Clean Technologies and Recycling

Clean Technologies and Recycling, 4(2): 125–145.
DOI: 10.3934/ctr.2024007
Received: 05 May 2024
Revised: 22 November 2024
Accepted: 26 November 2024
Published: 11 December 2024

https://www.aimspress.com/journal/ctr

Research article

Predicting wind power using LSTM, Transformer, and other techniques

Arun Kumar M[1], Rithick Joshua K[1], Sahana Rajesh[2], Caroline Dorathy Esther J[1] and Kavitha Devi MK[3,*]

1 Department of Computer Science and Engineering, Thiagarjar College of Engineering, Madurai-625015, Tamil Nadu, India
2 Department of Biotechnology, School of Bio Sciences and Technology, Vellore Institute of Technology, Vellore - 632014, Tamil Nadu, India
3 Faculty of Computer Science and Engineering, Thiagarjar College of Engineering, Madurai-625015, Tamil Nadu, India

* **Correspondence:** Email: mkkdit@tce.edu; Tel: +919842093048.

Abstract: Predicting wind turbine energy is essential for optimizing renewable energy utilization and ensuring grid stability. Accurate forecasts enable effective resource planning, minimizing reliance on non-renewable energy sources and reducing carbon emissions. Additionally, precise predictions support efficient grid management, allowing utilities to balance supply and demand in real time, ultimately enhancing energy reliability and sustainability. In this study, we bridge the gap by exploring various machine learning (ML) and deep learning (DL) methodologies to enhance wind power forecasts. We emphasize the importance of accuracy in these predictions, aiming to overcome current standards. Our approach utilized these models to predict wind power generation for the next 15 days, utilizing the SCADA Turkey dataset and Tata Power Poolavadi Data collected. We used R^2 scores alongside traditional metrics like mean absolute error (MAE) and root mean square error (RMSE) to evaluate model performance. By employing these methodologies, we aim to enhance wind power forecasting, thereby enabling more efficient utilization of renewable energy resources.

Keywords: temporal dependencies; feature engineering; ensemble methods; grid integration; hyperparameter tuning

1. Introduction

With the global population expanding rapidly and fossil fuel reserves dwindling, the imperative for clean, sustainable energy sources has never been more pressing. In this context, wind energy emerges as a viable solution, offering a renewable and eco-friendly substitute for conventional energy sources. Using wind to power windmills that grind grain and pump water has been a fundamental part of human civilization for generations. However, despite its potential, wind energy has historically faced economic challenges, often being deemed costly and unpredictable compared to conventional fuels like petroleum.

To address these issues and improve the precision and dependability of wind power projections, the topic of wind prediction has attracted a lot of interest. For wind prediction, there have historically been two primary methods used: historical data processing with artificial intelligence techniques and numerical simulation and meteorological information perception. The former utilizes real-time meteorological data to construct simulation models, while the latter draws historical data and AI algorithms for prediction.

Deep learning's introduction has transformed wind power prediction in recent years by making it possible to create complicated models that can capture the intricate correlations seen in wind data. Convolutional neural networks (CNN), long short-term memory networks (LSTMs), bidirectional LSTM (BiLSTM), gated recurrent units (GRU), bidirectional GRU (BiGRU), deep confidence networks (DBN), and autoencoders (AE) are a few examples of the deep learning models that have shown impressive effectiveness in resolving nonlinear issues and enhancing the accuracy of wind power prediction.

This work explores deep learning (DL) and machine learning (ML) approaches for wind power prediction with the goal of advancing grid management and integration of renewable energy sources. Through an investigation of diverse modeling methodologies and assessment criteria, the goal is to augment the dependability and efficacy of wind energy projections, thus expediting the shift toward a more sustainable energy landscape. The input to the model comprises multivariate time-series data, including wind speed, direction, temperature, and temporal features like hour and day. The output is the predicted wind power generation for a specified time horizon.

2. Literature review

LSTMs can capture complex temporal dependencies inherent in wind speed data. LSTMs can remember past information for an extended period, which allows them to capture these complex temporal dependencies and make accurate predictions for the short term. LSTMs can also handle nonlinearity and complex patterns in data very well. Recent studies delved into utilizing LSTMs for wind prediction, frequently implementing hybrid models with other techniques to enhance accuracy. Concurrently, researchers strive to refine the optimization process and enhance data preprocessing, both contributing to improved wind prediction accuracy.

In [1], a new optimization technique was presented based on stochastic fractal search and particle swarm optimization (SFS-PSO) to optimize the parameters of the LSTM network. This sequential hybrid architecture utilizes LSTM for initial predictions, followed by SFS-PSO optimization to refine the results and elevate evaluation criteria. The algorithm specifically incorporates SFS to enhance the exploitation phase, leading to improved performance and improved evaluation matrices such as mean absolute error (MAE), Nash–Sutcliffe efficiency (NSE), mean square error (MSE), coefficient of determination (R^2), and root mean squared error (RMSE). The experimental results illustrated that the proposed optimization of LSTM using the SFS-PSO model achieved the best results, with R^2 equaling 99.99% in predicting wind power values.

In [2], the goal was to beat LSTM and TCN model accuracy by incorporating the CEEMDAN (complete ensemble empirical mode decomposition with adaptive noise) algorithm. This technique decomposes the original wind energy data into multiple intrinsic mode functions (IMFs) representing different time scales and frequency components. CEEMDAN is applied to decompose initial wind energy data to cut down volatility and randomized change. In the training process of LSTM and TCN, LM is applied to discover the ideal super parameters that should move forward the exactness of the model and decrease those preparing the long haul of the model. TCN excels at long-term prediction due to its ability to capture long-range dependencies. The combination TCN and LSTM allows the model to capture both short-term and long-term trends in wind energy data, potentially leading to more accurate predictions. In this model, LSTM predicts the short-term components (IMF1 and IMF2) extracted by CEEMDAN, and TCN predicts the long-term components (IMF3 to IMFn) extracted by CEEMDAN. CEEMDAN-TCN and CEEMDAN-LSTM have close accuracy of 91.2% and 92.5%, respectively, while the accuracy of CEEMDAN-LSTM-TCN is 99.8%. Longer input data offers more information but includes both valuable and misleading elements. If the learning rate is too small, the model will converge too slowly, and the training time will be too long or will fall into a local minimum; on the other hand, if the learning rate is adjusted too large, the loss function may oscillate during the training process, making it difficult to converge. Therefore, it is necessary to seek suitable hyperparameters to balance various constraints to achieve better prediction results [3].

Multi-step grid search was proposed in [3] to find good parameters. It is divided into two steps. In the first step, relatively good parameters are found over a wide range, with a larger amount of parameter updates each time. In the second step, the search scale is narrowed based on good values obtained in the first stage and the update step is smaller, reducing the total search time without missing too many probabilities. The proposed hybrid CNN-LSTM model beats the traditional single models like RNN, LSTM, and CNN models in accuracy. The values for MAE, RMSE, CC, and R^2 of CNN-LSTM are 0.4783, 0.6480, 0.9528, and 0.9070, respectively. While LSTMs excel at capturing temporal dependencies, they may struggle with long-term predictions (days or weeks) due to the vanishing gradient problem.

In [4], a prediction model was proposed based on a gated transformer for medium- to long-term wind power prediction. In this paper, numerical weather prediction (NWP) data information was introduced as auxiliary information, improving data correlation and ensuring the effectiveness of data feature extraction. The inclusion of NWP data significantly improved the average accuracy of all prediction models by 117% for medium to long-term forecasts, as demonstrated in this study. Dilated convolution serves as an effective tool for extracting features from lengthy sequential data. The model's encoder component handles historical wind power and NWP data. Using multi-head attention and a gating mechanism, the encoder adeptly captures crucial features. As a result, the encoder produces an attention-weight map, which provides insights into the distribution of information within the data. The hyperparameter settings in this paper were as follows: Batchsize = 128; learning rate = 1×10^{-4}; using Kaiming weight initialization; epoch = 400; and dropout = 0.1. The gated transformer model outperformed all other models in terms of prediction accuracy, achieving an improvement of approximately 8% for short-term forecasts and 11% for medium- to long-term forecasts.

