

Wind Power Forecasting Using Grey Wolf Optimized Long Short-Term Memory Based on Numerical Weather Prediction

Mohamed El-Dosuky^{1,2*}, Reema Alowaydan¹, Bashayer Alqarni¹

¹Department of Computer Science, Arab East Colleges, Riyadh, KSA

²Faculty of Computers and Information Sciences, Mansoura University, Mansoura, Egypt

Email: *maldosuky@arabeast.edu.sa

How to cite this paper: El-Dosuky, M., Alowaydan, R. and Alqarni, B. (2024) Wind Power Forecasting Using Grey Wolf Optimized Long Short-Term Memory Based on Numerical Weather Prediction. *Journal of Power and Energy Engineering*, 12, 1-16. <https://doi.org/10.4236/jpee.2024.1212001>

Received: November 13, 2024

Accepted: December 15, 2024

Published: December 18, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Wind power generation is among the most promising and eco-friendly energy sources today. Wind Power Forecasting (WPF) is essential for boosting energy efficiency and maintaining the operational stability of power grids. However, predicting wind power comes with significant challenges, such as weather uncertainties, wind variability, complex terrain, limited data, insufficient measurement infrastructure, intricate interdependencies, and short lead times. These factors make it difficult to accurately forecast wind behavior and respond to sudden power output changes. This study aims to precisely forecast electricity generation from wind turbines, minimize grid operation uncertainties, and enhance grid reliability. It leverages historical wind farm data and Numerical Weather Prediction data, using k-Nearest Neighbors for pre-processing, K-means clustering for categorization, and Long Short-Term Memory (LSTM) networks for training and testing, with model performance evaluated across multiple metrics. The Grey Wolf Optimized (GWO) LSTM classification technique, a deep learning model suited to time series analysis, effectively handles temporal dependencies in input data through memory cells and gradient-based optimization. Inspired by grey wolves' hunting strategies, GWO is a population-based metaheuristic optimization algorithm known for its strong performance across diverse optimization tasks. The proposed Grey Wolf Optimized Deep Learning model achieves an R-squared value of 0.97279, demonstrating that it explains 97.28% of the variance in wind power data. This model surpasses a reference study that achieved an R-squared value of 0.92 with a hybrid deep learning approach but did not account for outliers or anomalous data.

Keywords

Wind Power Forecasting, Long Short-Term Memory, Numerical Weather

1. Introduction

Decarbonized economy aims at producing low levels of carbon emissions to lower global warming which poses a threat to humans [1]. Recently, new energy sources based on solar and wind energy have developed at a great rate. Generating wind power is among the most promising and ecological energy sources [2]. But wind power massive integration poses challenges in economic and safe power grid operation [3]. Wind Power Forecasting (WPF) is a vital task to improve efficiency of energy and operation safety of power grids [4].

Based on the time scales, WPF can be classified into very-short-term [5], short-term [6], medium-term, and long-term prediction [7] [8].

The rest of this paper is organized as follows. In Section 2, a gentle review of Wind Power Forecasting is provided before presenting the related work. In Section 3, the methodology and implementation details are provided, while Section 4 provides the results and discussions. Section 5 contains the conclusion and future work.

2. Related Work

Zhang *et al.* developed a wind power forecasting model using Numerical Weather Prediction data and actual wind farm data. They used a multivariate time series clustering algorithm and deep learning network, enhancing efficiency and accuracy. The Seq2Seq model outperformed other methods, with clustering significantly improving forecast accuracy [9]. This model faces numerous issues, including parameter tuning for Seq2Seq, which significantly affects forecasting performance due to its impact on layers, units, learning rate, momentum, and periods.

A study uses a Long Short-Term Memory (LSTM) deep learning network for wind power forecasting and the Gaussian Mixture Model (GMM) for error distribution analysis. The LSTM model outperforms other models in accuracy and convergence speed, while GMM offers superior evaluation performance, making it valuable for practical wind turbine power dispatching applications [10].

Shabbir *et al.* studied Estonia's short-term wind energy forecasting using real-time data. They applied three models: Support Vector Machine, Non-linear Autoregressive Neural Networks, and Recurrent Neural Network-Long Short-Term Memory. Results showed ML and deep learning models outperformed Transmission System Operator's forecasting algorithm [11]. Li *et al.* developed a hybrid prediction model for short-term wind power forecasting, combining support vector machine and an improved dragonfly algorithm. The model improved traditional dragonfly performance and was compared to other models [12].

Tang and Chien developed a short-term wind power prediction model using time convolution neural network and variational mode decomposition [13]. They extracted modal components with rich feature information and trained the model

using preferred components and historical power data. The model achieved higher prediction accuracy with MAPE and R^2 of 2.79% and 0.9985, respectively. Hong and Rioflorido developed a hybrid deep learning neural network for 24 h-ahead wind power generation forecasting, using a Convolutional Neural Network and a Radial Basis Function Neural Network [14].

A study provided a deep learning model for short-term wind turbine power forecasts using Temporal Convolutional Networks (TCN) [15], while another study proposed a transformer-based WPF model, based on attention mechanisms [16]. Finally, Chen proposed an ultra-short-term prediction model using a multi-layer Bi-GRU and Fully Connected layer, using real-world wind farm data [17].

A recent study performs wind power prediction based on various machine learning and deep learning models [18].

3. Methodology

This work proposes a framework, as shown in **Figure 1**. Initially, the wind farm historical data and data from Numerical Weather Prediction (NWP) are joined. Then preprocessing is performed to impute data for completing missing values using k-Nearest Neighbors (KNN). K-means-Clustering is used to prepare data for classification. The dataset is dichotomized into training and testing parts. Training constructs a model using Long Short-Term Memory (LSTM) which is then verified. Certain metrics are calculated to assess the accuracy of the model.

