Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automatica Marina 2026

RBVM
Simposio CEA de Robótica, Bioingeniería, Visión por Computador y Automática Marina

Enhancing Affordable Wind Power Forecasting through Temporal Convolutional Networks: A Comparative Study with LSTM Architectures

Peñacoba Yagüe, Mario[a,*], Sierra García, Jesús Enrique[a], Santos Peñas, Matilde[b]

[a] *Departamento de Digitalización, Universidad de Burgos, Campus Río Vena, Avda. Cantabria, s/n, 09006 Burgos, España.*
[b] *Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Madrid, España.*

Enhancing Affordable Wind Power Forecasting through Temporal Convolutional Networks: A Comparative Study with LSTM Architectures

Abstract

The global transition toward low-carbon energy systems demands increasingly accurate tools to address the inherent variability of renewable resources. In particular, wind power forecasting plays a crucial role in improving energy planning, enhancing grid stability, and enabling cost-efficient operation of wind-based systems. At the same time, the democratization of clean energy technologies requires forecasting methods that are not only reliable, but also computationally efficient and suitable for deployment on affordable hardware platforms. In this context, this work evaluates the effectiveness of advanced deep learning architectures for short-term wind power forecasting, with a particular focus on Temporal Convolutional Networks (TCN). Building upon previous research employing Long Short-Term Memory (LSTM) networks on low-cost devices, this study provides a rigorous comparative analysis between both approaches. The evaluation examines forecasting accuracy and real-time performance through metrics such as MAE, RMSE, R^2, predictions per second, and inference time per sample. The results demonstrate that TCNs can significantly reduce inference time—achieving up to an order of magnitude faster prediction speed than LSTMs—while maintaining competitive accuracy across all scenarios and outperforming recurrent models in specific feature configurations. These findings highlight the suitability of convolution-based architectures for real-time forecasting and their potential integration into low-cost, edge-computing solutions for small-scale or distributed wind energy systems.

Keywords: Wind Power Forecasting, Renewable Energy, Temporal Convolutional Networks (TCN), Long Short–Term Memory (LSTM), Low-Cost Hardware, Energy Management.

1. Introducción

The global shift toward sustainable energy systems has intensified the need to reduce greenhouse gas emissions and increase the share of renewable energy sources in electricity generation. Among these, wind energy has become one of the most promising alternatives due to its abundance, technological maturity, and capacity to support large-scale decarbonization efforts (Behera et al, 2025). However, the inherent intermittency and variability of wind resources continue to pose significant challenges for grid integration, energy planning, and operational reliability. As a result, accurate short-term forecasting of wind turbine power output has emerged as a critical requirement for both large-scale wind farms and small, distributed generation systems (Liu et al, 2025).

Predictive models capable of anticipating rapid changes in wind conditions can significantly enhance energy scheduling, improve reserve allocation, reduce operating costs, and contribute to the stability of electrical systems. At the same time, the democratization of renewable energy technologies—driven by the increasing installation of small or personal wind turbines—requires forecasting solutions that are not only accurate but also computationally efficient and suitable for deployment on affordable embedded devices (Serrano et al, 2022). These constraints motivate the exploration of machine learning (ML) architectures that can deliver high predictive performance while maintaining low inference times, especially in edge-computing environments (Sierra-García & Santos, 2020).

Traditionally, recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have been widely adopted for time-series forecasting due to their ability to capture long-term temporal dependencies

*Autor para correspondencia: mpenacoba@ubu.es

(Lindemann et al, 2021). Recent studies have demonstrated that LSTM networks can achieve strong predictive accuracy even when executed on low-cost hardware such as the Raspberry Pi, enabling accessible and scalable approaches to energy management in small wind turbine systems (Buestán Andrade et al, 2023). Nevertheless, recurrent architectures present inherent limitations in training and inference speed, making them less suitable for real-time applications where latency is critical.

To address these challenges, this work investigates the use of Temporal Convolutional Networks (TCN) as an alternative deep learning architecture for wind power forecasting. TCNs leverage causal and dilated convolutions to model temporal dependencies without the sequential bottleneck of RNNs, providing substantial advantages in parallelization, computational efficiency, and training stability. These properties make TCNs particularly attractive for scenarios requiring rapid inference and deployment on resource-constrained devices (Wu et al, 2022).