Table 1. Review of literature on AI-driven wind power prediction methods.

Paper name	Authors	Methodology	Hyperparameters	Strength and weakness	Accuracy (R^2)
Wind power prediction based on machine learning and deep learning models	Tarek, Zahraa & Shams, Mahmoud & Elshewey, Ahmed & El-kenawy, El-Sayed & Ibrahim, Abdelhameed & Abdelhamid, Abdelaziz & El-dosuky, Mohamed.	LSTM and SFS-PSO model (stochastic fractal search and particle swarm optimization)	ω, C_1, C_2, r_1, r_2, β, β' Batch size-64, Learning rate-0.0001, Epochs-50, Optimizer-Adam, Activation function (in output)-Linear, Activation function (in hidden (5 Layers))-ReLU, KNN regressor n-neighbors = 2, weights = distance, Bagging regressor n-estimators = 10, max-samples = 1, GB regressor n-estimators = 200, learning-rate = 0.1	The LSTM layer could potentially learn and exploit the temporal patterns within wind data to improve prediction accuracy. The SFS technique can help select relevant features from the wind dataset, potentially reducing noise and boosting training efficiency. PSO can improve robustness by optimizing hyperparameters.	MAE: 0.000002 NSE: 1.2×10^{-7} MBE: 0.00001 R^2: 99.99% RMSE: 0.00002
Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN	Chenjia Hu, Yan Zhao, He Jiang, Mingkun Jiang, Fucai You, Qian Liu	LSTM and time convolution network (TCN) with complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN)	Not mentioned	CEEMDAN decreases the influence of non-stationary and random fluctuation in the original data on the prediction. Accurately captures both short-term and long-term dependencies using LSTMs and TCNs.	MAE: 0.136 MSE: 0.059 R^2: 0.998
Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network	Zhipeng Shen, Xuechun Fan, Liangyu Zhang, Haomiao Yu	A hybrid LSTM-CNN model that uses grid search with early stopping	Learning rate = 0.011 and input length = 26 Three convolutional layers filter size is (1, 3), (1, 3), and (1, 2), and the stride is (1, 1). Filter 1 = x16 size (1,3) Filter 2= x16 size (1,3) Filter 3 = x16 size (1,2) Dropout ratio = 0.3 LSTM cells 1= 10 LSTM cells 2 = 10 Activation function - ReLU, Optimizers - Adam, Epochs- 30, Batch size- 128	The hybrid approach of LSTM and CNN combined with grid search and early stopping offers potential strengths for wind power prediction, namely addressing both temporal and spatial aspects of the data, finding the configuration that minimizes prediction error, and improving efficiency by stopping unnecessary evaluations.	MAE: 0.9870 R^2: 94.44% RMSE: 1.3718 CC: 0.9723
Research on Wind Power Prediction Based on a Gated Transformer	Huang, Qiyue & Wang, Yapeng & Yang, Xu & Im, Sio-Kei	Gated transformer	Batchsize = 128; Learning rate = 1×10^{-4}, Using Kaiming weight initialization; Epoch = 400; Dropout = 0.1	Efficiently processes long historical and future NWP data sequences relevant to wind power prediction. Transformer models can be computationally expensive, and the addition of the gating mechanism might further increase this complexity. The use of NWP data gives improved data correlation and improves feature extraction.	With NWP MSE: 0.0555 MAE: 0.1376 Without NWP MSE: 0.0631 MAE: 0.1408
Multistep short-term wind speed forecasting using a transformer	Huijuan Wu, Keqilao Meng, Daoerji Fan, Zhanqiang Zhang, Qing Liu	Transformer model with EEMD decomposition and specific adjustments for wind speed prediction	EEMD mode number = 16 Learning rate = 0.003, Optimization- Adam, Batch size = 512, Dropout = 0.1 Window size of $\boldsymbol{X}$ = 18 Sequence length of $\boldsymbol{X}$ (lx) = 24 Model dimension = 128 Feedforward dimension (de) = 256 Heads number = 2 Encoder layers = 1 Decoder layers = 1 Window size of $\boldsymbol{Y}$ = 18	Removal of start/end tags (SOS/EOS) makes the model more suitable for fixed-length prediction. EEMD potentially reduces noise and improves model generalizability. EEMD can extract different time-scale features. Transformers can capture long-range dependencies and are well-suited for handling sequential data.	MAE: 0.167 RMSE: 0.221 MAPE: 22.40 R: 0.9717
Wind Power Forecasting with Deep Learning Networks: Time-Series Forecasting	Wen-Hui Lin, Ping Wang, Kuo-Ming Chao, Hsiao-Chung Li, Zong-Yu Yang, and Yu-Huang Lai	Temporal convolutional network (TCN) algorithm of DLNs	Number of filters = 32 Kernel size = 10 Dilations = [1,2,4,8,16] Epochs = 50 Optimizer = Adam Number of stacks = 4 NP = 10, F = 0.5, CR = 0.3, G = 30	The proposed scheme effectively solves the long-distance dependency problem, as demonstrated by the input of large amounts of temporal-spatial series data such as one-year wind power data.	MAPE: 5.13%

In [5], it was proposed an EEMD and transformer hybrid forecasting system for multistep short-term wind speed forecasting based only on wind speed history data. EEMD decomposition strategy is employed in the data preprocessing stage to effectively reduce noise pollution in the original data. The proposed model is trained using very large-scale wind speed data (19 years of data), and performance evaluation is performed on wind speed data throughout the year. This paper set a new benchmark for wind speed forecasting by achieving significantly lower prediction errors compared to previous methods. At 3, 6, 12, and 24 hours, the model's average error (MAE) is 0.243, 0.290, 0.362, and 0.453, respectively, representing a substantial improvement over the state-of-the-art. This EEMD-transformer model outperforms the EEMD-gated recurrent unit (GRU) model in mean absolute error (MAE), root mean square error (RMSE), and R; MAE decreased by 3.5%, RMSE decreased by 4.7%, and R improved by 0.0018, while mean absolute percentage error (MAPE) increased by 0.91.

3. Methodology

3.1. Materials

A crucial field of study focused on improving the dependability and efficiency of renewable energy sources is wind power prediction. Using the Tata Power dataset, a unique methodology for wind power prediction is proposed in this study that combines classic machine learning (ML) methods with deep learning (DL) models.

This approach combines the strength of ML algorithms in capturing intricate patterns with the capability of DL models in handling temporal dependencies. By amalgamating these techniques, it is aimed to surpass conventional methods and achieve more accurate wind power predictions.

3.1.1. Dataset collection

The methodology is initiated by collecting the Tata Power dataset, encompassing multivariate time-series data related to power generation, wind speed, and wind direction, among other parameters. Preliminary preprocessing steps involved handling missing values and ensuring uniformity in data formats. Date and time information was meticulously extracted and formatted to facilitate temporal analysis. Furthermore, numerical features were scaled to normalize the data distribution, easing subsequent model training.

3.1.2. Data processing

To gain insights into the dataset's characteristics, an extensive exploratory data analysis (EDA) phase was conducted. To find underlying patterns and relationships between data, descriptive statistics and visualization tools like pair plots and correlation heatmaps were used. To further illustrate the connection between wind direction and wind speed and to get important insights into directional wind patterns, a wind rose plot was also built.