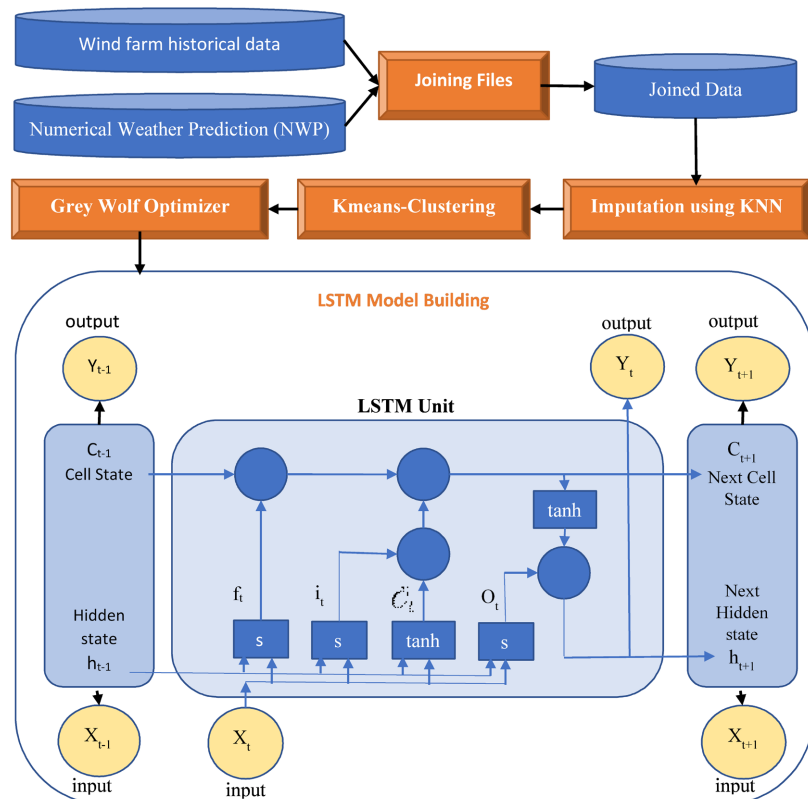


Figure 1. Proposed framework.

The Grey Wolf Optimizer (GWO) enhances Long Short-Term Memory (LSTM) performance for wind power data by optimizing the model’s hyperparameters and addressing the inherent complexity of wind data, which is highly non-linear and volatile. GWO, inspired by the hunting strategies of grey wolves, balances exploration and exploitation to efficiently tune parameters like learning rate, layer depth, and neuron count, enabling LSTMs to better capture temporal patterns and adapt to wind power’s intermittent behaviour. It also aids in selecting relevant features and initializing weights, reducing noise and improving model focus. By helping LSTMs navigate uncertainties and avoid local minima during training, GWO significantly boosts forecasting accuracy, as shown by lower error metrics in results compared to standalone LSTM or traditional optimization techniques.

3.1. Data Imputation Using K-Nearest Neighbors (KNN)

Missing data can be handled in different ways. The entire observation can be removed or replaced with the average, median or mode. These ways may reduce dataset variability. On the contrary, KNN imputation keeps dataset value and variability. **Figure 2** shows an example of 3-nearest neighbors.

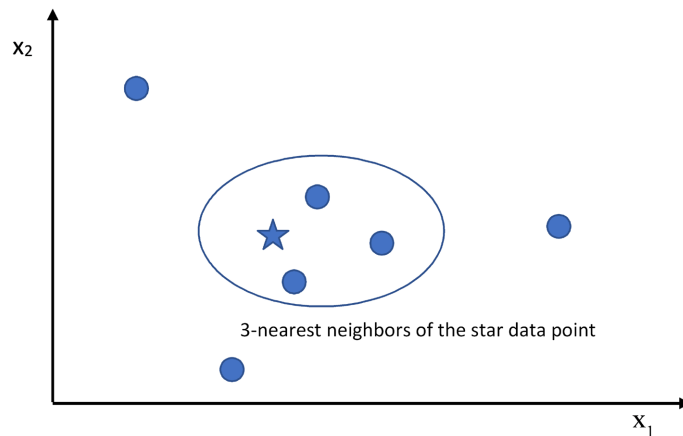


Figure 2. Three-nearest neighbors.

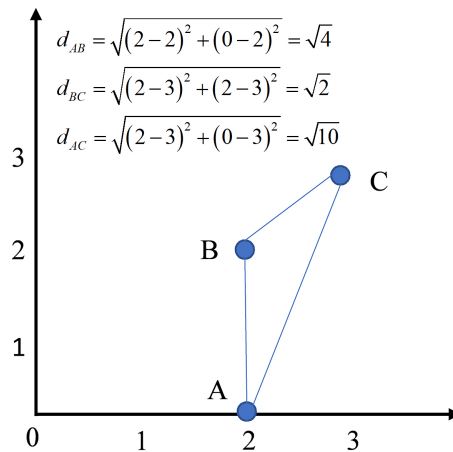


Figure 3. Euclidean distances.

As shown in **Figure 3**, Euclidean distances can be calculated for the data. The nearest neighbors are those having the shortest distance. At $k = 1$, A is the nearest neighbor to B.

3.2. K-Means-Clustering

Figure 4 shows how K-means-Clustering works. K-means clustering is an unsupervised machine learning algorithm used to group data points into distinct clusters. It aims to minimize the sum of squared distances between data points and their respective clusters. The algorithm involves initialization, assignment, update, and iteration until convergence. The algorithm assigns each data point to a cluster based on the nearest centroid, with the centroids representing the center points. However, the algorithm is sensitive to the initial placement of centroids, so it is often run multiple times with different initializations.

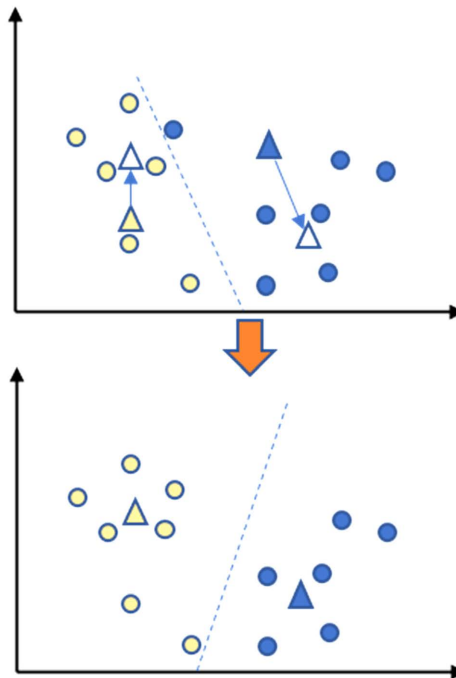


Figure 4. How K-means-Clustering works.

