**
  $R_i \leftarrow$ Select an incomplete record from $\mathcal{D}_{miss}$;
  for $j = 1, ..., M$ **do**
    if $r_{ij} = NaN$ **then**
      r_{ij} = PredictRFR(rf_{model}, R_i);
      $\mathcal{D}_{compute} = \mathcal{D}_{compute} \cup A_j$;
    end
  end
end
$\mathcal{D}_{Final} = \mathcal{D}_{comp} \cup \mathcal{D}_{compute}$;
return $\mathcal{D}_{Final}$

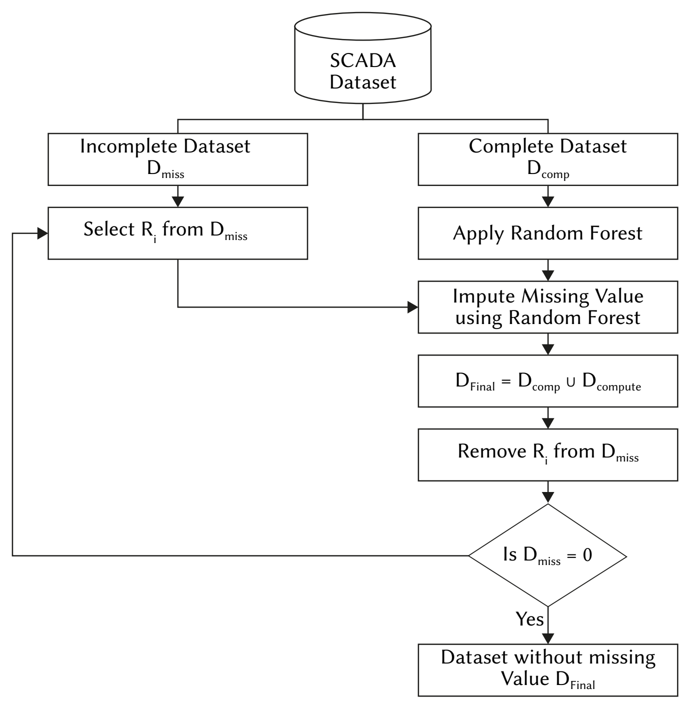


Fig. 3. RFR Imputation framework.

The effectiveness of the imputation method is evaluated by using RMSE and MAE parameters.

TABLE II. Details of Dataset Used in the Experiment

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
Yalova Wind Farm	Active power, Theoretical power, Wind speed, Wind direction	10 min period 1 January 2018 to 31 December 2018	54000kW	50530 records with 4 attribute

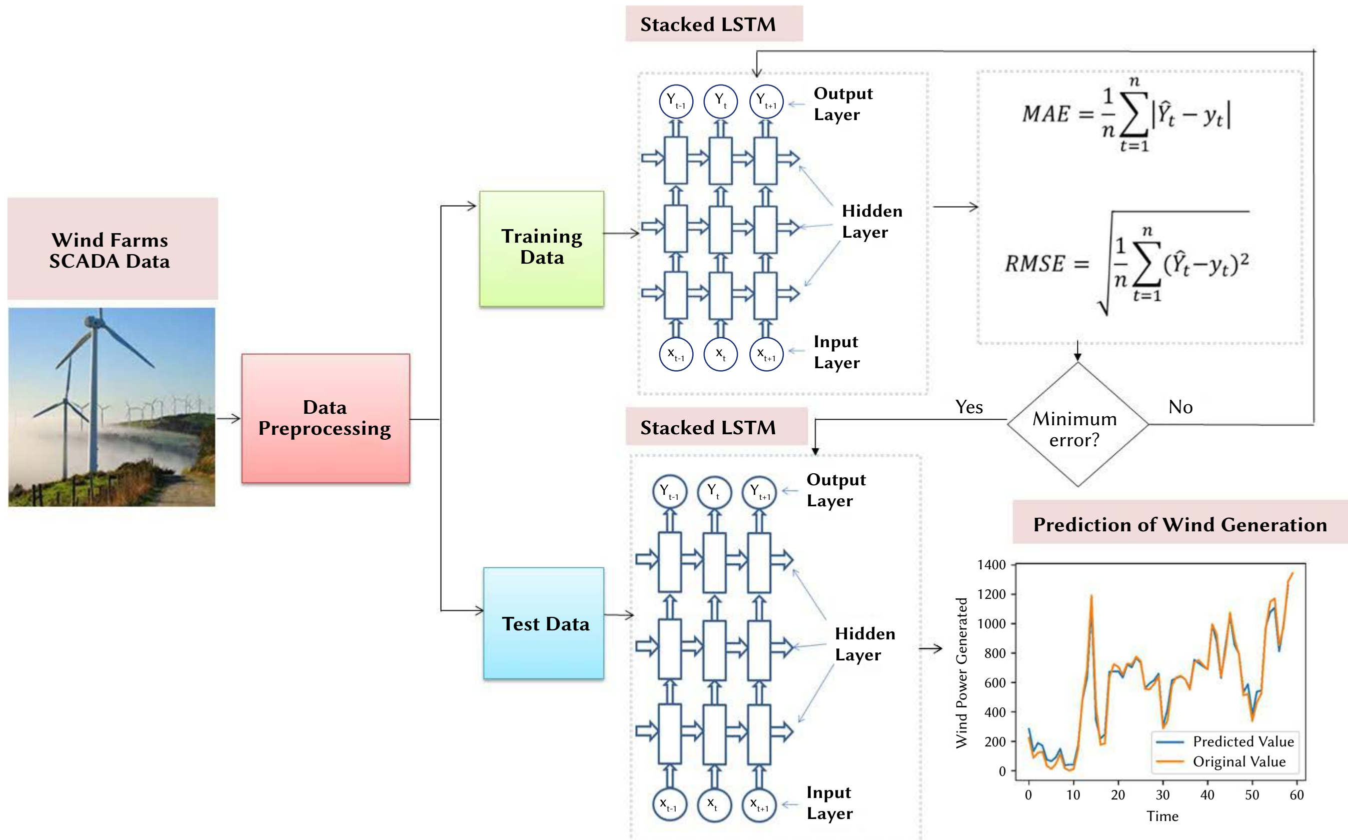


Fig. 4. Proposed stacked LSTM architecture.