3.1.3. Feature engineering

Domain-specific knowledge was used to engineer additional features that may impact wind power generation. Features related to weather conditions, geographical location, and turbine specifications were incorporated to enrich the dataset and improve predictive performance. To lower noise and boost model efficiency, features that had no discernible impact on the target variable (power generation) were found in the dataset and pruned. This process involved careful analysis and domain expertise to retain only the most relevant features for prediction. To maintain consistency in the distribution of data and keep characteristics with higher magnitudes from controlling the model-training procedure, numerical features were scaled and normalized. During model optimization, this phase proved crucial for enhancing convergence and stabilizing the training process. Principal component analysis (PCA) and feature importance ranking were two methods used to identify the most useful features and lower the dataset's dimensionality. By focusing on the most relevant features, we aimed to streamline model training and enhance predictive accuracy. From Figure 1, we observe the distribution of the data. The LV Active Power and Theoretical Power Curve exhibit a plateau-like pattern, with higher data density at the lower and upper ends and noticeably lower density in the middle range. This distinct distribution highlights the non-uniform nature of the dataset. The data indicates that the wind direction is predominantly from the northeast. Additionally, the wind speed primarily falls within the range of 0–20 m/s.

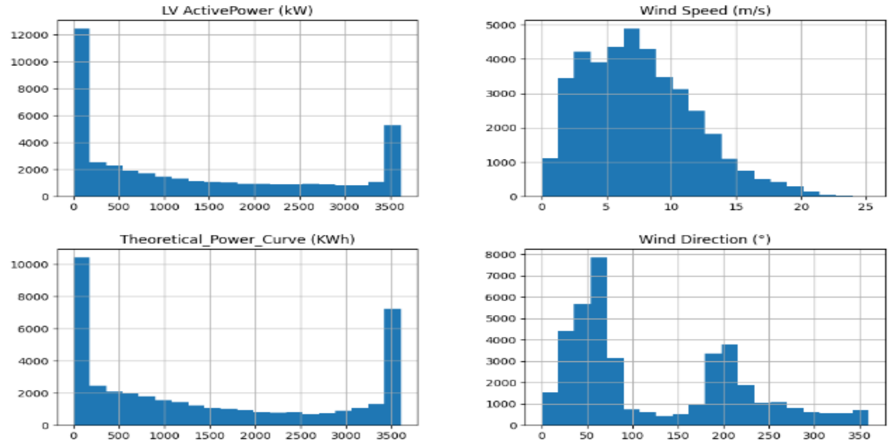

Figure 1. Data visualization of four parameters.

3.1.4. Visualization

As we can observe from Figure 2, there is a high positive association between wind speed and the theoretical power curve, as indicated by the correlation coefficient, which is extremely close to 1. This implies that the theoretical power curve prediction grows together with wind speed. Though not as strong as the link between wind speed and the theoretical power curve, the correlation between LV Active Power and the Theoretical Power Curve [7] is likewise positive and rather strong. This suggests that the theoretical power curve provides a decent estimate of actual power output, but other factors besides wind speed also influence the LV Active Power generation. To extract more meaningful features, new attributes like week, month, season, day, and hour are created from the existing Date/Time column. Finally, categorical features such as Seasons are encoded using a dictionary for the machine learning models to understand them better. Numerical features (power, wind speed, theoretical power curve, and LV Active Power) are plotted against categorical features (week, month, season) using bar charts (Figures 3–5). This helps identify trends in power generation based on these factors.

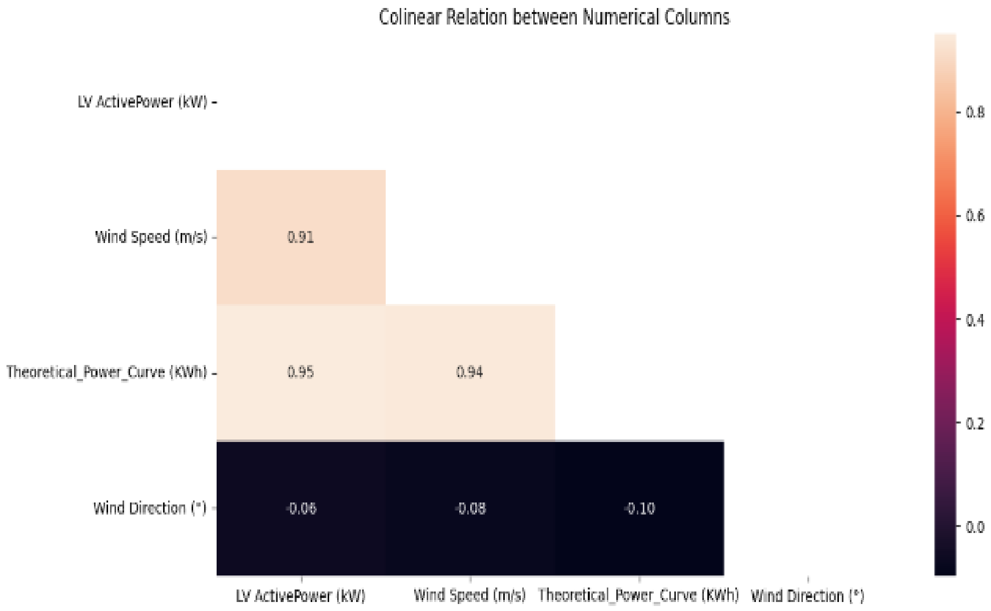

Figure 2. Colinear relation between numerical parameters.

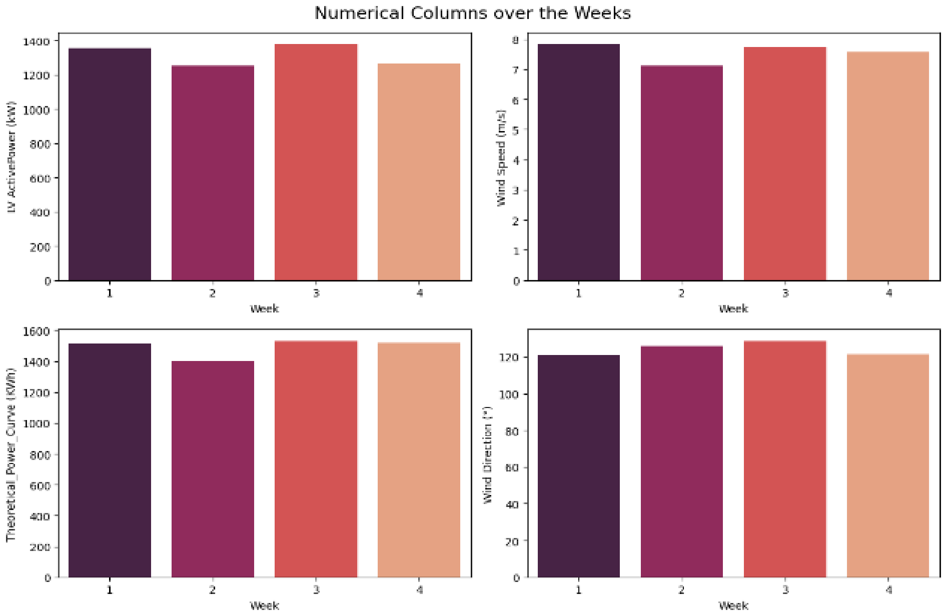

Figure 3. Numerical columns over the weeks.