3.3. Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNNs) have feedback connections and can process input sequences of arbitrary length. However, training RNNs can be challenging due to gradient-based weight update algorithms [19]. Long Short-Term Memory (LSTM) is a type of RNN that can work with sequential data, making it effective for time series forecasting in various fields [20] [21].

The LSTM architecture, consisting of three logistic sigmoid gates and one tanh layer, is commonly used with a forget gate to limit information flow, with output typically ranging from 0 - 1. An LSTM cell has 3 inputs h_{t-1} , x_t and C_{t-1} and 2 outputs h_t and C_t . For a given time t , h_t is the hidden state, C_t is the cell

state or memory, x_t is the current data point or input. The first sigmoid layer has two inputs h_{t-1} and x_t where h_{t-1} is the hidden state of the previous cell [22].

In an LSTM network, the forget gate decides how much of the previous cell state to retain, while the input gate determines what new information to add. A candidate layer generates values to update the cell state, which combines these inputs with the retained information from the forget gate. For output, a sigmoid layer selects relevant parts of the cell state, which is scaled by a tanh layer to produce the final cell output. Together, these gates and layers enable the LSTM to effectively manage information flow and maintain long-term dependencies [23].

An LSTM network computes the output by mapping an input sequence $x = (x_1, \dots, x_T)$ to an output sequence $y = (y_1, \dots, y_T)$ and calculating the network unit activations using the following equations iteratively from $t = 1$ to T [24]:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \quad (4)$$

$$m_t = o_t \odot h(c_t) \quad (5)$$

$$y_t = \phi(W_{ym}m_t + b_y) \quad (6)$$

where W represents weights, W_{ic} , W_{fc} , W_{oc} are diagonal weights for peephole connections, the b represents bias, σ is the logistic sigmoid function, and i , f and o are the input, forget, output gates. Also, c is cell activation, while \odot is element-wise multiplication of vectors. Finally, ϕ is the network output activation function.

Multi-variate time series forecasting is a challenging task due to complex factors in real-life sequential data, such as dependencies, randomness, and noise. Hyperparameters and learning properties also affect performance. Traditional manual fine-tuning can lead to sub-optimal model representations and ill-performed learning parameters. To address these challenges, a new approach incorporates automatic feature extraction and optimal network configuration identification into the vanilla LSTM structure. Convolutional Neural Networks (CNNs) are hybridized with LSTM to extract fundamental features and construct more accurate feature representations.

3.4. Grey Wolf Optimizer (GWO)

The GWO is an optimization algorithm inspired by nature, modelled after grey wolves' social hierarchy and hunting mechanisms. It uses a four-class pack of wolves, alpha, beta, delta, and omega, with the best-fitting wolf being the alpha wolf. The algorithm updates the hierarchy each iteration, adjusting solutions based on the wolves' fitness function. The algorithm further explains grey wolf

hunting mechanisms mathematically [25].

The search process uses random positions of α , β , and δ , with coefficient vectors A and C assigned random values to facilitate a global search for prey.

Once a prey is found, the wolves encircle it; as shown in **Figure 1**. For i iteration, equations that represent the positions are:

$$D = |C \cdot V_{p(i)} - V_i| \quad (7)$$

$$V_{(i+1)} = V_{p(i)} - A \cdot D \quad (8)$$

where V_p is vector of prey position, i is the current iteration, and V denotes grey wolf position of a . A and C are computed as the following equations:

$$A = 2a \cdot r_1 - a \quad (9)$$

$$C = 2r_2 \quad (10)$$

Assuming that a grey wolf's location is (X, Y) and that of its prey is (X', Y') , the wolves can then adjust their positions in response to the prey's location, as shown in **Figure 5**.

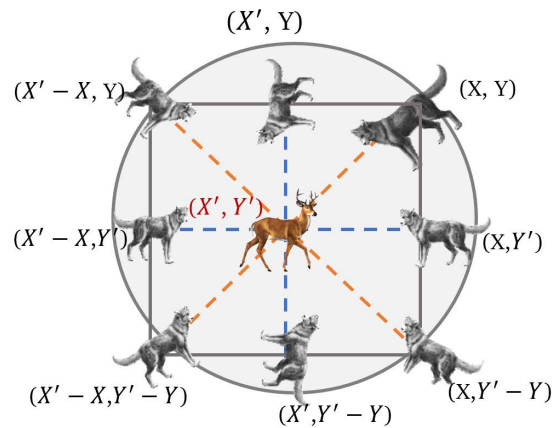


Figure 5. Grey wolves encircling the prey.

The alpha wolf, typically the pack leader, detects prey locations. The Alpha, Beta, and Delta wolves provide mathematical solutions for optimum prey location [26]:

$$\begin{aligned} D_\alpha &= |C_1 \cdot V_\alpha - X| \\ D_\beta &= |C_2 \cdot V_\beta - X| \\ D_\delta &= |C_3 \cdot V_\delta - X| \end{aligned} \quad (11)$$

$$\begin{aligned} V_1 &= V_\alpha - A_1 \cdot D_\alpha \\ V_2 &= V_\beta - A_2 \cdot D_\beta \\ V_3 &= V_\delta - A_3 \cdot D_\delta \end{aligned} \quad (12)$$

Next, as indicated by the following equation, the other wolves have to adjust their locations in accordance with the positions of the top three search agents (α ,

β , and δ):

$$V_{(i+1)} = \frac{V_1 + V_2 + V_3}{3} \tag{13}$$

The grey wolves surround a prey, stop moving, and attack it. The value of A decreases, resulting in a decrease in the prey’s position. The next grey wolf position is between its current and prey location.

3.5. Implementation

Figure 6 shows class diagram of the implementation. The class diagram, which shows properties like training_data, test_data, lstm_model, and gwo_optimizer for the main WPF class. Wind energy training and testing data are included in the dataset. The GWO technique is utilized for hyperparameter tuning, and the LSTM model is used for forecasting. Be aware that composition, also known as aggregation, is the relationship between classes. The composition relationship, shown by the diamond shape at the end of the association line, denotes a strong “whole-part” relationship between classes.