C. Dataset

The characteristics of the two datasets include data instances, attributes, capacity, resolution, and record count which are detailed in Table II.

1. Sotavento Wind Farm

This dataset is from the Sotavento wind farm in Galicia, Spain with a total capacity of 17560 kW which consists of 24 onshore turbines. This wind farm gives real-time information on wind direction, wind speed and turbine's power generation. This paper uses five years data from 2014-2019 to test the proposed forecast model. The resolution of the data is hourly. Some attribute of this dataset has missing value. Wind power and wind speed is having 756 and 500 missing values respectively. Pre-processing strategies with deep learning methods are combined to evaluating the effectiveness of pre-processing. Two approaches have been used while performing experimentation, first is simply removing missing value and second approach is to use RFR imputer.

2. Yalova Wind Farm

The Yalova Wind Farm (YWF) situated in western Turkey records wind related information such as wind speed, wind direction, turbine power (theoretical) and generated. YWF possess a total capacity of 54,000 kW, obtained from 36 wind turbines. The wind generated information is collected and reserved utilizing a Supervisory Control and Data Acquisition (SCADA) system. Here the data is recorded at an interval of 10 minutes. The format of the data available is CSV.

IV. Proposed Model

Fig. 4 shows the proposed architecture. For the experimentation purpose two datasets are used, first is from Sotavento wind farm in Galicia and second from Yalova Wind in west Turkey. Data pre-processing is done on both the dataset as described in Section III. The dataset is partitioned into training and testing samples by employing 70:30 ratio. The model is trained on train set using stacked LSTM. In order to improve the accuracy of WPF, various network structures, along with different numbers of layers and neurons in each layer, are tested. Min-Max scaling function is employed to scale the input features before applying to the deep learning models. Min-Max scalar is expressed by Equation (7) as,

$$X_{\text{scaled}} = \frac{(X_i - \min(X))}{(\max(X) - \min(X))} \tag{7}$$

A. Stacked LSTM

A stacked LSTM is a neural network architecture that comprises multiple LSTM layers, creating a multi-layer structure, as illustrated in the Fig. 5. Incorporating multiple LSTM layers increases the model's complexity and depth. In this composite LSTM structure, each output layer of the LSTM model serves as input to subsequent layers within the same block. This design allows the model to capture the temporal patterns in the data and combine the learned representations from previous layers, resulting in a higher-level abstraction in the final output. Each intermediate LSTM layer produces a sequence vector as its output, which serves as the input for the next LSTM layer. Unlike a single-output LSTM, the stacked LSTM provides an output for each timestamp. Equations (8) to Equations (13) illustrate N[th] layer of unrolled stacked LSTM.

$$\text{ForgetGate}: f_t^N = \sigma(W_f^N \cdot [h_{t-1}^N, h_t^{N-}] + b_f^N) \tag{8}$$

$$\text{InputGate}: i_t^N = \sigma(W_i^N \cdot [h_{t-1}^N, h_t^{N-1}] + b_i^N) \tag{9}$$

$$\text{ProcessInput:}: \hat{C}_t^N = \tanh(W_c^N \cdot [h_{t-1}^N, h_t^{N-1}] + b_c^N) \tag{10}$$

$$\text{CellUpdate}: C_t^N = f_t^N \star C_{t-1}^N + i_t^N \star \hat{C}_t^N \tag{11}$$

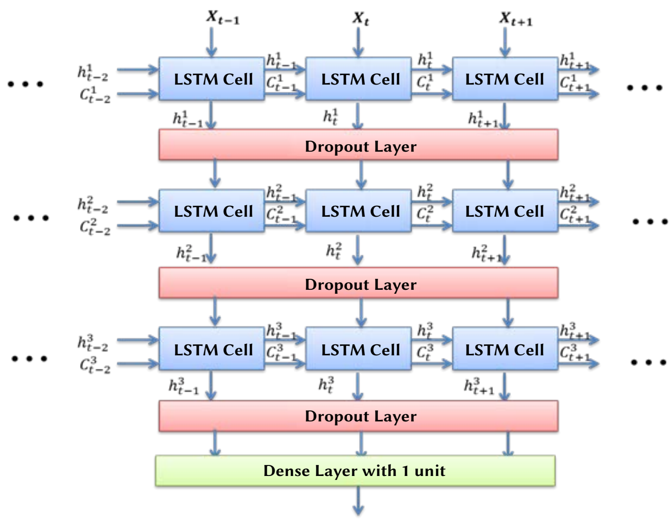


Fig. 5. Detail three-layer LSTM architecture.

$$\text{OutputGate}: o_t^N = \sigma(W_o^N \cdot [h_{t-1}^N, h_t^{N-1}] + b_o^N) \tag{12}$$

$$\text{Output}: h_t^N = o_t^N \star \tanh(C_t^N) \tag{13}$$

A stacked LSTM neural network outperform a regular LSTM due to its increased capacity and ability to capture more complex patterns in the data. Here are some reasons why a three-layer LSTM could outperform a regular LSTM:

- **Hierarchical Feature Extraction**: A stacked LSTM network has an additional layer to extract hierarchical features from the data. This allows it to learn abstract representations of the input data at different levels of granularity. The first layer can capture low-level features, the second layer can capture mid-level features, and the third layer can capture high-level features. This hierarchical feature extraction can make the network more effective at capturing complex patterns.
- **Increased Capacity**: The additional layer in a stacked LSTM increases the network's capacity to learn and represent data. It can model more intricate relationships and dependencies within the data, which is particularly useful for tasks involving long sequences or complex temporal dependencies.
- **Improved Generalization**: A deeper network, such as a three-layer LSTM, can sometimes generalize better to unseen data. While a deeper network has the potential to overfit the training data, appropriate regularization techniques (e.g., dropout) can help mitigate this risk. With proper regularization, a three-layer LSTM can learn to generalize well to new, unseen examples.
- **Better Representation Learning**: The stacked architecture allows for more sophisticated representation learning. Each layer can transform the input data into a more meaningful representation. This can be especially advantageous for tasks where the input data is complex or contains multiple levels of abstraction.