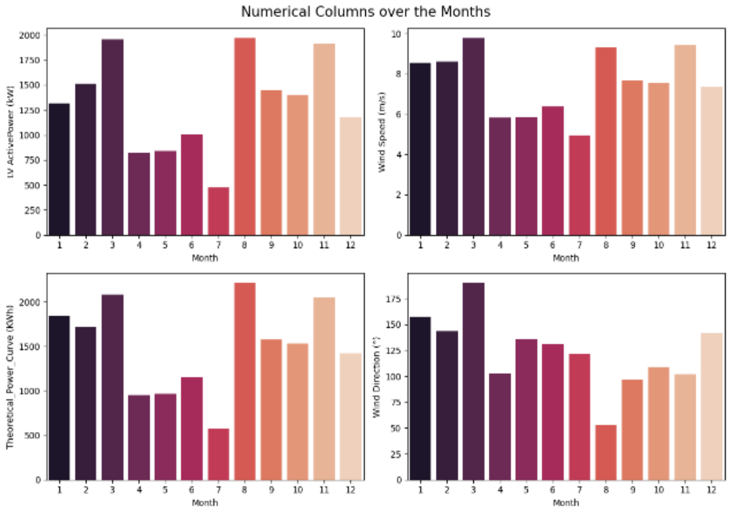

Figure 4. Numerical columns over the months.

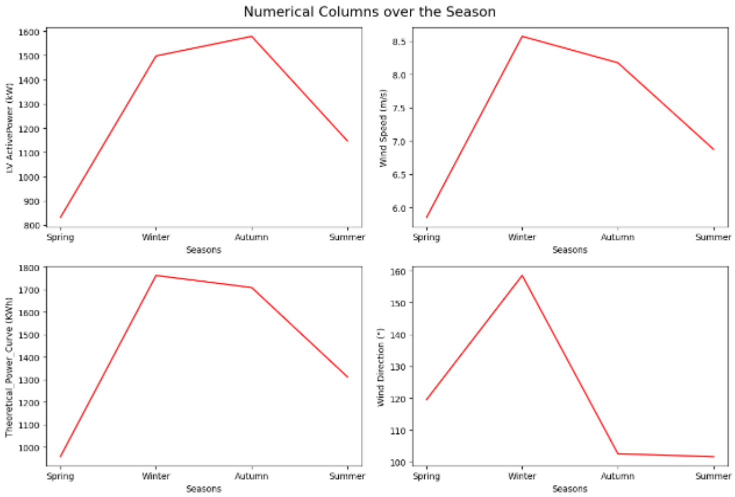

Figure 5. Numerical columns over the seasons.

3.1.5. Model selection

The capacity of ensemble regression models to represent intricate interactions within the dataset leads to selection. Gradient boosting regressor, support vector regressor (SVR), random forest regressor, linear regression, extra trees regressor, AdaBoost regressor, decision tree regressor, XGBoost regressor, and XGBoost with random forest (XGBRF) regressor are the ensemble models that were chosen for this investigation. These models provide several ways to prevent overfitting and identify the underlying trends in the data.

Each ensemble regression model was picked for wind power prediction based on how well it captured various features of the dataset. Gradient boosting regressor offers sequential model fitting, reducing bias and variance in predictions. Support vector regressor (SVR) effectively handles high-dimensional data and captures complex nonlinear relationships. Random forest regressor [8] is robust to overfitting and computationally efficient, suitable for large datasets. Linear regression provides simplicity and interpretability, serving as a baseline for comparison.

Extra trees regressor utilizes ensemble learning for improved robustness and generalization. AdaBoost regressor focuses on improving weak learners' performance through sequential training. Decision tree regressor captures nonlinear relationships in data through hierarchical partitioning. XGBoost regressor [9] offers scalability, speed, and regularization techniques for handling diverse datasets. XGBoost with random forest regressor (XGBRF) combines strengths for enhanced performance and robustness. Each model contributes to a comprehensive approach to wind power prediction, capitalizing on respective strengths to capture the dataset's complexity effectively.

3.1.6. Regression algorithms that were considered for predicting wind turbine power generation

1. Gradient boosting regressor (GBR): A method of ensemble learning that minimizes the residual errors from the previous tree while creating successive trees. The total of the predictions made by several poor learners is the expected output.

$$[\ \widehat{y_i}\ =\ \sum_{k=1}^{K}(f_k(x_i)\ ]$$

where $(\widehat{y_i})$ is the predicted output for the (ith) instance, (K) is the number of weak learners (trees), and $(f_k(x_i))$ represents the prediction of the (kth) weak learner for the (ith) instance.

2. Support vector regressor (SVR): Utilizes support vectors for regression. SVR maximizes the margin by locating the hyperplane that fits the data points the best.

$$[\,minimize\left(\frac{1}{2}|w|^2\ +\ C\sum_{i=1}^{n}(\xi_i\ +\ \xi_i^*\ )\right)]$$

where (w) are the weights of the hyperplane, (C) is the regularization parameter, and (n) is the number of data points.

3. Random forest regressor (RFR): A decision tree-based strategy for group learning. During training, it creates a number of decision trees and outputs the average forecast of each tree.

$$[\,\widehat{y_i}\ =\ \frac{1}{N}\sum_{j=1}^{N} f_j(x_i)\ ]$$

where $(\widehat{y_i})$ is the predicted output for the (i^{th}) instance, (N) is the total number of trees, and $(f_j(x_i))$ represents the prediction of the (j^{th}) tree for the (i^{th}) instance.

4. Linear regression: An approach to linear modeling that presupposes a linear relationship between the goal variable and the input features.

$$[\widehat{y_i}\ =\ \beta_0\ +\ \beta_1 x_{i1}\ +\ \beta_2 x_{i2} + \cdots +\ \beta_p x_{ip}\ ]$$

where $(\widehat{y_i})$ is the predicted output for the (i^{th}) instance, $(\,\beta_0)$ is the intercept, and $(\beta_1, \beta_2, \dots, \beta_p)$ are the coefficients of the input features $(x_{i1}, x_{i2}, \dots, x_{ip})$.

5. Extra trees regressor (ETR): An ensemble learning algorithm similar to random forest but with random splits.

$$[\, \hat{y_\iota} = \frac{1}{N}\sum_{j=1}^{N} f_j(x_i) \,]$$

where $(\hat{y_\iota})$ is the predicted output for the $(\, i^{th} \,)$ instance, $(\, N \,)$ is the total number of trees, and $(f_j(x_i))$ represents the prediction of the $(\, j^{th} \,)$ tree for the $(\, i^{th} \,)$ instance.

6. AdaBoost regressor: A boosting algorithm that builds a strong model by combining multiple weak learners. It iteratively adjusts the weights of misclassified samples to focus on the harder cases.

$$[\, F_m(x) \;=\; F_{m-1}(x) \;+\; \alpha_m h_m(x) \,]$$

where $(F_m(x))$ is the ensemble prediction after $(\, m \,)$ iterations, $(\, \alpha_m \,)$ is the weight of the $(\, m^{th} \,)$ weak learner, and $(h_m(x))$ represents the prediction of the $(\, m^{th} \,)$ weak learner.

7. Decision tree regressor: A non-parametric model that forecasts a target variable's value using decision rules that are inferred from the attributes of the data.

$$[\, \hat{y_\iota} \;=\; predict(x_i) \,]$$

where $(\, \hat{y_\iota})$ is the predicted output for the $(\, i^{th} \,)$ instance and $(predict(x_i))$ represents the prediction of the decision tree for the $(\, i^{th} \,)$ instance.

8. XGBoost regressor: A gradient boosting algorithm that optimizes the mean squared error objective function by adding weak learners.

$$[\, \hat{y_\iota} \;=\; \sum_{k=1}^{K} f_k(x_i) \,]$$

where $(\hat{y_\iota})$ is the predicted output for the (i^th) instance, $(\, K \,)$ is the number of weak learners (trees), and $(f_k(x_i))$ represents the prediction of the (k^th) weak learner for the (i^th) instance.