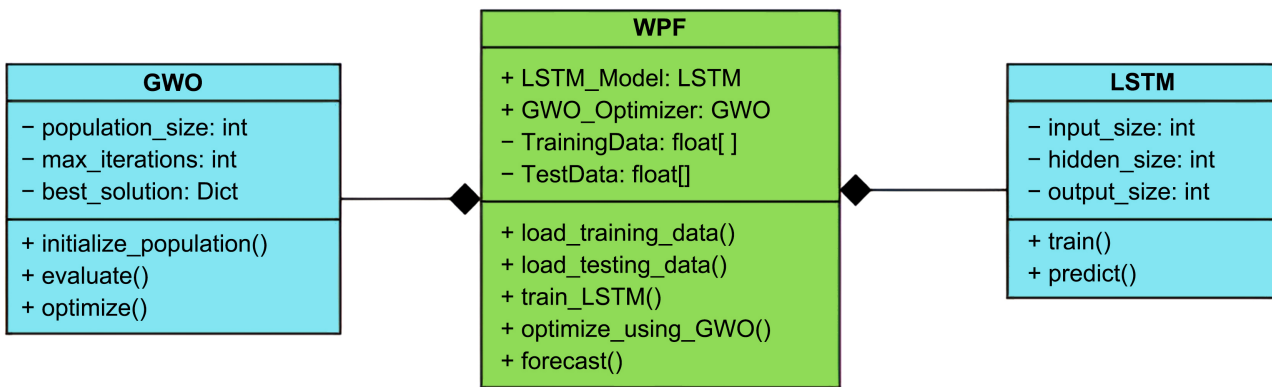


Figure 6. Class diagram.

For downloading the Python source code and full-merged dataset, go to the repository <https://github.com/dr-dos-ok/Wind-Power-Forecasting-onGithub.com>.

Figure 7 shows components of the project. The file C1.csv is the merged data. WPF usually merges NWP with historical data for providing accurate weather prediction. GWO.py is the implementation of GWO algorithm, while model.py provides the implementation of LSTM.



Figure 7. Project components.

4. Results and Discussion

KDE, or Kernel Density Estimation, is a non-parametric technique for estimating the probability density function of a random variable based on a sample of data points. KDE is particularly useful for visualizing the distribution of data in a continuous and smooth way, which is helpful when trying to understand the underlying distribution or when data points are sparse. **Figure 8** shows KDE for mod, dir, temp, and rh.

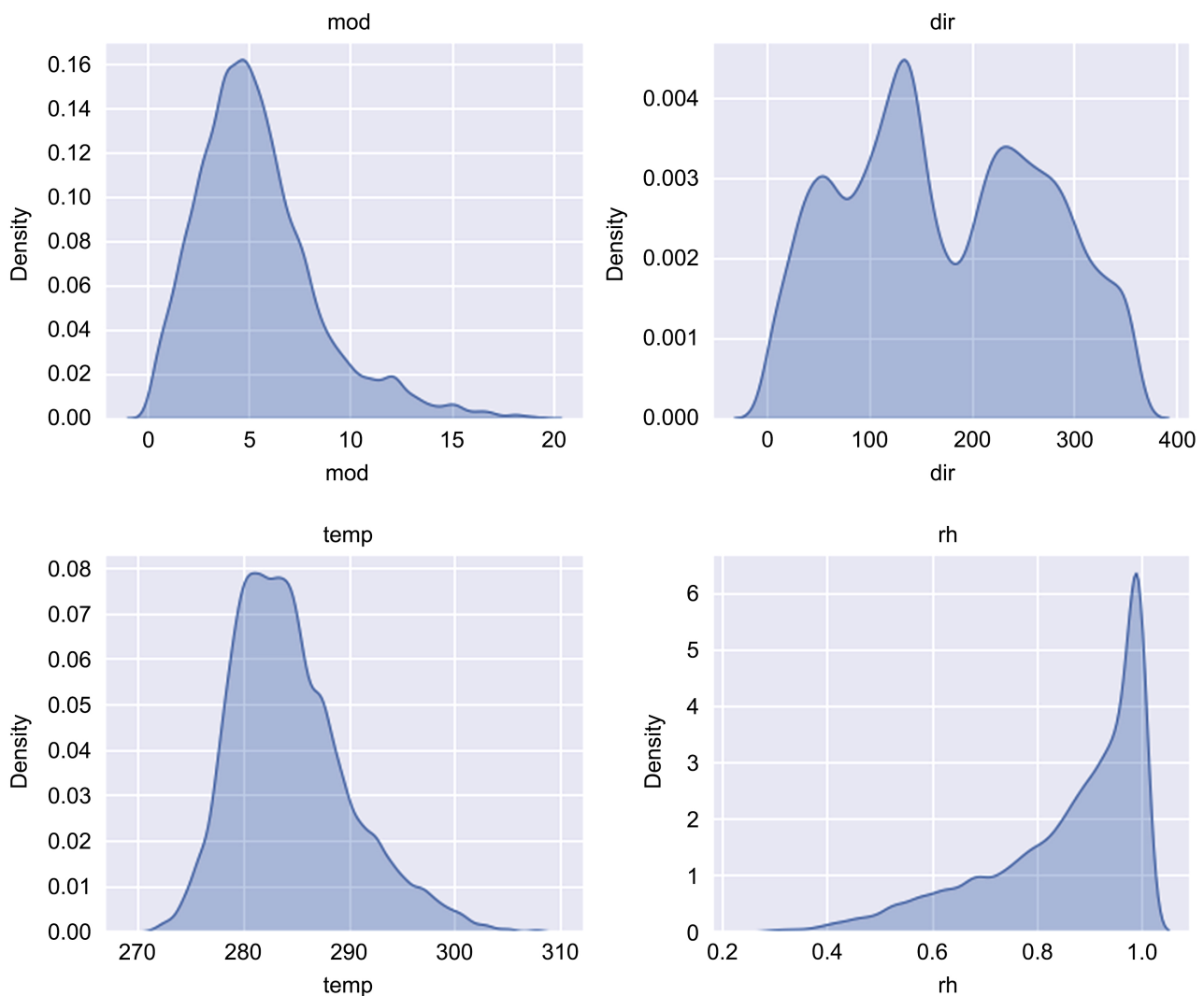


Figure 8. KDE for mod, dir, temp, and rh.

Figure 9 shows heatmap for all the features. The highest correlation is between speed and energy (0.9). Then the next higher correlation is between mod and energy (0.79). Direction and dir are moderately correlated (0.74). The same is said about speed and mod (0.77).