B. Cross-Validation

In this experiment we used rolling cross-validation method. It begin with a small initial subset of the data for training. Then, make forecasts for the subsequent data points and assess the accuracy of these forecasts. Importantly, the data points that were forecasted are subsequently incorporated into the training dataset for the next iteration. This process is repeated, allowing for the inclusion of previously forecasted data points in each subsequent training dataset, while forecasting the remaining data points. Rolling cross-validation provides a robust means of evaluating the performance of time-series models by continuously expanding the training dataset while testing against new, unseen data points. The experiment is performed on 5-fold-cross-validation as shown in Fig. 6.

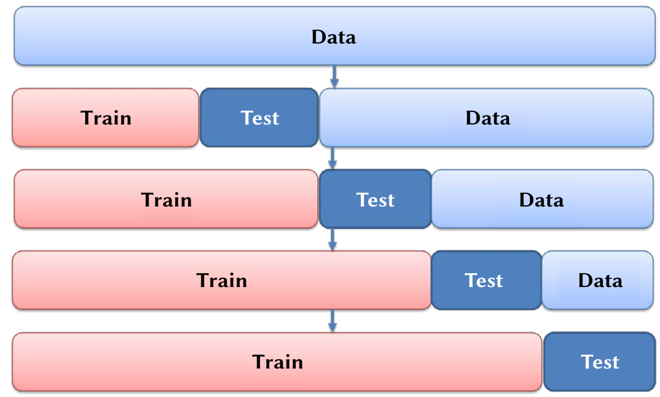


Fig. 6. Rolling cross-validation.

C. Evaluation Criteria

MAE and RMSE are selected for comprehensive and quantitative evaluation of prediction performance. RMSE and MAE represent absolute time series error and the scale is same as the data. A decrease in MAE and RMSE values signifies a reduction in model error residuals, indicating a higher level of accuracy in the prediction model. MAE and RMSE are expressed by Equation (14) and Equation (15) as,

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |\hat{Y}_t - y_t| \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (\hat{Y}_t - y_t)^2} \tag{15}$$

where n is the total number of the predicted values $\hat{Y}_t$ and y_t is the actual value.

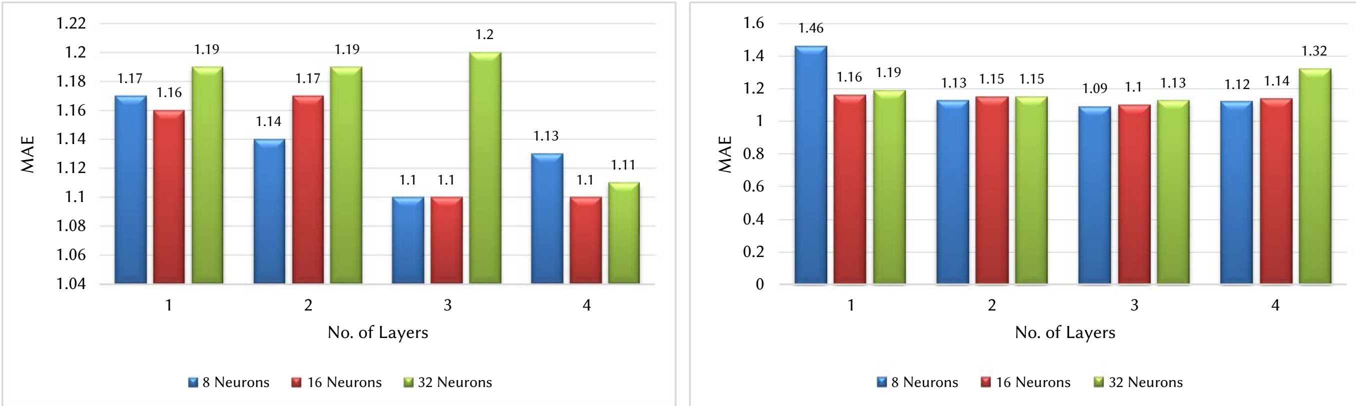
V. Experimental Analysis

In this section, results of proposed stacked LSTM on Sotavento and Yalova wind farm dataset are discussed. The proposed model is implemented using Python 3.10 employing Keras library. The loss function for LSTM was defined as the mean squared error, and the optimization method used was 'ADAM'. The dataset is partitioned into training and testing phases. Here the presented model follows training and testing ratio of 70:30. The performance of the forecasting models is assessed using RMSE and MAE.

A. Sotavento Wind Farm

In order to construct a stacked LSTM, trial and error method is used to select the size of the input layer and output layers. For that, experiments have been conducted on LSTM networks with 1 to 4 hidden layers, and the best LSTM network with the smallest MAE and RMSE has been selected.

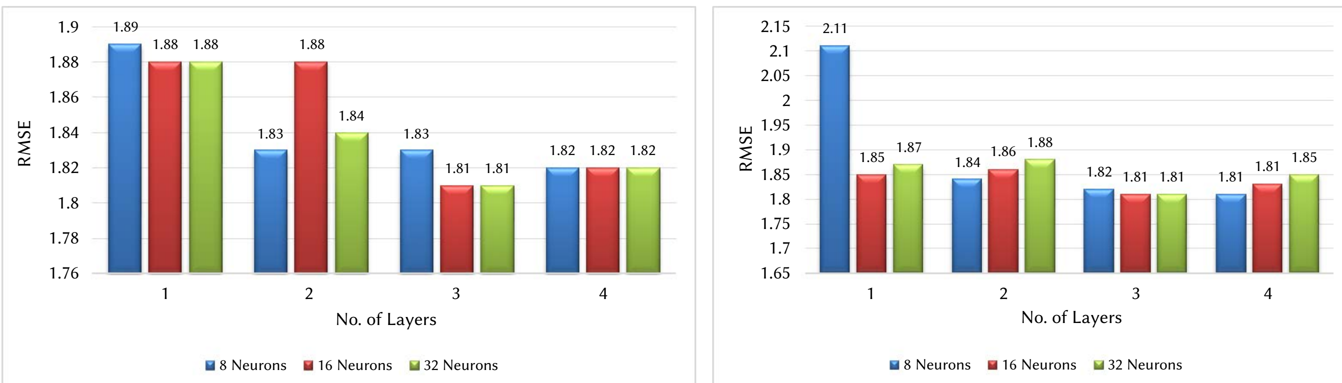
The model with the structure "16-16-16-1" works best. The proposed stacked LSTM shows a comparatively lesser MAE and RMSE values as depicted in Table III. When fourth layer is added, MAE and RMSE value start increasing because of number of parameters. It is also observed that when the missing values are imputed with the help of RFR, the performance is improved in every case as shown in Fig. 7b and Fig. 8b. Only for single layer it has slightly increases.



