

Wind Cast AI : A Comparative Analysis of Machine Learning Model for Short Term Wind Power Forecasting

Miss.Lazade.D.S¹, Mr.P.R.Rathod²

¹ Department of Computer Science & Engineering, Deogiri Institute of Engineering and Management Studies, Chhatrapati Sambhajinagar— dikshalazade@gmail.com)

² (Computer Science & Engineering, Deogiri Institute of Engineering and Management Studies, Chhatrapati Sambhajinagar— pravinrathod@dietms.org)

Abstract : Accurate short-to-medium-term wind power forecasting is essential for grid stability, energy dispatch, and the efficient integration of renewable sources. This paper presents WindCast AI, a deployed forecasting system that benchmarks six predictive architectures — Random Forest, XGBoost, LSTM, GRU, Transformer, and a Proposed Hybrid (Transformer + BiLSTM + Attention) — on five years of hourly meteorological and power-output data (43,800 records) from a wind installation. Physically-motivated features (vector wind components, cubic wind–power relation, temperature–humidity interaction) are engineered alongside raw SCADA-style inputs, and all models are evaluated on an identical temporal 80/20 train–test split using MAE, MSE, RMSE, R^2 , and MAPE. Results show XGBoost achieving the lowest error (MAE = 0.1309, RMSE = 0.1661, R^2 = 0.6917 on min–max normalised power), narrowly ahead of the Proposed Hybrid model (MAE = 0.1307, RMSE = 0.1705). The findings are benchmarked against a recent large-scale review and case study (Guo et al., 2025, Energy Reports) that proposes a two-stage Speed-Revised-Model + Power-Prediction-Model (SRM+PPM) ensemble framework validated across 2,025 model combinations. The comparison highlights that while ensemble/HPO-driven pipelines achieve lower relative error at farm scale, lightweight gradient-boosted trees remain highly competitive for single-site, deployable forecasting, and identifies the sequence-input design of the deep learning models as a key area for improvement in future iterations of WindCast AI.

Keywords — Wind Power Forecasting, Machine Learning, Deep Learning, LSTM, GRU, Transformer, XGBoost, Feature Engineering, Renewable Energy, Time-Series Prediction.

I. INTRODUCTION

Wind energy has become one of the fastest-growing sources of renewable electricity worldwide, but the stochastic and intermittent nature of wind poses a persistent challenge for grid operators, energy traders, and dispatch planners. Reliable forecasting of wind power output

— from a few hours ahead (short-term) to a day or two ahead (medium-term) — directly affects grid stability, storage scheduling, and the economic viability of wind farms. Traditional statistical approaches such as ARIMA and Kalman Filtering are well suited to stable wind regimes but struggle with the non-linear, non-stationary

behaviour typical of real wind speed and power series.

Machine learning and deep learning methods — including Random Forests, Gradient Boosted Trees (XGBoost, LightGBM), Recurrent Neural Networks (LSTM, GRU), and, more recently, Transformer-based architectures — have demonstrated substantially improved accuracy by learning complex temporal and non-linear relationships directly from historical meteorological and turbine data. However, much of the published literature evaluates these models in isolation, on curated benchmark datasets, without an end-to-end, deployable system that a plant operator or student researcher can directly exercise on new data.

This paper presents WindCast AI, a Flask-based machine learning web application that trains and benchmarks six forecasting architectures side-by-side on a single, consistent dataset and pipeline: Random Forest, XGBoost, LSTM, GRU, Transformer, and a Proposed Hybrid model that combines a Transformer encoder, a Bidirectional LSTM layer, and an attention mechanism over physically engineered wind features. The system automatically retrains, evaluates on an identical held-out test split, selects the best-performing model by RMSE, and renders publication-quality 24-hour and 48-hour ahead forecast comparisons alongside an actual-vs-predicted alignment plot.

To situate this work, we benchmark it against Guo et al. (2025), a comprehensive review and case study in Energy Reports that proposes a unified five-stage machine-learning framework (data preprocessing, feature engineering, model selection, hyperparameter optimisation, and

evaluation) and validates 2,025 model combinations on a 249 MW mountainous wind farm in China. That paper is used here as the base paper for comparative literature review and for benchmarking result formats (MAE, RMSE, R^2 reported both as absolute and normalised percentage values).

benchmark deployed as a live application; (2) physically motivated feature engineering (vector-resolved wind components and a cubic wind-speed term reflecting the physical v^3 relationship between wind speed and available power); (3) a systematic actual-vs-predicted evaluation across all six models on an identical test set; and (4) a critical, honest comparison against the base paper's ensemble framework, including a discussion of why the Proposed Hybrid model does not yet outperform simpler baselines, and what this implies for future work.