Figure 10 shows Windrose Axis for Wind Speed vs Wind Direction, while **Figure 11** shows Wind Speed vs Wind Energy.



Figure 9. Heatmap for all the features.

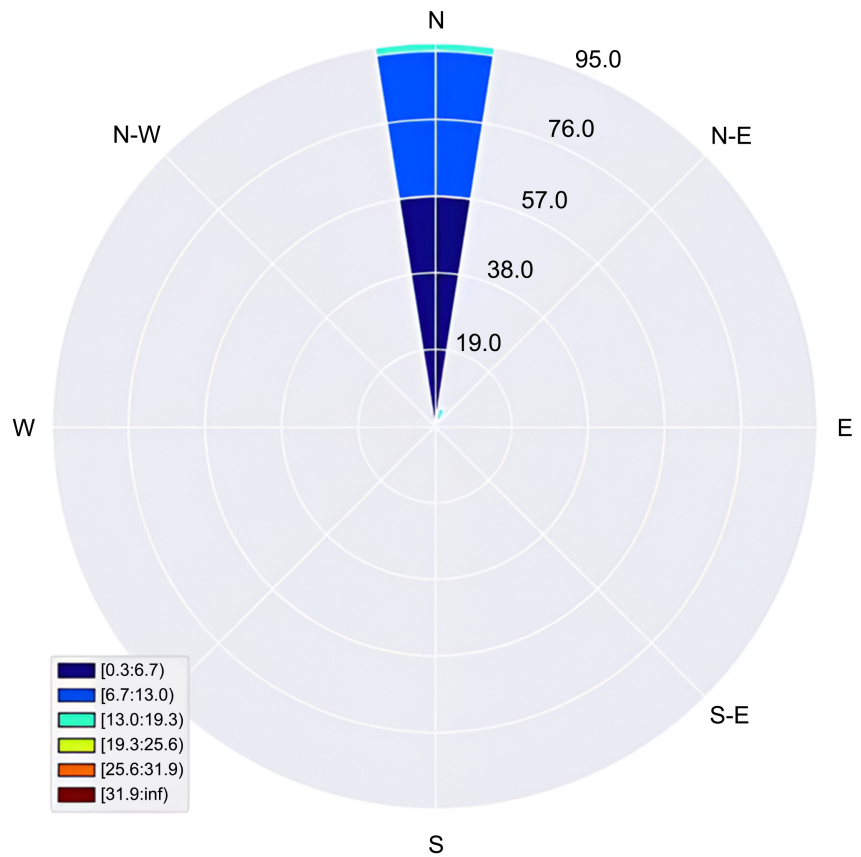


Figure 10. Wind speed vs wind direction.

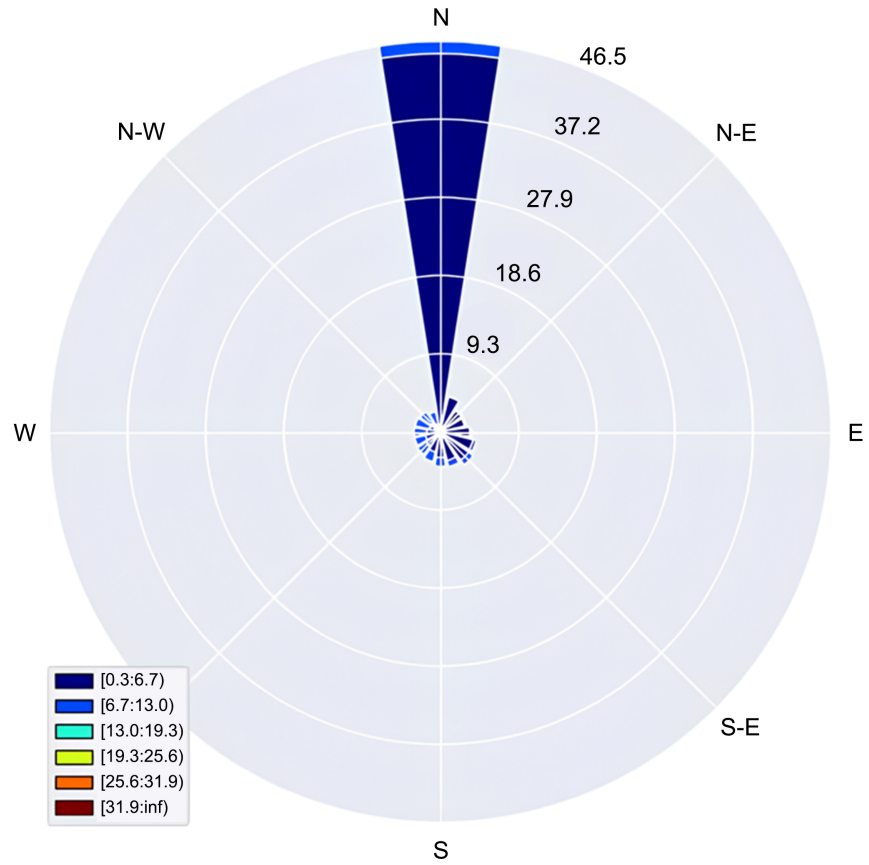


Figure 11. Wind speed vs wind energy.

Figure 12 depicts prediction plot. It is shifted for better comparing the actual and predicted values.