The main contribution of this study is a comprehensive comparison between TCN and LSTM architectures applied to real Supervisory Control and Data Acquisition (SCADA) data from a wind turbine. Four combinations of input features—Theoretical Power Curve (TPC), Wind Speed (WS), Wind Direction (WD), and their permutations—are evaluated to assess how different information sources affect forecasting accuracy and computational performance. The analysis is conducted using standard error metrics (MAE, RMSE, R^2) and real-time performance indicators such as predictions per second and inference time per sample.

The results demonstrate that TCN models achieve significantly faster inference—up to one order of magnitude—while maintaining competitive accuracy across all scenarios and outperforming LSTM in specific input configurations. These findings highlight the potential of TCNs as a high-performance alternative for real-time renewable energy forecasting and reinforce the feasibility of incorporating advanced deep learning models into low-cost and edge-based hardware platforms.

The remainder of this paper is structured as follows. Section 2 describes the dataset, preprocessing methods, and neural network architectures. Section 3 presents the experimental setup and evaluation metrics. Section 4 provides a detailed comparison of the results obtained for the TCN and LSTM models. Section 5 discusses the implications of the findings for real-time wind energy forecasting and edge-computing applications. Finally, Section 6 concludes the study and outlines potential directions for future work

2. Machine Learning Architectures

2.1. Temporal Convolutional Network (TCN) Architecture

Temporal Convolutional Networks (TCNs) are deep learning architectures specifically designed for sequence modeling using causal and dilated convolutions. Unlike recurrent architectures, TCNs avoid sequential processing and rely entirely on convolutional operations, enabling them to capture long-term temporal dependencies while maintaining high computational efficiency (Telicko et al, 2025). Their parallel structure, combined with receptive fields that grow exponentially through dilation, makes them particularly suitable for real-time forecasting tasks and edge-computing environments.

The specific TCN architecture implemented in this work for wind power prediction is composed of the following components:

I. Input Layer:

The input sequence, consisting of combinations of Theoretical Power Curve (TPC), Wind Direction (WD), and Wind Speed (WS), is passed to the TCN model as a multivariate time series.

II. Dilated Causal Convolutional Blocks:

The core of the network is built from stacked convolutional blocks, each containing:

- Causal 1D convolutions, ensuring that predictions at time (t) depend only on present and past values;
- Dilated convolutions, which expand the receptive field exponentially, allowing the model to incorporate long-range temporal information without requiring deep recurrent layers;
- Residual connections, stabilizing training and allowing gradients to flow effectively across layers;
- Dropout layers, inserted within each block to mitigate overfitting by randomly deactivating convolutional filters during training.

Each convolutional block contains 64 filters, a kernel size of 3, and dilation factors increasing in powers of two across the stacked layers (1, 2, 4, 8…). This configuration enables the model to efficiently learn multi-scale temporal patterns in the SCADA data.

III. Fully Connected Layer:

The output of the final convolutional block is flattened and passed through a dense layer with a single unit. This layer aggregates the temporal features extracted by the TCN and generates the final power prediction value.

IV. Output Layer:

The final output corresponds to the predicted wind turbine power for the latest timestep in the input sequence.

A schematic representation of the TCN architecture used in this work is provided in Fig. 1 (Zuo et al, 2023).

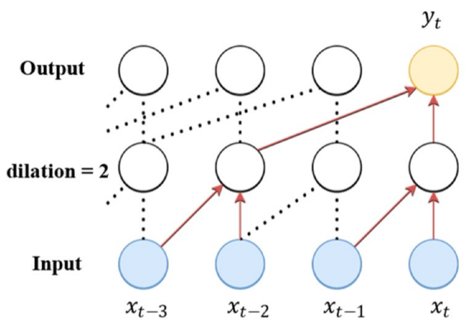

Fig. 1. Temporal Convoluutional Network Scheme.

2.2. LSTM Network Architecture

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to overcome the limitations of standard RNNs, particularly when learning long-term dependencies in sequential data. Through memory cells and gating mechanisms (input, output, and forget gates), LSTM networks can regulate the flow of information over time, making them well suited for time series forecasting problems (Torres et al, 2022).

The specific LSTM architecture used in this study follows the structure below, consistent with previous research on affordable wind power forecasting:

I. Input Layer:

The input sequence, composed of different combinations of TPC, WD, and WS variables, is fed into the first LSTM layer.

II. LSTM Layers:

The model consists of two stacked LSTM layers:

The first LSTM layer contains 64 units and is configured to return a full sequence, allowing temporal patterns to be propagated across layers. A dropout layer follows it to reduce overfitting.

The second LSTM layer also contains 64 units but returns only the final hidden state, condensing the learned temporal features into a single representation for prediction.