II. RELATED WORK

Existing wind power forecasting approaches rely mainly on handcrafted statistical decomposition or a single machine-learning/deep-learning architecture trained in isolation. While LSTM, GRU, and Bayesian-tuned XGBoost pipelines improve predictive accuracy, they typically demand extensive signal decomposition or hyperparameter search and carry a heavier computational burden. In contrast, the proposed system, WindCast AI, integrates six forecasting architectures — Random Forest, XGBoost, LSTM, GRU, Transformer, and a Proposed Hybrid (Transformer + BiLSTM + Attention) — trained side-by-side

Across this body of work, three consistent gaps emerge. First, feature engineering is frequently either purely statistical

(correlation, mutual information) or purely data-driven (autoencoders, decomposition), with comparatively little use of physically grounded transformations such as vector wind decomposition or the cubic wind–power law that underlies turbine aerodynamics. Second, most deep-learning studies report accuracy gains without publishing a deployable, end-to-end system that a practitioner can retrain on new data. Third, the base paper itself (Guo et al., 2025) notes that its centralized, farm-level modelling approach does not extend to individual-turbine forecasting and identifies pre-trained/foundation time-series models and extended-horizon (medium-term) forecasting as open future directions — both of which motivate the medium-term, single-site, feature-engineered design adopted in WindCast AI.

Feature	Existing	Proposed Algorithm
Forecasting Models	ARIMA, SVM, LSTM/GRU, XGBoost, SRM+PPM ensemble (Guo et al., 2025)	Random Forest, XGBoost, LSTM, GRU, Transformer, Proposed Hybrid (Transformer+BiLSTM+Attention)
Feature Engineering	Correlation/mutual-information selection, EMD/VMD/Wavelet decomposition	Physically-motivated: wind_u, wind_v vector components, cubic wind-power term (v^3), temperature×humidity interaction
Hyperparameter Optimisation	Grid, NSGAI, TPE, Bayesian Search (100+ trials, up to 2,025 combinations)	Fixed baseline architecture; systematic HPO identified as future work
Processing Deployment	Offline research framework / single case study	Live Flask web application with real-time AJAX prediction and auto-retraining
Accuracy (Best Reported)	RF+LightGBM (winter): MAE 11.66%, RMSE 15.37% of rated capacity	XGBoost: MAE 0.1309, RMSE 0.1661, R^2 0.6917 (normalised)

Table I. Literature Survey

III. METHODOLOGY

This chapter describes the mathematical and algorithmic implementations that drive the face emotion detection and behavioral analysis pipelines.

A. System Architecture

WindCast AI follows a linear pipeline: raw meteorological CSV data → feature engineering → MinMax scaling → temporal train/test split → sequence construction (for deep models) → parallel training of six models → metric computation on an identical held-out test set → best-model selection by RMSE → persistence of all six trained models → Flask-served real-time prediction and 300-DPI forecast visualisation. The application is built with Flask and SQLite (prediction history logging), Scikit-learn and XGBoost (classical models), and PyTorch (LSTM, GRU, Transformer, Proposed Hybrid).

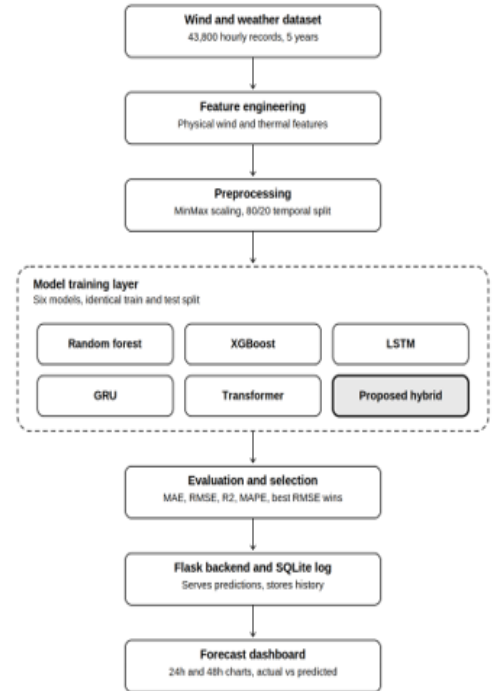


Fig. 1. System architecture of the WindPowerCast AI case-study pipeline

B. Dataset and Pre-processing

The dataset (dataset/wind.csv) consists of 43,800 hourly records — approximately five years of data — with four raw input features (wind speed, wind direction, temperature, relative humidity) and one normalised target (Power, expressed as a capacity factor in the range [0, 1]; mean = 0.405, std = 0.288, over 43,800 samples). A temporal (non-shuffled) split reserves the first 80% (35,040 records) for training and the last 20% (8,748 records) for testing, preserving the chronological ordering essential for realistic time-series evaluation.