(a) MAE value after removing NA

(b) MAE value after imputing missing value using RFR

Fig. 7. MAE value of Sotavento wind farm dataset.



(a) RMSE value after removing NA

(b) RMSE value after imputing missing value using RFR

Fig. 8. RMSE value of Sotavento wind farm dataset.

TABLE III. MAE and RMSE Value of Sotavento Wind Farm

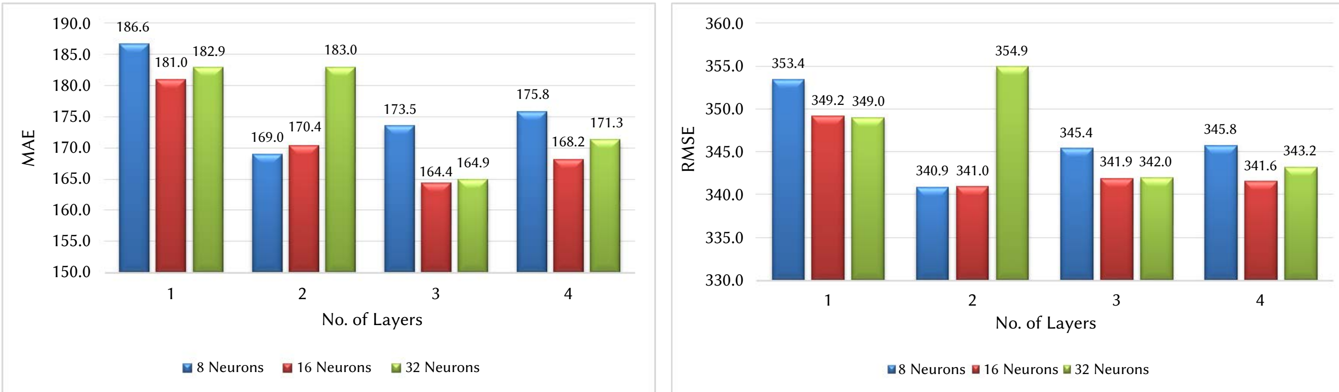
No. of Layers	No. of Neurons	Removing NA records		Missing value imputation using RFR	
		MAE	RMSE	MAE	RMSE
1	8	1.17	1.89	1.46	2.11
2		1.14	1.83	1.13	1.84
3		1.1	1.83	1.09	1.82
4		1.13	1.82	1.12	1.81
1	16	1.16	1.88	1.1	1.85
2		1.17	1.88	1.15	1.86
3		1.1	1.81	1.1	1.81
4		1.1	1.82	1.14	1.83
1	32	1.19	1.88	1.19	1.87
2		1.19	1.84	1.15	1.88
3		1.2	1.81	1.13	1.81
4		1.11	1.82	1.32	1.85

B. Yalova Wind Farm

Yalova Wind Farm dataset does not contain any missing value or outliers. The dataset is a clean dataset so it is directly divided into train and test set. From the experimental results shown in Table IV it is observed that the prediction performance decreases along with increase of amount of layers. The MAE and RMSE values for each model are plotted in Fig. 9. The graph shows that architecture with 3 layers and 16 neurons provides the smallest error value compared to other architectures. The minimum MAE of this architecture is 164.35, and the RMSE is 341.92. Therefore, the structure of "16-16-16-1" is selected as the optimal model.

TABLE IV. MAE and RMSE Value of Sotavento Wind Farm

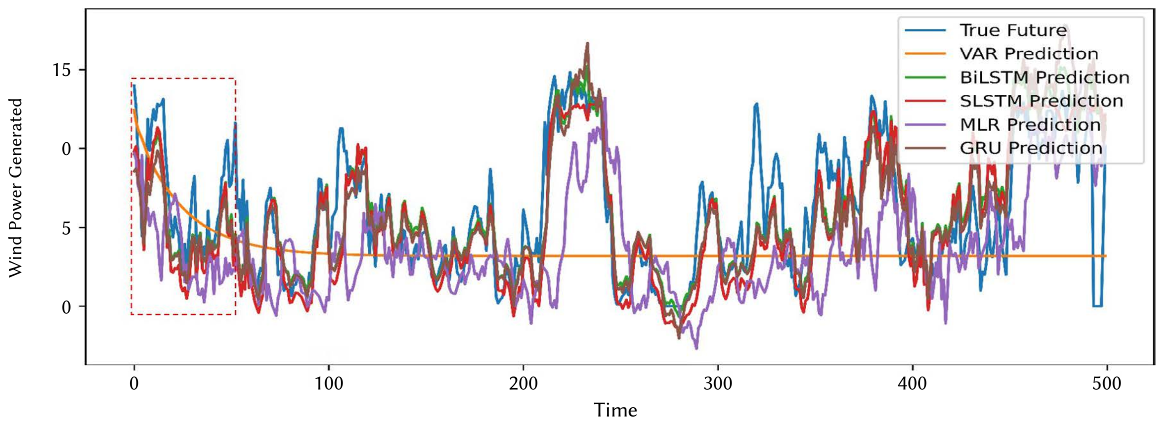
No. of Layers	No. of Neurons	Removing NA records	
		MAE	RMSE
1	8	186.63	353.44
2		168.97	340.87
3		173.54	345
4		175.82	345.77
1	16	180.96	349.15
2		170.38	340.99
3		164.35	341.92
4		168.15	341.57
1	32	182.88	349
2		182.98	354.92
3		164.89	341.97
4		171.34	343.21