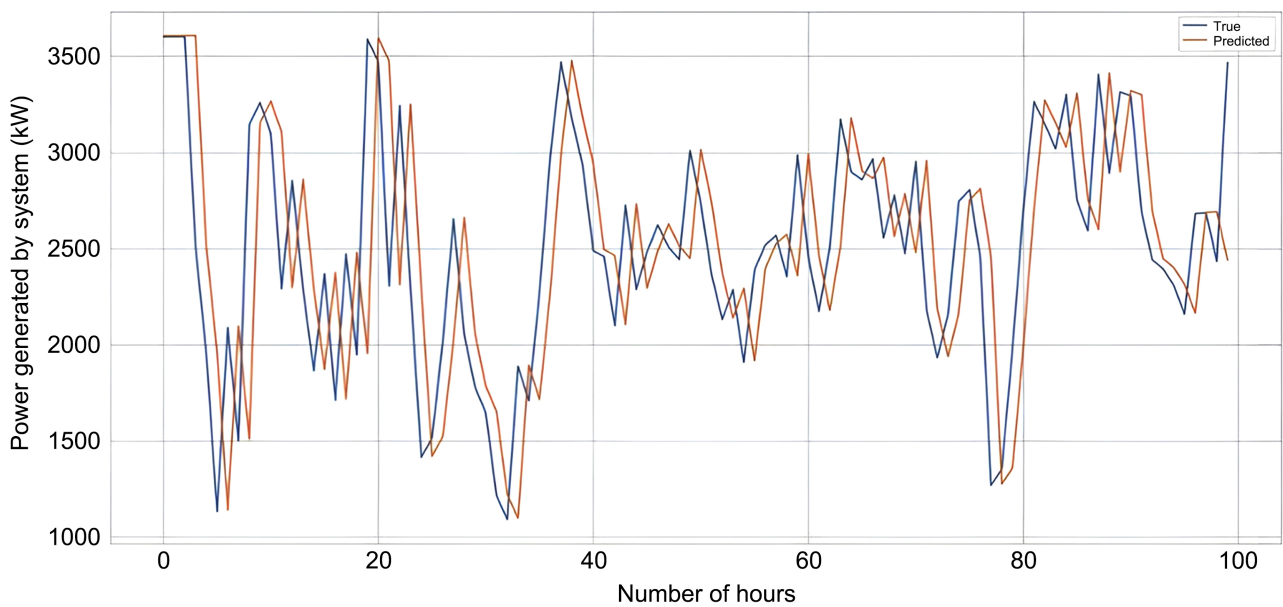


Figure 12. Prediction plot.

As shown in **Figure 13**, this time, it plots the first 20 values of the actual and predicted values. This plot is showing clearly the relation between predicted and actual values.

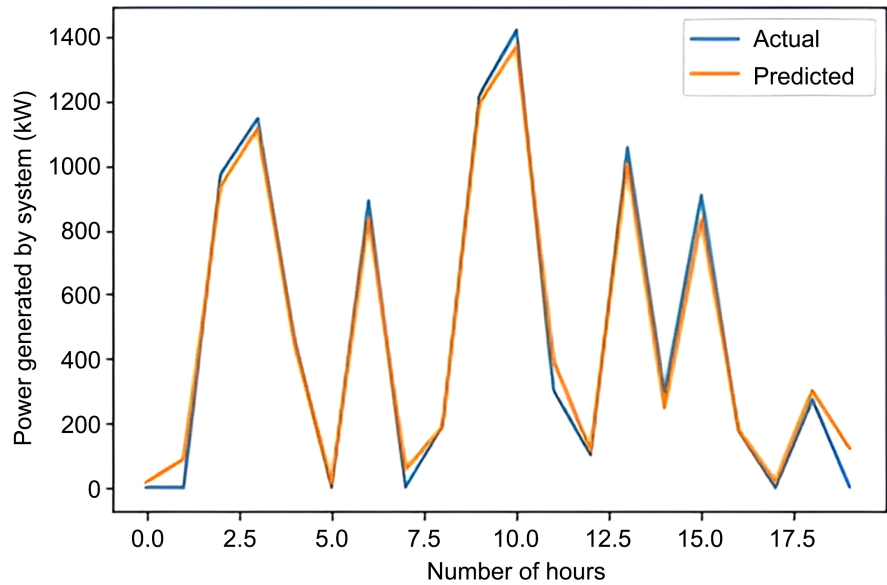


Figure 13. Plotting first 20 values of predicted vs actual values.

The authors use Mean Absolute Error (MAE), R-squared (R^2), and Mean absolute percentage error (MAPE) to measure the accuracy of their proposed regression model, focusing on average errors between predicted and actual values.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (14)$$

MAE is a reliable error metric, less sensitive to outliers, and is expressed in the same units as predicted and actual values, making it easier to understand errors.

R^2 is a statistical metric assessing the goodness of fit of a regression model, determining the proportion of variance in the dependent variable explained by independent variables.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (15)$$

It is also known as coefficient of determination with values ranging from 0 to 1, with 0 indicating no model explanation and 1 indicating all model explanation. A higher R-squared value indicates better model fit.

The coefficient of determination is a crucial metric in predicting electricity generation, as it measures how accurately a machine's predictions match the actual electricity generated. R-squared is a widely used evaluation metric for regression models, indicating a better fit. However, R-squared has limitations, such as being sensitive to predictors and not providing statistical significance or predictive power information.

MAPE is a statistical measure of prediction accuracy, expressed as a ratio using the following equation:

$$\text{MAPE} = 100 \frac{1}{n} \sum_{i=1}^n \left| \frac{F_i - A_i}{A_i} \right| \quad (16)$$

where A_t is the actual value at time t , and F_t is forecast value at the same time.

When MAPE is stated as a percentage, a lower number denotes superior performance or accuracy. It is frequently used to assess the accuracy of forecasts in demand planning and forecasting. It is sensitive to extreme numbers and becomes undefined when the actual values are 0, among other drawbacks.

The loss vs. epochs graph for the first 100 epochs is displayed in **Figure 14**. As can be observed, the curve exhibits a steep decline followed by oscillations as it slides below the 0.02 threshold, indicating an acceptable total loss.

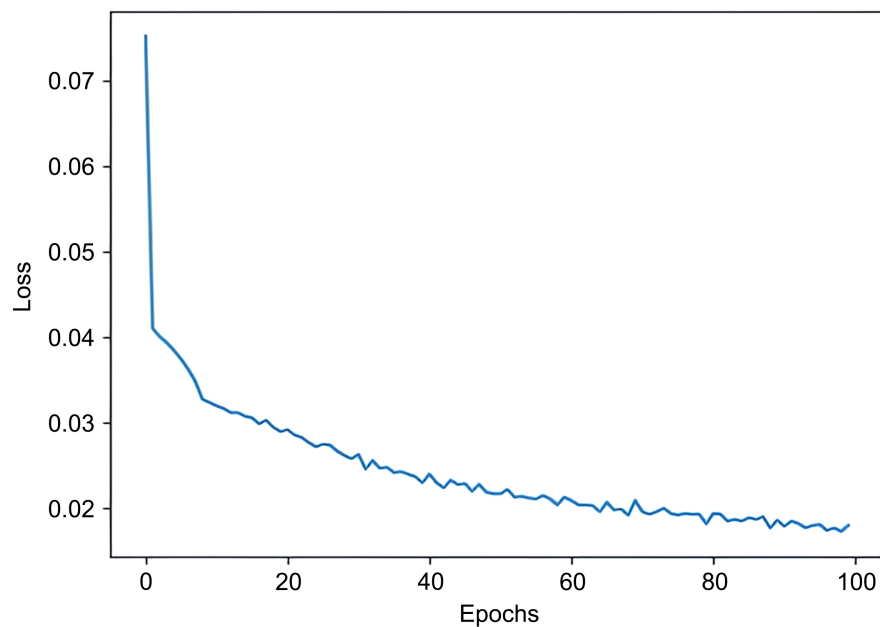


Figure 14. Loss vs. epochs.