III. Fully Connected Layer:

A dense layer with a single neuron combines the extracted temporal features to compute the predicted wind turbine power. This layer acts as the regression head of the network.

IV. Output Layer:

The final output corresponds to the predicted active power at the next timestep.

A diagram illustrating the LSTM architecture used in this work is presented in Fig. 2 (Peñacoba et al, 2024).

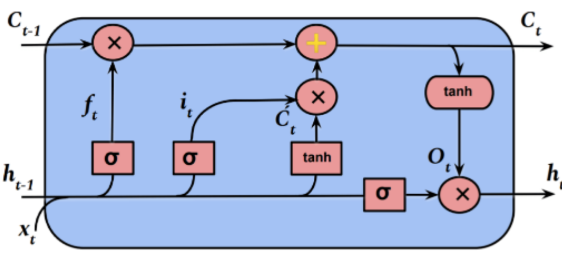

Fig. 2. LSTM Network architecture.

3. Model training

This section describes the dataset employed for the experiments, the preprocessing steps carried out prior to training, and the strategy followed to prepare the Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM) models for comparison. All experiments use the same wind turbine data in order to ensure a fair and consistent evaluation of both architectures.

3.1. Dataset

The data used in this study originate from a Supervisory Control and Data Acquisition (SCADA) system installed on an operational wind turbine (Buestán Andrade et al, 2024). The dataset includes the following variables:

I. Timestamp:

The measurement time, sampled every 10 minutes.

II. Active Power (kW):

The real electrical power delivered by the turbine at each timestamp. This variable is used as the ground-truth signal for training the forecasting model.

III. Wind Speed (m/s):

The wind velocity at hub height, directly influencing the kinetic energy available for conversion into electrical power.

IV. Theoretical Power Curve (kWh):

The expected power output derived from the turbine's manufacturer model, providing a reference estimation of performance under ideal aerodynamic conditions.

V. Wind Direction (°):

The azimuthal direction of the incoming wind, which affects turbine yaw control and energy capture efficiency.

These five signals form the basis for constructing the different input configurations evaluated in this work.

3.2. Pre-processing

Prior to training, the dataset underwent a series of preprocessing steps designed to standardize the input variables and ensure that both models received the same data representation. To systematically analyze the influence of different combinations of features, a dedicated function was implemented to generate the four scenarios evaluated in this study: TPC-WD, TPC-WS, WD-WS, and TPC-WD-WS.

Following established practices in wind power forecasting and previous studies using this dataset, all variables were normalized using standardization. Each feature was transformed by subtracting its mean and dividing by its standard deviation, ensuring that all input channels shared a comparable scale and facilitating model convergence.

The complete dataset was then partitioned into three subsets:

- 70% for training the neural networks,
- 20% for validation and hyperparameter tuning, and
- 10% for final testing and performance evaluation.

This data split ensures that the models are assessed on unseen samples while retaining sufficient data for both optimization and generalization analysis.

4. Experimental set up

4.1. Hardware

All experiments were executed on a Dell Vostro 5410 laptop computer, selected as a mid-range hardware platform to evaluate the computational performance of both the TCN and LSTM models under realistic conditions. The device is equipped with an 8th-generation Intel Core i7 processor, 8 GB of RAM, and operates under the Windows operating system (Peñacoba et al, 2026). With an approximate market price around EUR 700 (within a range of EUR 300–1000), this machine represents a standard, widely accessible computing environment that offers significantly more resources than low-cost embedded devices, while remaining far from high-performance server-grade hardware.

This choice of hardware allows us to explore the inference capabilities of both neural network architectures in a setting that is representative of typical personal workstations.

4.2. Software

The development and training of the TCN and LSTM models were carried out using a Python-based software environment. Both architectures were implemented using TensorFlow and Keras, which provide optimized tools for constructing deep learning models and managing training pipelines. Convolutional layers, dilated convolutions, and residual connections used in the TCN were implemented through Keras' high-level API, while recurrent layers for the LSTM architecture were also built using built-in Keras modules.

Data handling and normalization were performed using Scikit-learn, and all experiments were executed using Python 3.11. The use of standard open-source libraries ensures full reproducibility and facilitates future extensions or comparisons.

To enhance stability during training, both models employed the Adam optimizer, dropout regularization, and early stopping based on validation loss. The codebase was executed in the same environment for both architectures to ensure a fair comparison of training and inference performance.

4.3. Metrics

Two categories of metrics were considered in this study: forecasting accuracy and computational performance (Peñacoba et al, 2024).