9. XGBRF regressor: A variant of XGBoost designed for random forests, which applies a similar boosting approach but utilizes randomization in the tree-building process.

$$[\, \hat{y_\iota} = \sum_{k=1}^{K} f_k(x_i) \,]$$

where $(\hat{y_\iota})$ is the predicted output for the (i^th) instance, $(\, K \,)$ is the number of weak learners (trees), and $(f_k(x_i))$ represents the prediction of the (k^th) weak learner for the (i^th) instance.

For every ensemble regression model, training and assessment are done iteratively. The model is fitted using the training data ('X_1', 'y_{train}'), and its performance is evaluated using the testing data ('$X1_{test}$', 'y_{test}'). To assess the model's goodness of fit and prediction accuracy, evaluation measures like the root mean squared error (RMSE) and coefficient of determination (R^2 score) are computed.

Hyperparameter tuning is essential for optimizing the performance of each regression model. Each model has a defined hyperparameter search space that contains values for n-estimators, max-depth, learning rate, min-child-weight, and base-score. To effectively search throughout the parameter grid and identify the ideal hyperparameters that optimize the R^2 score, randomized search cross-validation [10] is used.

The model with the highest R^2 score is chosen as the best-performing model following hyperparameter adjustment. K-fold cross-validation is used to analyze the performance of the chosen model in order to gauge its capacity for generalization. The cross-validation scores of many models are compared using visualization techniques, making it easier to determine which model is best for predicting wind output. To facilitate the training process of the models, a function ‘compile_and_fit (model, window, patience = 3)’ is crafted. This function streamlines the compilation and training steps. It incorporates early stopping via ‘tf.keras.callbacks.EarlyStopping’ to mitigate overfitting. The models are assembled using the Adam optimizer with a learning rate of ($lr = 0.01$) and mean absolute error (MAE) as the loss function.

For multi-step prediction [11], two distinct strategies are explored:

1. Single-shot predictions: This approach entails predicting the entire time series in one step. It is suitable for scenarios where a sequence of future values needs to be forecasted collectively.

2. AutoRegressive model: In this paradigm, the model makes predictions iteratively, with each output being fed back as input for the subsequent prediction step. This method is adept at capturing dynamic temporal dependencies within the data.

Two baseline models [12] are devised to establish a performance benchmark against which the more complex models can be evaluated:

Last baseline: This model simply repeats the last input timestep for the required number of output timesteps. It serves as a straightforward yet intuitive baseline for comparison purposes.

Repeat baseline: Here, the previous 15 days’ data is replicated, assuming that the subsequent 15 days will exhibit similar patterns. This simplistic model offers a basic estimation strategy.

These baseline models provide initial insights into the predictive capacity of the subsequent models.

Several single-shot models are developed, each with its unique architecture and capabilities:

Linear model: This model predicts the entire sequence in one step with linear layers. It reshapes the output to conform to the desired output shape.

$$[\, y_{linear} = W_{linear} \times x + b_{linear} \,]$$

where (W_{linear}) represents the weight matrix, (x) denotes the input, and (b_{linear}) is the bias vector.

Dense model: The ability of the model to recognize complex patterns in the data is improved by adding dense layers between the input and the output. Performance may be enhanced by this architecture’s introduction of nonlinear transformations into the prediction process.

$$[\, y_{dense} = \sigma\,(W_{dense} \times x + b_{dense} \,)\,]$$

where (σ) represents the activation function, (W_{dense}) denotes the weight matrix, (x) is the input, and (b_{dense}) is the bias vector.

CNN model: By incorporating a convolutional layer, this model is adept at capturing local patterns and dependencies within the data. The convolutional operation enables the model to extract spatial features, making it particularly suitable for sequential data analysis.

$$[\, y_{conv} = \sigma\,(W_{conv} * x + b_{conv}\,)\,]$$

where ($*$) denotes the convolution operation, (σ) represents the activation function, (W_{conv}) is the convolutional kernel, (x) is the input, and (b_{conv}) is the bias vector.

RNN model: Benefitting from LSTM layers [13], this model focuses on capturing temporal dependencies within the data. The recurrent nature of LSTM enables it to retain information over time, facilitating the prediction of multi-step sequences.

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

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

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

$$[\, \widetilde{C_t} = tanh\, tanh \ \left(W_C \cdot [h_{t-1}, x_t] + b_C \right)]$$

$$[\, C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}]$$

$$[\, h_t = o_t \odot tanh\, tanh\, (C_t)]$$

where (i_t), (f_t), (o_t), $(\widetilde{C_t})$, (C_t), and (h_{t-1}) represent the input gate, forget gate, output gate, cell input, cell state, and hidden state at the time (t), respectively. (W_i), (W_f), (W_o), and (W_C) denote the weight matrices, (x_t) represents the input at time (t), and (b_i), (b_f), (b_o), (b_C) are the bias vectors.

Because wind power generation datasets contain temporal relationships within sequential data, long short-term memory (LSTM) models were utilized because of their effectiveness in capturing these dependencies. The LSTM architecture used a number of layers with progressively larger units to efficiently record complex temporal patterns. In order to reduce overfitting and enhance model generalization, dropout layers were added. For wind power prediction, transformers, which are renowned for their capacity to detect long-range dependencies, were also investigated. The transformer architecture consisted of multiple transformer encoder blocks, followed by global average pooling and dense layers for prediction. Both LSTM and transformer models were meticulously trained and evaluated to gauge effectiveness in predicting wind power generation over multiple time steps.

Hyperparameter tuning was conducted to optimize the performance of LSTM and transformer models, ensuring they captured the complex temporal dynamics of wind power generation accurately.

To increase overall prediction accuracy and robustness, predictions from numerous separate models were combined using ensemble learning approaches, which took advantage of varied viewpoints on the data. Each ensemble model was trained and evaluated iteratively, and predictions were aggregated to produce a final prediction, resulting in a more accurate and reliable wind power prediction model.

4. Results and discussion

4.1. Dataset description

The dataset offers a thorough summary of environmental factors and wind turbine performance [14]. LV ActivePower (kW), wind speed (m/s), theoretical power curve (KWh), and wind direction (°) are its four main components. The LV ActivePower (kW) column indicates the real power output that a wind turbine produces. With a mean of roughly 1307.68 kW and a standard deviation of 1312.46 kW, its values range from a minimum of −2.471405 kW to a maximum of 3618.732910 kW. The wind speed at the turbine location is measured and given in m/s in the Wind Speed (m/s) column. With a mean of roughly 7.56 m/s and a standard deviation of 4.23 m/s, wind speeds range from 0 to 25.206011 m/s. The theoretical power curve (KWh) most likely represents the estimated power production of the turbine in KWh under ideal circumstances. With a standard deviation of 1368.02 KWh and a mean of roughly 1492.18 KWh, its values range from 0 to 3600 KWh. Finally, the column wind direction (°)

offers information about the predominant wind direction at the turbine location, expressed in degrees (°). Values for wind direction span from 0° to 359.997589°, with a standard deviation of 93.44° and a mean direction of roughly 123.69°. To see pairwise correlations between the numerical features (wind speed, theoretical power curve, wind direction, and LV ActivePower), a scatter matrix was plotted (Figure 6). Wind turbine systems are unable to produce any power when the wind speed is less than 4 m/s, as can be seen from the scatterplot between wind speed and the theoretical power curve. The relationship between wind speed and altitude is linear between 4 and 11 m/s, meaning that increasing wind speed allows turbines to produce more electricity; beyond 11 m/s, the power output reaches a saturation point of 3600 KWh.

Table 2. ML model evaluation metrics.