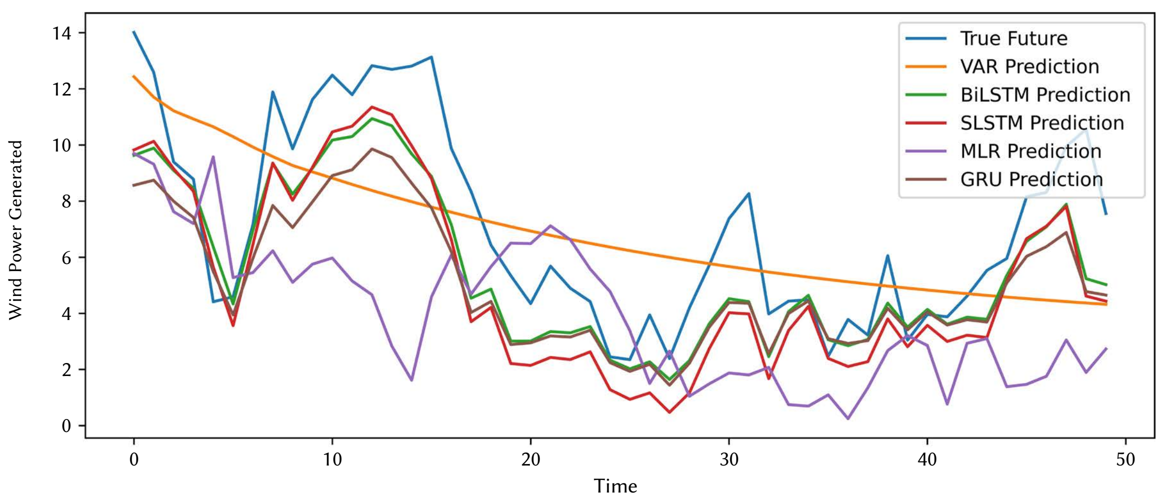
(a) MAE value

(b) RMSE value

Fig. 9. MAE and RMSE value of YalovaWind Farm dataset.



(a) Actual vs Predicted values for Sotavento wind farm



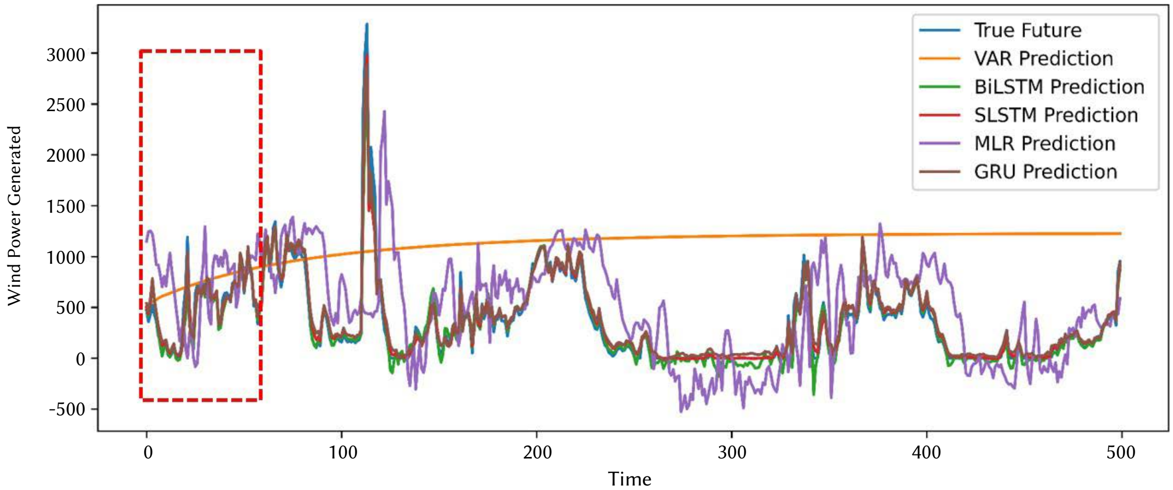
(b) Detailed view of clipped section for Sotavento wind farm

Fig. 10. Comparison of experimental results on Sotavento wind farm dataset.

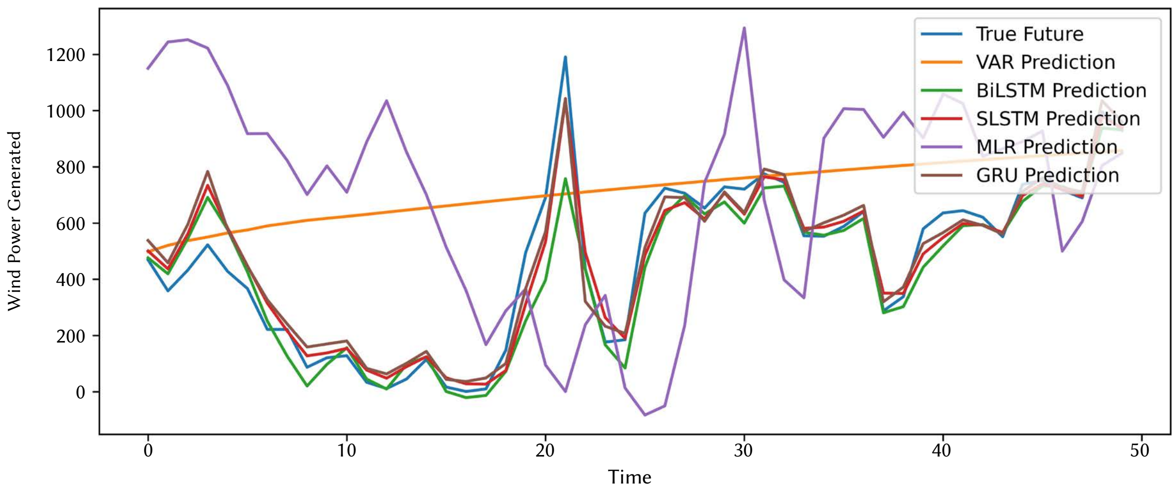
C. Comparative Analysis

To verify the proposed models accuracy, four models including Multiple linear regression, VAR, GRU, BiLSTM are employed to conduct WPF of two wind farms. Table V shows the comparison results between proposed models and benchmark models. For regression model, the input is a multivariate time series of length ten (corresponding to 10 hours) and output is a real number corresponding to the next hour into the future. An input and hidden layers contain 16 neurons. Where as an output layer contains one neuron. The total number of epochs is 100, batch size is 64, and training rate is 0.001.