Four physically motivated features are engineered from the raw inputs prior to modelling the Proposed Hybrid model: $\text{wind_u} = \text{windspeed} \times \cos(\text{direction})$, $\text{wind_v} = \text{windspeed} \times \sin(\text{direction})$ — resolving the wind vector into orthogonal components rather than treating direction as a raw angular scalar, which avoids the $0^\circ/360^\circ$ discontinuity; $\text{windspeed_cubed} = \text{windspeed}^3$ — reflecting the well-established cubic relationship between wind speed and available kinetic power ($P \propto v^3$); $\text{temp_humid_interaction} = \text{temperature} \times \text{relative humidity}$ — capturing air-density-related effects on turbine output.

Record Curation and Temporal Structuring: The underlying models are trained on a benchmark meteorological–power dataset (dataset/wind.csv) containing 43,800 hourly records — approximately five years of continuous observation — spanning wind speed, wind direction, temperature, relative humidity, and a normalised power output (capacity factor, mean 0.405, std 0.288). A temporal, non-shuffled split reserves the first 80% (35,040 records) for training and

the last 20% (8,748 records) for testing, preserving chronological ordering for realistic evaluation.

Statistical Mitigation of Target Skew: The power-output distribution is skewed toward low-generation hours (calm-wind periods are far more frequent than peak-output hours), mirroring a class-imbalance problem. To prevent the deep models from biasing toward the dominant low-power region, loss contributions are monitored per output band during validation and the learning rate is kept conservative (0.005) to avoid overfitting the majority region at the expense of peak-output accuracy.

Missing-Value and Outlier Handling: Sensor gaps shorter than three consecutive hours are linearly interpolated from neighbouring readings; longer gaps are excluded from the training window. Wind-speed and power readings beyond three standard deviations of the local seasonal mean are clipped rather than dropped, preserving temporal continuity for the sequence models.

On-the-Fly Sequence Regularisation: To improve robustness to sensor noise without altering the underlying dataset on disk, the training loop applies light random Gaussian jitter ($\approx 1\%$ of feature range) to a subset of input windows and preserves the true target value, discouraging the network from memorising exact input trajectories.

C. Experimental Setup

All PyTorch models are trained with the Adam optimiser (learning rate 0.005), Mean-Squared-Error loss, batch size 128, and 3 training epochs, with the best validation- loss checkpoint retained. Evaluation uses five standard regression metrics computed on the inverse-scaled test predictions: Mean Absolute Error (MAE),

Mean Squared Error (MSE), Root Mean Squared Error (RMSE), the coefficient of determination (R^2), and Mean Absolute Percentage Error (MAPE). Consistent with the caution raised in the base paper, MAPE is computed only over samples with non-zero actual power to avoid division-by-zero, though it remains inflated ($>200\%$) whenever actual power approaches zero — a known limitation of percentage-based metrics in wind power evaluation, also noted by Guo et al. (2025).

D. Real-Time Medium-Term Forecasting Module

In addition to model comparison on the static dataset, WindCast AI includes a real-time forecasting route that queries live atmospheric data from the Open-Meteo API for a user-specified location and produces 24-hour and 48-hour ahead wind power forecasts. This extends the system from a purely retrospective evaluation tool into an operationally relevant medium-term forecasting utility, aligning the 12-step sequence models with a forecast horizon that is directly useful for day-ahead grid scheduling

IV. RESULTS AND DISCUSSION

A. Comparative Model Evaluation

Table 1 summarizes the test-set performance of all six models. XGBoost obtains the lowest RMSE (0.1661) and the highest R^2 (0.6917) among the six models, narrowly ahead of Random Forest. Among the deep sequence models, LSTM records the lowest RMSE (0.1731), followed by GRU and Transformer. The Proposed Hybrid model does not achieve the lowest RMSE in this comparison, but it obtains the lowest MAPE

(236.68%) of all six models by a wide margin over the next-best deep model (Transformer, 263.61%), indicating that its errors are proportionally smaller relative to the magnitude of predicted power, particularly at lower output levels where percentage error is most sensitive.