As shown in **Figure 15**, the loss vs. MAPE curve is depicted for the first 100 epochs. As noticed, the curve has a sharp drop then it oscillates while sliding down to beneath a certain value which means that the overall value is acceptable.

This paper calculates the MAE giving 64.37996572 value. Also, this paper calculates the R^2 score giving 0.97279005691.

This outcome surpasses the study reported in [14], which had an R-squared value of 0.92. With data spanning four seasons and a 24-hour period, this earlier study used hybrid deep learning CNN, Radial Basis Function Neural Network (RBFNN), and double Gaussian function (DGF). There are two issues with this earlier work. First of all, it doesn't explain how to get rid of outliers and lessen their detrimental impact. Second, the outcome of forecasting performance when false data is present is not mentioned.

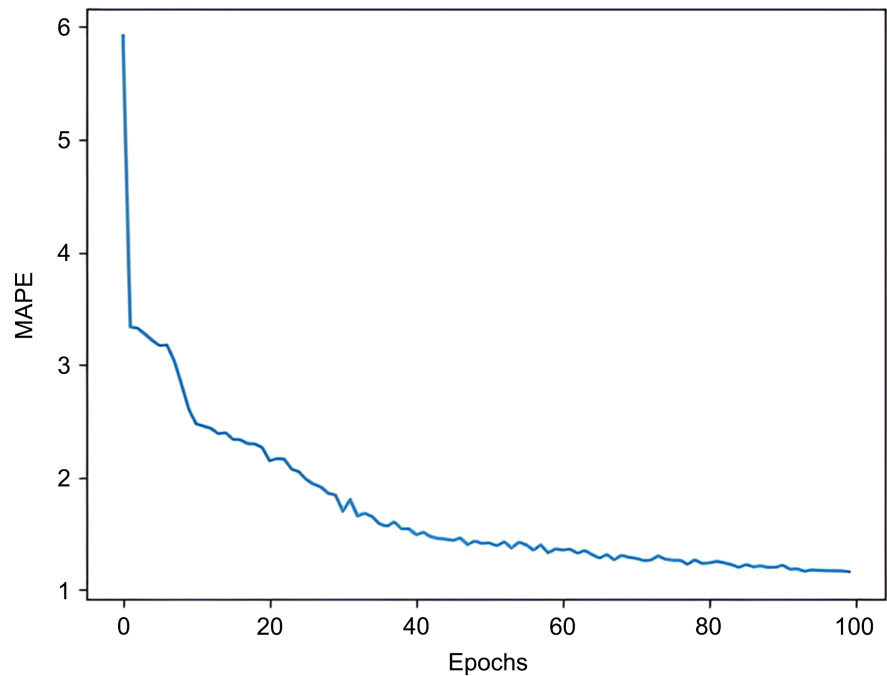


Figure 15. MAPE vs. epochs.

Not all datasets and models are comparable to the one used in this paper for various reasons. This paper focused on short-term wind power forecasting which is generally easier than long-term forecasting due to the higher predictability of wind patterns over shorter periods and the reduced complexity of influencing factors. Short-term predictions rely primarily on recent meteorological data, such as wind speed, direction, and air pressure, which are accurately captured by modern sensors and forecasting tools. This allows models like Grey Wolf Optimized LSTMs to effectively learn temporal patterns and produce precise forecasts. In contrast, long-term forecasting involves additional uncertainties, such as seasonal variations, climate dynamics, and land-use changes, which introduce greater complexity and reduce accuracy. Furthermore, long-term predictions are more prone to error accumulation and require broader modeling approaches, making them less precise. As a result, short-term forecasting is better suited for operational decisions like grid balancing, while long-term forecasting is typically used for strategic planning with a higher tolerance for uncertainty.

5. Conclusions and Future Work

The paper aims to improve wind power forecasting by accurately predicting electricity generation from wind turbines, reducing uncertainty in grid operations, and enhancing grid reliability. It integrates wind farm historical data, Numerical Weather Prediction data, k-Nearest Neighbors, K-means-Clustering, and LSTM for training and testing. The Grey Wolf Optimized Deep Learning model can explain 97.28% of wind power data variance, outperforming a reference study.