On the Yalova wind farm test dataset, the VAR model produces significantly higher MAE (995.09 and 1006.24) compared to the 3-layer stacked LSTM. This difference can be attributed to the VAR model's limitation in handling stationary time series data effectively. When we utilize the GRU model, the MAE increases by 0.07 and 25.04, and the RMSE increases by 0.05 and 7.28 for the Sotavento wind farm and Yalova Wind Farm, respectively, in comparison to the three-layer Stacked LSTM. The stacked LSTM model has much better performance than the baseline model, since it can capture longer temporal dependencies. Fig. 10a and Fig. 11a present graphical representations illustrating the



(a) Actual vs Predicted values for Sotavento wind farm



(b) Detailed view of clipped section for Sotavento wind farm

Fig. 11. Comparison of experimental results on Sotavento wind farm dataset.

predictions achieved by the five regressors for Sotavento Wind Farm and Yalova Wind Farm, respectively. The test result shows that, the proposed method is more robust and effective than the other five WPF methods. The fundamental reason for this is because the proposed method is based on the LSTM, which, due to the presence of self-feedback connections, is suited and successful in modelling time-series data. Subsequently, Fig. 10b and Fig. 11b provide a detailed view of the experiment. The detailed view shows the clarity of the improvements in measurements.

TABLE V. Comparative Analysis of RFR-Based Stacked LSTM

No.of Layers	Imputing Missing values using RFR		Yalova Wind Farm	
	MAE	RMSE	MAE	RMSE
Multiple Linear Regression	1.36	1.98	404.56	521.71
VAR	2.9	3.61	1159.44	1348.16
GRU	1.17	1.86	189.39	349.2
BiLSTM	1.75	2.28	180.44	342.94
Stacked LSTM	1.1	1.81	164.35	341.92

VI. Conclusion

Wind power is an essential area in renewable energy generation. This paper talks about the WPF utilizing a time series data acquired from wind mill. Here, the wind power is predicted effectively by employing the proposed stacked LSTM model. At first, utilizing two approaches such as removing missing values and RFR the input information is pre-processed. It is observed that the accuracy of the proposed model significantly increases when RFR is used for data imputation. It is followed by a 3-layer LSTM model referred as stacked LSTM to perform the task of WPF. From experimentation, it is observed that the proposed model achieved an average accuracy of 94.01% which is greater than the state-of-the-art approaches. The performance of the proposed model is evaluated using two error metrics such as MAE and RMSE. In addition, it is also observed that addition of a fourth layer slightly decreases the accuracy due to increased complexity. It is discovered from the analysis that the proposed model consistently outperformed other methods in terms of error. The proposed stacked LSTM model which is built on conventional LSTM, effectively captures the complex hidden patterns and changes in wind power output with respect to time due to its memory units and recurrent design. However, time required for training the data is comparatively high. Also, the performance of the proposed stacked LSTM requires large amount of data. In addition, finding an optimal hyperparameters is challenging and will require extensive experimentation.

This model can further be improved using a deep BiLSTM network. BiLSTMs are computationally more intensive than unidirectional LSTMs because they process the data in both directions. Furthermore, pre-processing methods which depict time-frequency analysis can be explored. Further research into the model's scalability, transferability to other contexts, and incorporation of additional relevant improvements could help in maximizing its practical and real time applications.

Acknowledgment

The authors wish to extend their appreciation to the Faculty Development Centre (VJTI-DBATU) focused on Geoinformatics, Spatial Computing, and Big Data Analytics. This center, established under the auspices of PMMMNMTT, MHRD, Government of India, New Delhi, has been involved in enhancing capacity in various fields, including Geoinformatics, AI & Machine Learning, Deep Learning, Big Data Analytics, and related domains. Their support has been invaluable in advancing our research in this study.

References

[1] S. K. Kim and S. Park, "Impacts of renewable energy on climate vulnerability: A global perspective for energy transition in a climate adaptation framework," *Science of The Total Environment*, vol. 859, pp. 160175, Feb. 2023, doi: 10.1016/J.SCITOTENV.2022.160175.

[2] S. M. H. D. Perera, G. Putrus, M. Conlon, M. Narayana, and K. Sunderland, "Wind Energy Harvesting and Conversion Systems: A Technical Review," *Energies*, vol. 15, no. 24, p. 1-34, 2022.

[3] T. M. Giannaros, D. Melas, I. Ziomas, "Performance evaluation of the Weather Research and Forecasting (WRF) model for assessing wind resource in Greece," *Renewable Energy*, vol. 102, pp. 190–198, 2017.

[4] C. Liu, X. Zhang, S. Mei, Z. Zhen, M. Jia, Z. Li, H. Tang, "Numerical weather prediction enhanced wind power forecasting: Rank ensemble and probabilistic fluctuation awareness," *Applied Energy*, vol. 313, p. 118769, 2022.

[5] S. Kumari, S. Sreekumar, S. Singh, D. P. Kothari, "Comparison Among ARIMA, ANN, and SVR Models for Wind Power Deviation Charge Reduction," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, vol. 1, 2022, pp. 551–557, IEEE.

[6] F. Yao, W. Liu, X. Zhao, L. Song, "Integrated Machine Learning and Enhanced Statistical Approach-Based Wind Power Forecasting in Australian Tasmania Wind Farm," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/9250937.

[7] Z. Wu, G. Luo, Z. Yang, Y. Guo, K. Li, Y. Xue, "A comprehensive review on deep learning approaches in wind forecasting applications," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 2, pp. 129–143, 2022.

[8] Y. Wang, R. Zou, F. Liu, L. Zhang, Q. Liu, "A review of wind speed and wind power forecasting with deep neural networks," *Applied Energy*, vol. 304, p. 117766, 2021.

[9] H. Zhen, D. Niu, M. Yu, K. Wang, Y. Liang, and X. Xu, "A Hybrid Deep Learning Model and Comparison for Wind Power Forecasting Considering Temporal-Spatial Feature Extraction," *Sustainability*, vol. 12, no. 22, pp. 9490, May 2020, doi: 10.3390/SU12229490.

[10] K.-S. Chen, K.-P. Lin, J.-X. Yan, and W.-L. Hsieh, "Renewable Power Output Forecasting Using Least-Squares Support Vector Regression and Google Data," Sustainability, vol. 11, no. 11, pp. *3009*, May 2019, doi: 10.3390/SU11113009.