Model name	R^2 score	RMSE
Gradient boosting regressor	94.646846	302.227414
Random forest regressor	97.319413	213.867095
Extra trees regressor	97.667454	199.500571
Decision tree regressor	95.247386	284.770665
XGB regressor	97.965704	186.309943
XGBRF regressor	94.158605	315.709240

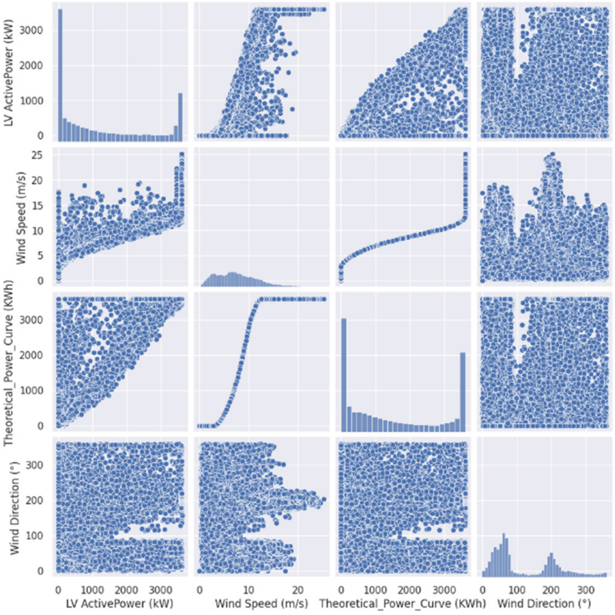

Figure 6. Scatter matrix.

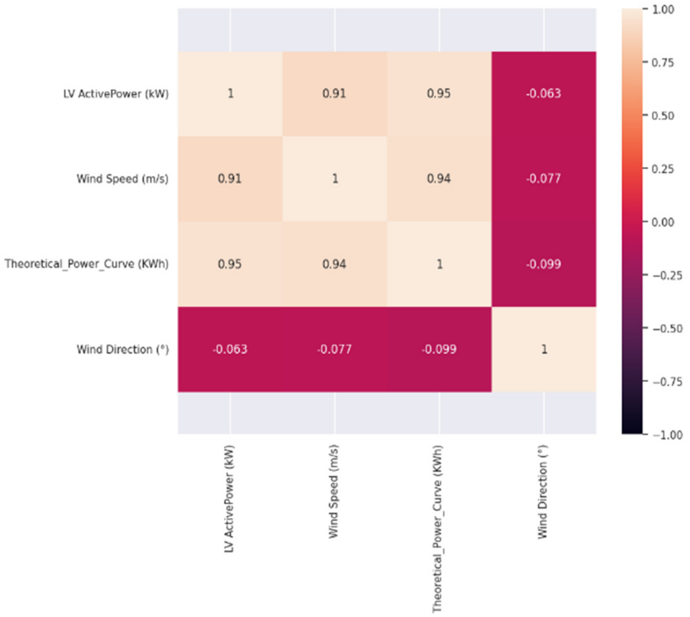

Figure 7. Correlation matrix.

The associations between the variables LV Active Power (kW), wind speed (m/s), Theoretical Power Curve (kWh), and wind direction (°) are revealed by the correlation matrix in Figure 7. Notably, LV Active Power (kW) exhibits strong positive correlations with Theoretical Power Curve (kWh) (roughly 0.95) and wind speed (m/s) (nearly 0.91). These correlations suggest that higher theoretical power output and wind speeds are associated with higher active power generation. Furthermore, there is a strong positive connection (about 0.94) between wind speed (m/s) and Theoretical Power Curve (kWh), indicating that higher wind speeds result in higher theoretical power output. On the other hand, the variable wind direction (°) exhibits minor correlations with the remaining variables, suggesting a limited linear relationship with theoretical power output, wind speed, and active power. Among the models assessed [15], the Random Forest Regressor, Extra Trees Regressor, and XGB Regressor stand out as notable strengths. XGB Regressor power production predictions are depicted in Figure 8. These models demonstrate high R^2 scores, indicating the ability to effectively capture the variance in wind power. Moreover, low RMSE values signify a high level of accuracy in predicting wind power output. This suggests that these models could be valuable assets in accurately forecasting wind power, a critical aspect of renewable energy management. Decision Tree Regressor and Gradient Boosting Regressor models perform well but may fall short in accuracy compared to top-performing alternatives. Further optimization through techniques like cross-validation and hyperparameter tuning can enhance the reliability and robustness of the selected models, ensuring effectiveness in real-world wind power prediction scenarios.

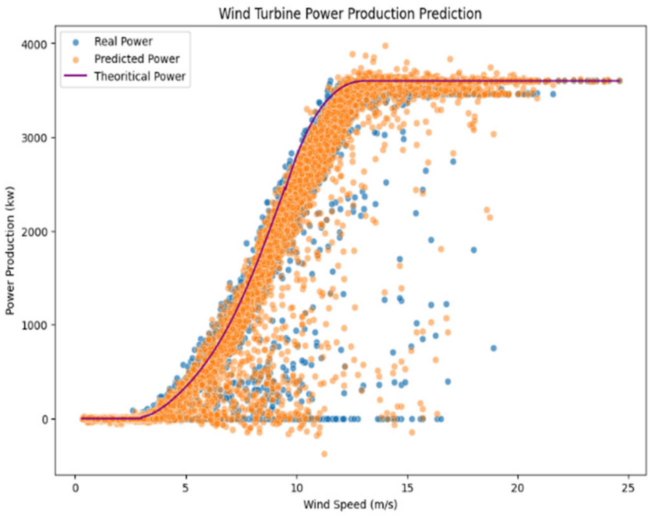

Figure 8. Wind turbine power production prediction.

4.2. DL methodology

Mean absolute error (MAE) and root mean squared error (RMSE) were two important metrics taken into consideration while assessing the efficacy of the wind power forecasting models. The baseline models, last baseline and repeat baseline, provided important reference points. The repeat baseline outperformed the last baseline, which is predicted from previous data as illustrated in Figure 9. However, both baseline models were outperformed by the single-shot models.

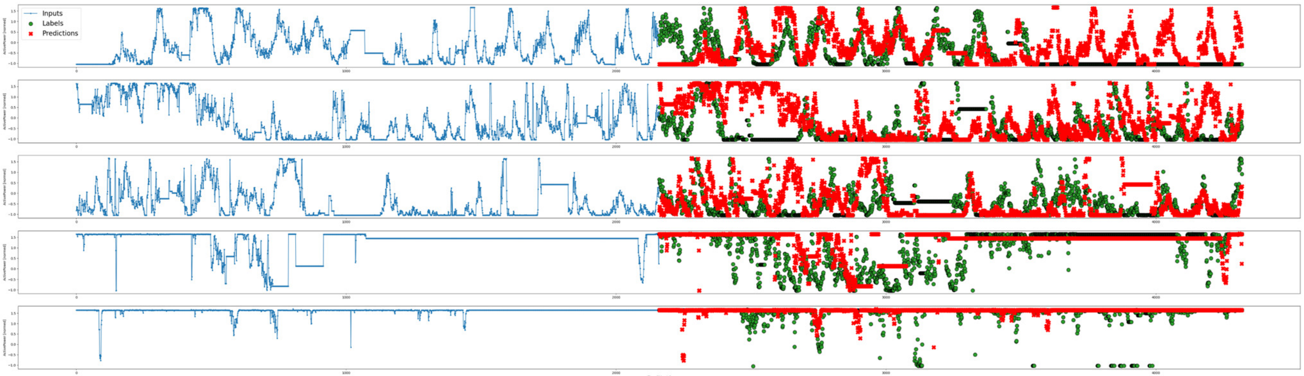

Figure 9. Repeat baseline model.