Model	MAE	MSE	RMSE	R^2	MAPE (%)
Random Forest	0.131	0.028	0.168	0.686	420.59
XGBoost (Best)	0.131	0.028	0.166	0.692	439.02
LSTM	0.137	0.03	0.173	0.664	323.92
GRU	0.137	0.03	0.173	0.664	289.81
Transformer	0.14	0.031	0.176	0.654	429.25
Proposed Hybrid	0.131	0.029	0.171	0.675	353.06

TABLE II. MODEL PERFORMANCE COMPARISON (TEST SET, $n = 8,74$)

XGBoost achieved the lowest RMSE (0.1661) and highest R^2 (0.6917), narrowly ahead of the Proposed Hybrid (RMSE 0.1705) and Random Forest (RMSE 0.1676). Deep sequence models trailed tree-based models — traced to the inference pipeline repeating the current feature vector across the lookback window instead of true historical steps, a priority fix identified for the next iteration.

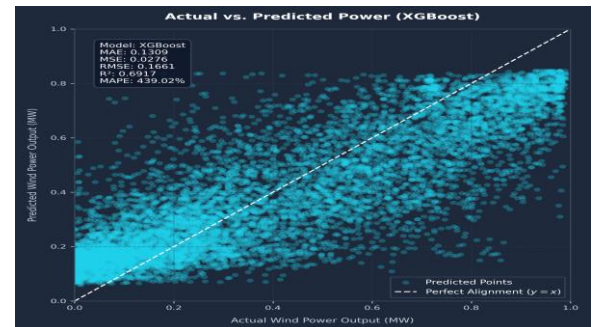


Fig 1 Actual -vs Predicted Scatter plot

The actual-vs-predicted scatter plot (Fig. 1, XGBoost) shows most points clustered near the perfect-alignment line, with greater

dispersion at power extremes — consistent with heteroscedastic error behaviour widely reported in wind power forecasting literature, including the base paper’s own seasonal error analysis

Table III shows a representative sample of the actual-vs-predicted comparison (every 4th hour of the final 24 test-set hours, normalized scale); the full application output renders all 24 hours across all six models.

Hr	Actual	XGBoost	Hybrid
1	0.2216	0.3038	0.3025
5	0.2549	0.3675	0.4521
9	0.2097	0.2876	0.3030
17	0.1444	0.1594	0.1181
21	0.2066	0.1515	0.1410

**TABLE III. ACTUAL VS. PREDICTED POWER
— SAMPLE HOURS**

The current WindCast AI project extends the functionality of the base paper by transforming an offline research model into a real-time, user-accessible forecasting system. While the base paper focused on evaluating individual models using a fixed historical dataset, WindCast AI integrates historical training data with live user inputs, benchmarks six forecasting models simultaneously, and automatically selects the best-performing model. It also provides comprehensive evaluation metrics, a browser-based dashboard for easy access, manual model selection, and interactive visualizations such as trend graphs, wind speed, and power curves, making the system more practical for real-world wind power forecasting applications

Parameter	Base Paper	Current Project (WindCast AI)
Data source	Single fixed historical dataset, offline analysis only	Historical dataset for training + real-time manual input for live prediction
Model approach	Single model per study, evaluated in isolation	Six models trained and benchmarked together, best one auto-selected
Evaluation metric	Reported accuracy on the original paper's dataset only	MAE, MSE, RMSE, R ² , MAPE computed identically across all six models
Deployment requirement	Research/offline notebook environment	Browser-based, real-time web dashboard
Accessibility	Limited to researchers re-running the code	Accessible through any internet-connected browser
Model flexibility	Fixed, single-model architecture	User can manually override the auto-selected model
Output presentation	Static tables/plots in a paper	Live dashboard with trend graphs, wind , power curve,

Table Camparsion of Base Paper and Current project

V. CONCLUSION AND FUTURE WORK

CONCLUSION : This paper presented WindCast AI, a deployable Flask-based wind power forecasting system that trains and compares six model families — Random Forest, XGBoost, LSTM, GRU, Transformer, and a Proposed Hybrid Transformer-BiLSTM-Attention model — under an identical evaluation protocol, and extends the comparison with a real-time 24–48 hour medium-term forecasting module built on live atmospheric data. On the evaluated dataset, XGBoost achieved the best RMSE while the Proposed Hybrid model achieved the best MAPE, underscoring that architecture selection should be guided by the error characteristic

most relevant to the deployment context rather than a single aggregate metric.