[11] J. Zhou, X. Yu, and B. Jin, "Short-Term Wind Power Forecasting: A New Hybrid Model Combined Extreme-Point Symmetric Mode Decomposition, Extreme Learning Machine and Particle Swarm Optimization," *Sustainability*, vol. 10, no.9, pp. 3202, May 2018, doi: 10.3390/SU10093202.

[12] B. Xiong, L. Lou, X. Meng, X. Wang, H. Ma, Z. Wang, "Short-term wind power forecasting based on Attention Mechanism and Deep Learning," *Electric Power Systems Research*, vol. 206, 2022.

[13] L. Ye, B. Dai, M. Pei, P. Lu, J. Zhao, M. Chen, B. Wang, "Combined approach for short-term wind power forecasting based on wave division and Seq2Seq model using deep learning," *IEEE Transactions on Industry Applications*, vol. 58, no. 2, pp. 2586–2596, 2022.

[14] A. Alkesaiberi, F. Harrou, Y. Sun, "Efficient wind power prediction using machine learning methods: A comparative study," *Energies*, vol. 15, no. 7, p. 2327, 2022.

[15] J. Jamii, M. Mansouri, M. Trabelsi, M. F. Mimouni, W. Shatanawi, "Effective artificial neural network- based wind power generation and load demand forecasting for optimum energy management," *Frontiers in Energy Research*, vol. 10, 2022.

[16] D.-D. Yuan, M. Li, H.-Y. Li, C.-J. Lin, B.-X. Ji, "Wind power prediction method: Support vector regression optimized by improved jellyfish search algorithm," *Energies*, vol. 15, no. 17, p. 6404, 2022.

[17] X. Shi, X. Lei, Q. Huang, S. Huang, K. Ren, Y. Hu, "Hourly Day-Ahead Wind Power Prediction Using the Hybrid Model of Variational Model Decomposition and Long Short-Term Memory," *Energies 2018, Vol. 11, Page 3227*, vol. 11, p. 3227, may 2018, doi: 10.3390/EN11113227.

[18] K. Qu, G. Si, Z. Shan, X. Kong, X. Yang, "Short- term forecasting for multiple wind farms based on transformer model," *Energy Reports*, vol. 8, pp. 483–490, 2022.

[19] E. G. S. Nascimento, T. A. de Melo, D. M. Moreira, "A transformer-based deep neural network with wavelet transform for forecasting wind speed and wind energy," *Energy*, vol. 278, pp. 127–678, 2023.

[20] A. Lagos, J. E. Caicedo, G. Coria, A. R. Quete, M. Martz'zinez, G. Suvire, J. Riquelme, "State-of-the- Art using bibliometric analysis of Wind-Speed and- Power forecasting methods applied in power systems," *Energies*, vol. 15, no. 18, p. 6545, 2022.

[21] A. Tascikaraoglu, M. Uzunoglu, "A review of combined approaches for prediction of short-term wind speed and power," 2014, doi: 10.1016/j.rser.2014.03.033.

[22] E. López, C. Valle, H. Allende, E. Gil, H. Madsen, "Wind Power Forecasting Based on Echo State Networks and Long Short-Term Memory," *Energies 2018, Vol. 11, Page 526*, vol. 11, p. 526, may 2018, doi: 10.3390/EN11030526.

[23] F. Shahid, A. Zameer, A. Mehmood, M. A. Z. Raja, "A novel wavenets long short term memory paradigm for wind power prediction," *Applied Energy*, vol. 269, pp. 115098, may 2020, doi: 10.1016/J.APENERGY.2020.115098.

[24] J. F. Toubeau, P. D. Dapoz, J. Bottieau, A. Wautier, Z. D. Grève, F. Vallée, "Recalibration of recurrent neural networks for short-term wind power forecasting," *Electric Power Systems Research*, vol. 190, pp. 106639, may 2021, doi: 10.1016/J.EPSR.2020.106639.

[25] Z. Sun, M. Zhao, "Short-Term Wind Power Forecasting Based on VMD Decomposition, ConvLSTM Networks and Error Analysis," *IEEE Access*, vol. 8, pp. 134422– 134434, 2020, doi: 10.1109/ACCESS.2020.3011060.

[26] S. Siami-Namini, N. Tavakoli, A. S. Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pp. 1394–1401, may 2019, doi: 10.1109/ICMLA.2018.00227.

[27] H. S. Dhiman, P. Anand, D. Deb, "Wavelet transform and variants of SVR with application in wind forecasting," *Advances in Intelligent Systems and Computing*, vol. 757, pp. 501–511, 2019, doi: 10.1007/978-981-13-1966-245.

[28] Y. Zhang, J. Le, X. Liao, F. Zheng, Y. Li, "A novel combination forecasting model for wind power integrating least square support vector machine, deep belief network, singular spectrum analysis and locality- sensitive hashing," *Energy*, vol. 168, pp. 558–572, may 2019, doi: 10.1016/J.ENERGY.2018.11.128.

[29] S. Han, Y. Liu, J. Li, "Wind power combination prediction based on the maximum information entropy principle," 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6321153.

[30] M. I. A. A. Khalaf, J. Q. Gan, "Deep classifier structures with autoencoder for higher-level feature extraction," *IJCCI 2018 - Proceedings of the 10th International Joint Conference on Computational Intelligence*, pp. 31–38, 2018, doi: 10.5220/0006883000310038.

[31] C. Fu, G.-Q. Li, K.-P. Lin, and H.-J. Zhang, "Short-Term Wind Power Prediction Based on Improved Chicken Algorithm Optimization Support Vector Machine," Sustainability, vol. 11, pp. 512, May 2019, doi: 10.3390/SU11020512.

[32] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine

translation," *arXiv preprint arXiv:1609.08144*, 2016.

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[34] J. Shah, D. Vaidya, and M. Shah, "A comprehensive review on multiple hybrid deep learning approaches for stock prediction," *Intelligent Systems with Applications*, vol. 16, 2022, https://doi.org/10.1016/j.iswa.2022.200111.

[35] R. K. Behera, S. Das, S. K. Rath, S. Misra, R. Damasevicius, "Comparative study of real time machine learning models for stock prediction through streaming data.," *The Journal of Universal Computer Science*, vol. 26, no. 9, pp. 1128–1147, 2020.