The dense model performed the best out of all the single-shot models, including the CNN model, LSTM model, linear model, and dense model. In comparison to the other single-shot models, the thick model produced the lowest MAE and RMSE when it included thick layers for more intricate transformations. This suggests that more complex patterns and connections in the data were captured by the dense model, producing more accurate predictions.

The LSTM model significantly underperformed in comparison to the other models but still produced competitive results, whereas the linear model and CNN model performed rather similarly.

With the CNN model concentrating on spatial patterns and the LSTM model specializing in capturing temporal relationships, each model's architecture has advantages and disadvantages of its own.

In terms of MAE and RMSE, the dense model performed best overall, as depicted in Figure 10. However, further analysis is warranted to explore additional factors such as computational efficiency, interpretability, and generalization performance on unseen data. Additionally, it would be valuable to conduct more extensive experimentation and fine-tuning to optimize the models further and potentially uncover new insights into wind power forecasting.

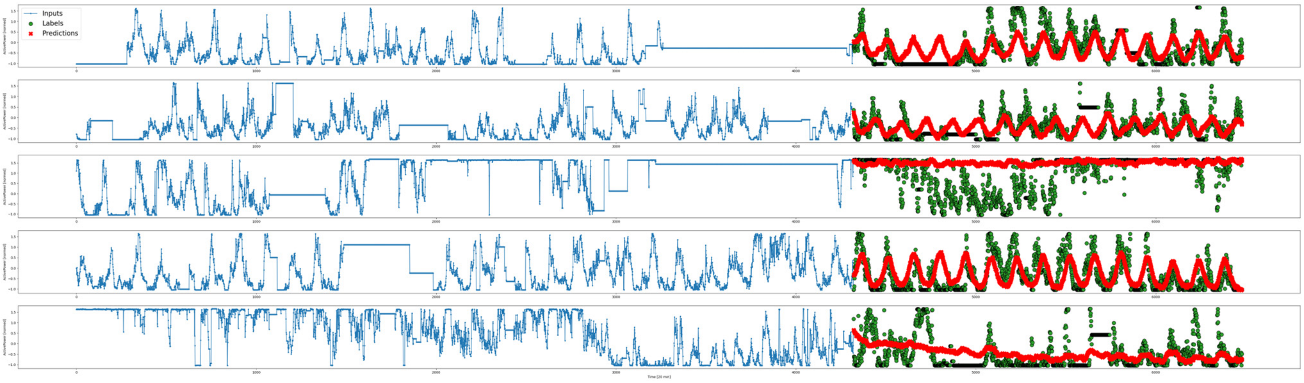

Figure 10. Dense model.

Mean absolute error (MAE) and root mean squared error (RMSE) performance of the LSTM model was not as good as that of other models, but it nevertheless showed that it could capture temporal dependencies in the wind power generation data. As observed from Figure 11, the LSTM model demonstrated its efficacy in modeling sequential data with competitive results, with an MAE of 0.0632 and RMSE of 0.0082. Because LSTMs are recurrent, they can store information across time, which makes them ideal for identifying subtle temporal patterns and dependencies in data.

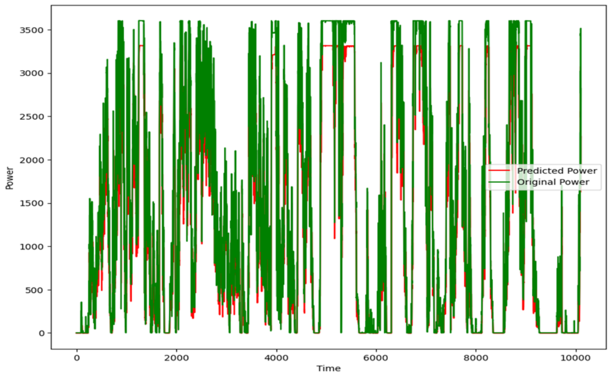

Figure 11. LSTM model.

However, with noticeably lower MAE and RMSE values, as depicted in Figure 12, the transformer model fared better than the LSTM model. With an MAE of 0.0074 and RMSE of 0.00083, the transformer model demonstrated its prowess in capturing long-range dependencies and spatial patterns within the wind power generation data. Transformers excel in handling sequences with complex relationships across both temporal and spatial dimensions, making them suitable for capturing the intricate dynamics present in wind power generation datasets.

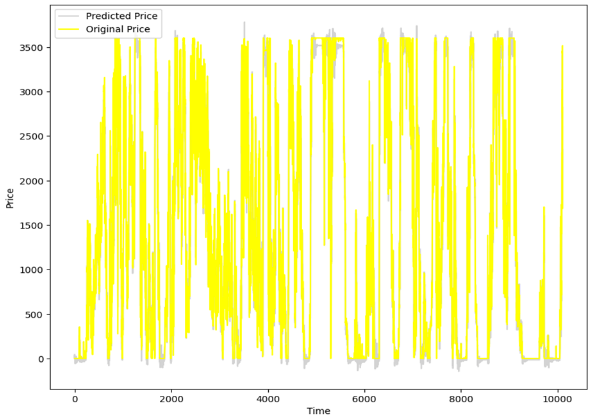

Figure 12. Transformer model.

Prediction accuracy increased because of combining the advantages of both architectures in the ensemble of transformer and LSTM models. The ensemble model, which had an MAE of 0.0462 and an RMSE of 0.0075, took advantage of the complementing qualities of transformers' skill in modeling long-range dependencies and spatial patterns and LSTM's capacity to capture temporal dependencies. The ensemble model outperformed the performance of the individual models alone by combining predictions from both architectures to provide predictions that were more correct, as shown in Figure 13. Figure 14 demonstrates the RMSE and MAE scores of the transformer model over 25 epochs.

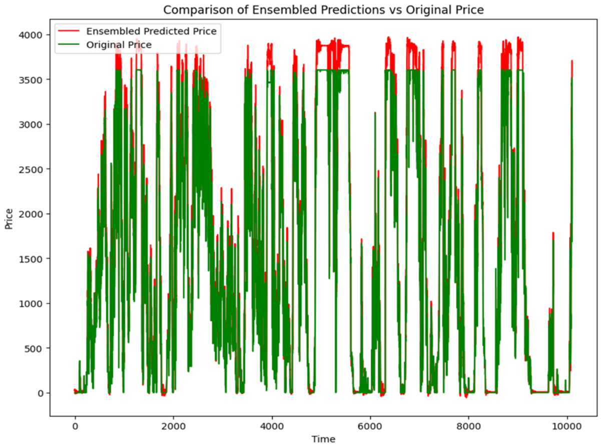

Figure 13. LSTM Transformer model.

The LSTM, transformer, and ensemble model comparison emphasizes how crucial it is to choose the right model architectures depending on the type of data and the intended prediction goal. While LSTM and transformer models illustrate distinct advantages in capturing temporal and spatial patterns, respectively, their combination in an ensemble model can further enhance prediction accuracy by complementing strengths. Further analysis is warranted to explore additional factors such as computational efficiency, interpretability, and generalization performance on unseen data. Additionally, continued experimentation and fine-tuning may uncover new insights and further optimize the models for wind power forecasting applications.

Table 3. DL model evaluation metrics.