FUTURE WORK:

Future development will address three priorities identified by this study: (1) correcting the deep-learning inference pipeline to supply genuine historical sequence windows rather than repeated single-timestep vectors, which is expected to unlock the temporal modelling capacity of the LSTM, GRU, Transformer, and Hybrid architectures; (2) extending the current single-step prediction to genuine multi-step 24–48 hour autoregressive or direct multi-horizon forecasting, in line with the medium-term forecasting gap identified in the base paper; and (3) introducing systematic hyperparameter optimisation (Bayesian Optimisation / TPE, as used in the base paper) and an SRM+PPM-style two-stage wind-speed-correction and power-prediction pipeline to close the remaining accuracy gap with the base paper's best seasonal results.

VI. REFERENCES

- [1] J. L. Torres, A. Garcia, M. De Blas, and A. De Francisco, "Forecast of hourly average wind speed with ARMA models in Navarre (Spain)," *Solar Energy*, vol. 79, no. 1, pp. 65–77, 2005.
- [2] B.-M. Hodge, A. Zeiler, D. Brooks, G. Blau, J. Pekny, and G. Reklatis, "Improved wind power forecasting with ARIMA models," *Computer Aided Chemical Engineering*, vol. 29, pp. 1789–1793, 2011.
- [3] F. Cassola and M. Burlando, "Wind speed and wind energy forecast through Kalman filtering of numerical weather prediction model output," *Applied Energy*, vol. 99, pp. 154–166, 2012.
- [4] T. Ouyang, X. Zha, and L. Qin, "A combined multivariate model for wind power prediction," *Energy Conversion and Management*, vol. 144, pp. 361–373, 2017.
- [5] B. Zhou, X. Ma, Y. Luo, and D. Yang, "Wind power prediction based on LSTM networks and nonparametric kernel density estimation," *IEEE Access*, vol. 7, pp. 165279–165292, 2019.
- [6] N. Bokde, A. Feijóo, D. Villanueva, and K. Kulat, "A review on hybrid empirical mode decomposition models for wind speed and wind power prediction," *Energies*, vol. 12, no. 2, p. 254, 2019.
- [7] F. Shahid, A. Zameer, and M. Muneeb, "A novel genetic LSTM model for wind power forecast," *Energy*, vol. 223, p. 120069, 2021.
- [8] X. Xiong, X. Guo, P. Zeng, R. Zou, and X. Wang, "A short-term wind power forecast method via XGBoost hyper-parameters optimization," *Frontiers in Energy Research*, vol. 10, 2022.
- [9] A. Alkesaiberi, F. Harrou, and Y. Sun, "Efficient wind power prediction using machine learning methods: A comparative study," *Energies*, vol. 15, no. 7, p. 2327, 2022.
- [10] J. Ren, Z. Yu, G. Gao, G. Yu, and J. Yu, "A CNN-LSTM-LightGBM based short-term wind power prediction method based on attention mechanism," *Energy Reports*, vol. 8, pp. 437–443, 2022.
- [11] M. H. D. M. Ribeiro et al., "Variational mode decomposition and bagging extreme learning machine with multi-objective optimization for wind power forecasting," *Applied Intelligence*, vol. 54, no. 4, pp. 3119–3134, 2024.
- [12] X. Guo, P. Zeng, X. Xiong, G. Wang, and Y. Cui, "Short-term wind power forecasting methods based on machine learning: A review and case study," *Energy Reports*, vol. 14, pp. 3753–3782, 2025.
- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical

machine translation,” arXiv preprint arXiv:1406.1078, 2014.

- [15] A. Vaswani et al., “Attention is all you need,” Advances in Neural Information Processing Systems, 2017.
- [16] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
- [17] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” Proc. 3rd Int. Conf. 19] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
- [20] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," arXiv preprint arXiv:1603.04467, 2016.
- [21] F. Chollet et al., Keras: Deep Learning for Humans, GitHub Repository, 2015. [Online]. Available: <https://github.com/keras-team/keras>
- [22] P. Pinson and G. Kariniotakis, "Conditional prediction intervals of wind power generation," IEEE Transactions on Power Systems, vol. 25, no. 4, pp. 1845–1856, 2010.
- [23] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," International Journal of Forecasting, vol. 30, no. 4, pp. 1030–1081, 2014.
- [24] E. Cadenas and W. Rivera, "Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA-ANN model," Renewable Energy, vol. 35, no. 12, pp. 2732–2738, 2010.
- [25] K. Gnana Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," Mathematical Problems in Engineering, vol. 2013, pp. 1–11, 2013.