[36] A. M. AlRassas, M. A. Al-qaness, A. A. Ewees, S. Ren, M. Abd Elaziz, R. Damaševičius, T. Krilavičius, "Optimized anfis model using aquila optimizer for oil production forecasting," *Processes*, vol. 9, no. 7, p. 1194, 2021.

[37] B. Y. El-Habil, S. S. Abu-Naser, "Global climate prediction using deep learning," *Journal of Theoretical and Applied Information Technology, vol. 100, no. 24*, 2022.

[38] J. Oruh, S. Viriri, A. Adegun, "Long short-term memory recurrent neural network for automatic speech recognition," *IEEE Access*, vol. 10, pp. 30069– 30079, 2022.

[39] D. Chen, J. Wen, C. Lv, "A spatio-temporal attention graph convolutional networks for sea surface temperature prediction," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, pp. 64–72, 2023, doi: https://doi.org/10.9781/ijimai.2023.02.011.

[40] M. Martínez-Comesaña, J. Martínez-Torres, P. Eguía- Oller, J. López-Gómez, "Use of optimised lstm neural networks pre-trained with synthetic data to estimate pv generation," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, pp. 1–10, 2023, doi: https://doi.org/10.9781/ijimai.2023.11.002.

[41] M. Roy, S. Majumder, A. Halder, U. Biswas, "Ecg-net: A deep lstm autoencoder for detecting anomalous ecg," *Engineering Applications of Artificial Intelligence*, vol. 124, 2023.

[42] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[43] F. J. Ordóñez, D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, pp. 115, 2016.

[44] H. N. Akouemo, R. J. Povinelli, "Data improving in time series using ARX and ANN models," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3352– 3359, 2017.

[45] F. Harlé, F. Chatelain, C. Gouy-Pailler, S. Achard, "Bayesian model for multiple change-points detection in multivariate time series," *IEEE Transactions on Signal Processing*, vol. 64, no. 16, pp. 4351–4362, 2016.

[46] O. Alghushairy, R. Alsini, T. Soule, X. Ma, "A review of local outlier factor algorithms for outlier detection in big data streams," *Big Data and Cognitive Computing*, vol. 5, no. 1, 2020.

[47] M. Galphade, V. Nikam, B. Banerjee, A. Kiwelekar, "Intelligent multiperiod wind power forecast model using statistical and machine learning model," *Bulletin of Electrical Engineering and Informatics*, vol. 11, pp. 1186–1193, may 2022, doi: 10.11591/EEI.V11I3.3756.

[48] T. Carpenito, J. Manjourides, "MISL: Multiple imputation by super learning," *Statistical methods in medical research*, vol. 31, no. 10, pp. 1904–1915, 2022.

Manisha Galphade

Manisha Galphade is Bachelor, and Masters in Engineering (Computer Science and Engineering) pursuing PhD in Computer Department of VJTI. She has 14 years of teaching experience. Her research interest includes Machine Learning, Deep learning, GIS, Geospatial Analysis, Weather Predictions, Wind Power Prediction Modelling, Data mining.

Dr V.B.Nikam

Dr V. B. Nikam Associate Professor, Computer Engg & IT,VJTI Mumbai, has done Bachelors,Masters and PhD in Computer Engineering.He has 25 yrs experience, guided 50+ PG,25+ UG projects,and Supervised 4 PhDs. He was felicitated with IBM TGMC-2010 DRONA award by IBM Academic initiatives. He is Senior Member (CSI),Senior Member (IEEE), Senior Member (ACM). He worked onBARC ANUPAM Supercomputer. He was invited to JAPAN for a K-Supercomputer study tour in 2013. He has received a grant-in-aid from NVIDIA for CUDA Teaching and Research,2013.Presently,he is PI & Coordinator, Faculty Development Center (Geo-informatics, Spatial Computing and BigData Analytics) funded by MHRD, Govt of India. He works in the area of Data Mining and Data Warehousing, Machine Learning, Geoinformatics,Big Data Analytics, Geo-Spatial Analysis,Cloud Computing, GPUHigh Performance Computing. You may visit his webpage www.drvbnikam.in.

Dr. Biplab Banerjee

Dr. Biplab Banerjee received the M.E. (Computer Science and Engineering) from Jadhavpur University, Kolkata, and Ph.D. in Satellite Image Analysis from the Indian Institute of Technology Bombay, Mumbai, India. Dr. Banerjee received the Excellence in Ph.D. Thesis Award for his Ph.D. thesis from the IIT Bombay. He is Postdoctoral Researcher at the University of Caen Basse-Normandy, France and the Istituto Italiano di Technologia Genova, Italy. He is engaged in many research projects including international collaborated projects. His interests include computer vision and machine learning. Dr Biplab currently supervising 9 PhD students, and has guided more than 25 MTech thesis so far. He is reviewer for IEEE journals, and Transactions in Image Processing, Applied Earth Observations and Remote Sensing, Neural Computing & Applications (Springer), Journal of Indian Society of Remote Sensing (Springer), Computer Vision and Image Understanding (Elsevier). He is Member of IEEE. Indo-Canadian Research Grants, Board of Research in Nuclear Sciences – Department of Atomic Energy, Vishveshwarya PhD Scheme, Institute for Plasma Research, GUJCOST-DST and many other MNCs.

Dr. Arvind W. Kiwelekar

Dr. Arvind W. Kiwelekar Professor in Computer Science, Dr. B. A. Technological University Lonere has done Ph. D. from Indian Institute of Technology, Mumbai. He has 26 years of experience. His research interest includes Software Engineering, Software Architecture, Applied Artificial Intelligence and Ontology.

Dr. Priyanka Sharma

Dr Priyanka Sharma has 24 years of experience that spans both Industry and Academia, she specialize in leading AI and Data Analytics based application development and integrated solution design for various inter-disciplinary domains. She is a NVIDIA DLI Ambassador and have trained more than 3000 professionals through NVIDIA Led hands-on training programs on Deep Learning, Computer Vision, Natural Language Processing and CUDA Programming. She has also been associated with other international firms as Corporate Trainer apart from being a passionate academician and professor in CSE Department, Nirma University. She has published more than 55 research papers in International Journal, Books and Conferences in the domain of Artificial Intelligence and Deep Learning. She was earlier Principal Investigator/Collaborator of NVIDIA Research Center at Nirma University and several other research projects funded under Shastri