Model name	MAE	RMSE
Baseline	0.7966	35.4678
Repeat baseline	0.6261	25.5001
Single shot model	0.2981	19.0071
Dense	1.1859	44.7766
CNN	1.2686	52.3993
RNN	0.6424	20.0467
LSTM	0.0632	0.0082
Transformer	0.0074	0.00083
Ensemble (LSTM + Transformer)	0.00462	0.00075

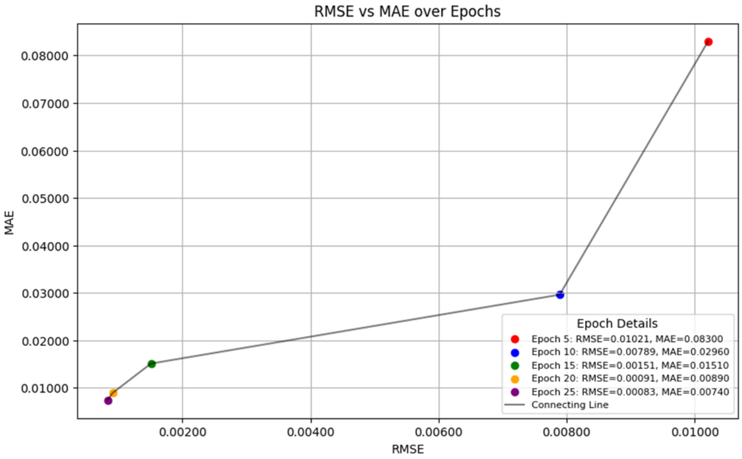

Figure 14. RMSE vs. MAE over epochs for the transformer model.

5. Conclusions

This work concludes with a thorough methodology for wind power prediction that incorporates ensemble methodologies, deep learning models, and conventional machine learning algorithms. By carefully preparing the data, conducting exploratory data analysis, and feature engineering, we were able to create the Tata Power, Turkey Scada dataset and acquire important insights into its properties. By ensemble regression models, including gradient boosting, random forest, and XGBoost, we captured complex relationships within the data, laying the foundation for accurate wind power predictions. Figure 15 illustrates a step-by-step technical roadmap outlining the progression of models employed in our study.

Furthermore, we explored the effectiveness of deep learning architectures, including LSTM and transformer, in capturing temporal and spatial dependencies within the wind power generation data.

While LSTM models demonstrated competitive performance in capturing temporal patterns, the transformer model excelled in modeling long-range dependencies and spatial patterns, leading to significantly improved prediction accuracy. Moreover, the combination of LSTM and transformer in an ensemble model further enhanced prediction accuracy, surpassing the performance of individual models alone. This highlights the importance of diverse model architectures to capture the nuanced dynamics present in wind power generation datasets effectively.

Overall, this study's findings advance the field of wind power prediction research by illuminating the advantages and disadvantages of different model architectures. The established methodology offers a strong foundation for precise wind power forecasting, which is essential for maximizing the dependability and efficiency of renewable energy sources. Future research may focus on exploring additional factors, such as computational efficiency and interpretability, and conducting further experimentation to optimize models for real-world wind power forecasting applications. By continuing to refine and innovate in this field, we can accelerate the transition to a sustainable energy future.

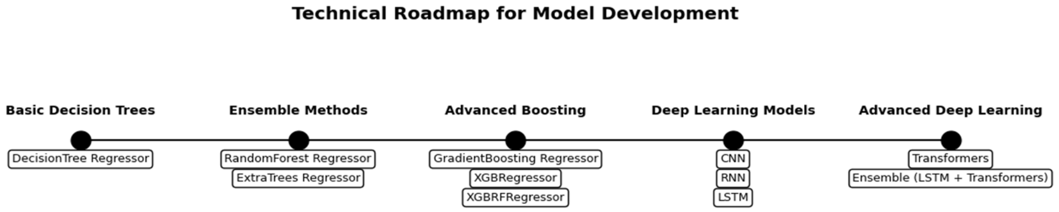

Figure 15. Technical roadmap for all models used.

Author contributions

Arun Kumar M was the head of the project and took the lead in developing the deep learning (DL) and Transformer models. Rithick Joshua K was responsible for the implementation of the machine learning (ML) models used in wind power prediction. Sahana Rajesh contributed by managing the preprocessing methods, including feature engineering and data normalization. Caroline Dorathy Esther handled the data collection and documentation throughout the project. Finally, Dr. Kavitha Devi MK was in charge of stakeholder responsibilities and the organization of the project's activities.

Conflict of interest

All authors declare that there are no competing interests.

Ethical approval

The authors declare that this research was conducted in accordance with the ethical principles.

Data availability

The data won't be available on request. Private source.

Use of AI tools

The authors declare that artificial intelligence (AI) tools, including regression and deep learning models, were used for data analysis and model development in this study. All model outputs were reviewed and validated to ensure the accuracy and relevance of the results.

References

1. Tarek Z, Shams MY, Elshewey AM, et al. (2023) Wind power prediction based on machine learning and deep learning models. *Comput Mater Continua* 74: 715–732. https://doi.org/10.32604/cmc.2023.032533
2. Hu C, Zhao Y, Jiang H, et al. (2022) Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN. *Energy Rep* 8: 483–492. https://doi.org/10.1016/j.egyr.2022.09.171
3. Shen Z, Fan X, Zhang L, et al. (2022) Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network. *Ocean Eng* 254: 111352. https://doi.org/10.1016/j.oceaneng.2022.111352
4. Huang Q, Wang Y, Yang X, et al. (2023) Research on wind power prediction based on a gated transformer. *Appl Sci* 13: 8350. https://doi.org/10.3390/app13148350
5. Wu H, Meng K, Fan D, et al. (2022) Multistep short-term wind speed forecasting using transformer. *Energy* 261: 125231. https://doi.org/10.1016/j.energy.2022.125231
6. Wu H, Wang P, Chao KM, et al. (2021) Wind power forecasting with deep learning networks: Time-series forecasting. *Appl Sci* 11: 10335. https://doi.org/10.3390/app112110335.
7. Eryilmaz S, Devrim Y (2018) Theoretical derivation of wind plant power distribution with the consideration of wind turbine reliability. *Reliab Eng Syst Saf* 185: 44–50. https://doi.org/10.1016/j.ress.2018.12.018
8. Ho CY, Cheng KS, Ang CH (2023) Utilizing the random forest method for short-term wind speed forecasting in the coastal area of central Taiwan. *Energies* 16: 1374. https://doi.org/10.3390/en16031374
9. Phan QT, Wu YK, Phan QD (2021) A hybrid wind power forecasting model with XGBoost, data preprocessing considering different NWPs. *Appl Sci* 11: 1100. https://doi.org/10.3390/app11031100
10. Ponkumar G, Jayaprakash S, Kanagarathinam K (2023). Advanced machine learning techniques for accurate very short-term wind power forecasting in wind energy systems using historical data analysis. *Energies* 16: 5459. https://doi.org/10.3390/en16145459
11. Bai W, Jin M, Li W, et al. (2024). Multi-step prediction of wind power based on hybrid model with improved variational mode decomposition and sequence-to-sequence network. *Processes* 12: 191. https://doi.org/10.3390/pr12010191
12. Yuan C, Li J , Xie Y, et al. (2022) Investigation on the effect of the baseline control system on dynamic and fatigue characteristics of modern wind turbines. *Appl Sci* 12: 2968. https://doi.org/10.3390/app12062968
13. Elsaraiti M, Merabet A (2021) Application of long-short-term-memory recurrent neural networks to forecast wind speed. *Appl Sci* 11: 2387. https://doi.org/10.3390/app11052387

14. Huang C, Liu C, Zhong M, et al. (2024). Research on wind turbine location and wind energy resource evaluation methodology in port scenarios. *Sustainability* 16: 1074. https://doi.org/10.3390/su16031074

15. Piotrowski P, Rutyna I, Baczyński D, et al. (2022). Evaluation metrics for wind power forecasts: A comprehensive review and statistical analysis of errors. *Energies* 15: 9657. https://doi.org/10.3390/en15249657