Future advancements include considering long-term wind power forecasting

which is more prone to error accumulation and requires broader modeling approaches.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Wang, Y., Hu, Q., Li, L., Foley, A.M. and Srinivasan, D. (2019) Approaches to Wind Power Curve Modeling: A Review and Discussion. *Renewable and Sustainable Energy Reviews*, **116**, Article ID: 109422. <https://doi.org/10.1016/j.rser.2019.109422>
- [2] Okumus, I. and Dinler, A. (2016) Current Status of Wind Energy Forecasting and a Hybrid Method for Hourly Predictions. *Energy Conversion and Management*, **123**, 362-371. <https://doi.org/10.1016/j.enconman.2016.06.053>
- [3] Zhang, L., Xie, Y., Ye, J., Xue, T., Cheng, J., Li, Z., et al. (2021) Intelligent Frequency Control Strategy Based on Reinforcement Learning of Multi-Objective Collaborative Reward Function. *Frontiers in Energy Research*, **9**, Article ID: 760525. <https://doi.org/10.3389/fenrg.2021.760525>
- [4] Yan, J., Liu, Y., Han, S., Wang, Y. and Feng, S. (2015) Reviews on Uncertainty Analysis of Wind Power Forecasting. *Renewable and Sustainable Energy Reviews*, **52**, 1322-1330. <https://doi.org/10.1016/j.rser.2015.07.197>
- [5] Zhao, Y., Ye, L., Wang, Z., Wu, L., Zhai, B., Lan, H., et al. (2019) Spatio-Temporal Markov Chain Model for Very-Short-Term Wind Power Forecasting. *The Journal of Engineering*, **2019**, 5018-5022. <https://doi.org/10.1049/joe.2018.9294>
- [6] Yang, N., Huang, Y., Hou, D., Liu, S., Ye, D., Dong, B., et al. (2019) Adaptive Non-parametric Kernel Density Estimation Approach for Joint Probability Density Function Modeling of Multiple Wind Farms. *Energies*, **12**, Article No. 1356. <https://doi.org/10.3390/en12071356>
- [7] Liu, H. and Chen, C. (2019) Data Processing Strategies in Wind Energy Forecasting Models and Applications: A Comprehensive Review. *Applied Energy*, **249**, 392-408. <https://doi.org/10.1016/j.apenergy.2019.04.188>
- [8] Zhou, D., Zuo, X. and Zhao, Z. (2022) Constructing a Large-Scale Urban Land Subsidence Prediction Method Based on Neural Network Algorithm from the Perspective of Multiple Factors. *Remote Sensing*, **14**, Article No. 1803. <https://doi.org/10.3390/rs14081803>
- [9] Zhang, Y., Li, Y. and Zhang, G. (2020) Short-Term Wind Power Forecasting Approach Based on Seq2Seq Model Using NWP Data. *Energy*, **213**, Article ID: 118371. <https://doi.org/10.1016/j.energy.2020.118371>
- [10] Zhang, J., Yan, J., Infield, D., Liu, Y. and Lien, F. (2019) Short-Term Forecasting and Uncertainty Analysis of Wind Turbine Power Based on Long Short-Term Memory Network and Gaussian Mixture Model. *Applied Energy*, **241**, 229-244. <https://doi.org/10.1016/j.apenergy.2019.03.044>
- [11] Shabbir, N., Jawad, M., Husev, O., Ur Rehman, A., Abid Gardezi, A., et al. (2022) Short-Term Wind Energy Forecasting Using Deep Learning-Based Predictive Analytics. *Computers, Materials & Continua*, **72**, 1017-1033. <https://doi.org/10.32604/cmc.2022.024576>
- [12] Li, L., Zhao, X., Tseng, M. and Tan, R.R. (2020) Short-Term Wind Power Forecasting Based on Support Vector Machine with Improved Dragonfly Algorithm. *Journal of Cleaner Production*, **242**, Article ID: 118447.

- <https://doi.org/10.1016/j.jclepro.2019.118447>
- [13] Tang, J. and Chien, Y. (2022) Research on Wind Power Short-Term Forecasting Method Based on Temporal Convolutional Neural Network and Variational Modal Decomposition. *Sensors*, **22**, Article No. 7414. <https://doi.org/10.3390/s22197414>
- [14] Hong, Y. and Rioflorida, C.L.P.P. (2019) A Hybrid Deep Learning-Based Neural Network for 24-H Ahead Wind Power Forecasting. *Applied Energy*, **250**, 530-539. <https://doi.org/10.1016/j.apenergy.2019.05.044>
- [15] Meka, R., Alaeddini, A. and Bhaganagar, K. (2021) A Robust Deep Learning Framework for Short-Term Wind Power Forecast of a Full-Scale Wind Farm Using Atmospheric Variables. *Energy*, **221**, Article ID: 119759. <https://doi.org/10.1016/j.energy.2021.119759>
- [16] Huang, S., Yan, C. and Qu, Y. (2023) Deep Learning Model-Transformer Based Wind Power Forecasting Approach. *Frontiers in Energy Research*, **10**, Article ID: 1055683. <https://doi.org/10.3389/fenrg.2022.1055683>
- [17] Chen, W., Qi, W., Li, Y., Zhang, J., Zhu, F., Xie, D., et al. (2021) Ultra-Short-Term Wind Power Prediction Based on Bidirectional Gated Recurrent Unit and Transfer Learning. *Frontiers in Energy Research*, **9**, Article ID: 808116. <https://doi.org/10.3389/fenrg.2021.808116>
- [18] Tarek, Z., Y. Shams, M., M. Elshewey, A., M. El-kenawy, E., Ibrahim, A., A. Abdelhamid, A., et al. (2023) Wind Power Prediction Based on Machine Learning and Deep Learning Models. *Computers, Materials & Continua*, **74**, 715-732. <https://doi.org/10.32604/cmc.2023.032533>
- [19] Yadav, A., Jha, C.K. and Sharan, A. (2020) Optimizing LSTM for Time Series Prediction in Indian Stock Market. *Procedia Computer Science*, **167**, 2091-2100. <https://doi.org/10.1016/j.procs.2020.03.257>
- [20] Xie, H., Zhang, L. and Lim, C.P. (2020) Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer. *IEEE Access*, **8**, 161519-161541. <https://doi.org/10.1109/access.2020.3021527>
- [21] Smagulova, K. and James, A.P. (2019) A Survey on LSTM Memristive Neural Network Architectures and Applications. *The European Physical Journal Special Topics*, **228**, 2313-2324. <https://doi.org/10.1140/epjst/e2019-900046-x>
- [22] Serrano-Pérez, J.d.J., Fernández-Anaya, G., Carrillo-Moreno, S. and Yu, W. (2021) New Results for Prediction of Chaotic Systems Using Deep Recurrent Neural Networks. *Neural Processing Letters*, **53**, 1579-1596. <https://doi.org/10.1007/s11063-021-10466-1>
- [23] Fan, C., Zhang, Y., Pan, Y., Li, X., Zhang, C., Yuan, R., et al. (2019) Multi-Horizon Time Series Forecasting with Temporal Attention Learning. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, 4-8 August 2019, 2527-2535. <https://doi.org/10.1145/3292500.3330662>
- [24] Sak, H., Senior, A. and Beaufays, F. (2014) Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. *Interspeech 2014*, Singapore, 14-18 September 2014, 338-342. <https://doi.org/10.21437/interspeech.2014-80>
- [25] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [26] Muro, C., Escobedo, R., Spector, L. and Coppinger, R.P. (2011) Wolf-Pack (*Canis lupus*) Hunting Strategies Emerge from Simple Rules in Computational Simulations. *Behavioural Processes*, **88**, 192-197. <https://doi.org/10.1016/j.beproc.2011.09.006>