To evaluate predictive quality, the following indicators were computed:

I. Mean Absolute Error (MAE):

Reflects the average magnitude of the deviations between real and predicted values (Equation 1).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \bar{y}_i| \tag{1}$$

II. Root Mean Square Error (RMSE):

Emphasizes larger errors by computing the square root of the mean squared deviations (Equation 2).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{2}$$

III. Coefficient of Determination (R^2):

Measures the proportion of variance in the ground-truth power signal explained by the model (Equation 3).

$$R^2 = 1 - \frac{\sum(y_i - \bar{y})^2}{(y_i - \bar{y}_i)^2} \tag{3}$$

These metrics allow for a comprehensive assessment of how accurately each model captures the underlying dynamics of wind turbine power output.

To analyze the real-time suitability of both architectures, two indicators were measured:

I. Predictions per second:

Number of forecasting operations the model can perform in one second, computed from the duration of each inference batch.

II. Inference time per sample:

Average time required to generate a single prediction.

Given an inference batch time T_i and fixed batch size, predictions per second are computed as shown in Equation 4.

$$Predictions = \frac{N_{batch}}{T_i} \tag{4}$$

The overall performance across multiple repetitions is then obtained by averaging results over all batches (Equation 5).

$$\overline{Predictions} = \frac{1}{N_{batch}}\sum_{i=1}^{N_{batch}} Predictions(i) \tag{5}$$

These indicators provide insights into the computational efficiency of both TCN and LSTM models, offering a practical perspective on their applicability for real-time wind power forecasting.

5. Results

5.1. Forecasting accuracy across all input scenarios

Table 1 summarizes the accuracy metrics obtained for both architectures (LSTM and TCN) under the four input-feature configurations evaluated. The results reveal a clear dependence of model performance on the selected input variables, as well as notable differences between the two neural network approaches.

TABLE I. PERFORMANCE METRICS FOR LSTM AND TCN

Performance Metrics for LSTM and TCN Across All Scenarios				
Scenario	***Method***	***MAE***	***RMSE***	***R^2***
TPC-WD	LSTM	0.0585	0.1039	0.9343
	TCN	0.0499	0.080	0.9608
TPC-WS	LSTM	0.0334	0.0608	0.9775
	TCN	0.0605	0.0926	0.9478
WD-WS	LSTM	0.0361	0.0618	0.9768
	TCN	0.0598	0.0936	0.9467
TPC-WD-WS	LSTM	0.0488	0.0762	0.9647
	TCN	0.0481	0.0785	0.9626

In the TPC–WD scenario, the TCN achieves the best overall accuracy among all experiments, reaching a MAE of 0.0499 and RMSE of 0.080, outperforming the LSTM model (MAE = 0.0585, RMSE = 0.1039). The corresponding R^2 of 0.9608 confirms that the TCN is able to capture the dynamics of wind direction–dependent fluctuations more effectively than the recurrent architecture in this configuration.

However, the trend reverses in the TPC–WS and WD–WS scenarios, where the LSTM model provides consistently lower errors. In both cases, LSTM achieves R^2 values exceeding 0.976, whereas the TCN drops to the 0.946–0.948 range. These results suggest that LSTM networks may be better suited for modeling dependencies dominated by wind speed, which often exhibits strong temporal continuity.

When the full set of features (TPC–WD–WS) is used, both architectures perform similarly. The TCN slightly improves the MAE (0.0481 vs. 0.0488), whereas LSTM obtains a marginally better RMSE (0.0762 vs. 0.0785) and R^2 (0.9647 vs. 0.9626). This indicates that when comprehensive information is available, both approaches converge to comparable predictive capability.

Overall, these results show that TCNs can outperform LSTMs in scenarios requiring the modeling of more complex, direction-dependent temporal patterns, while LSTMs maintain an advantage when forecasting is primarily driven by wind speed.

5.2. Time-series comparison: Real vs Predicted Signals

Figures 3–6 display the temporal comparison between the measured turbine power output and the predictions generated by both models for the four feature combinations. In all scenarios, both architectures successfully reproduce the characteristic intermittent behavior of the wind turbine, including sharp transitions, high-power plateaus, and low-generation periods.

In the TPC–WD configuration, the TCN provides a visibly tighter fit to the real signal, particularly during steep rising or falling edges, which aligns with its superior MAE and RMSE values. The LSTM also follows the general structure of the data but exhibits larger deviations during rapid changes (Fig.3).

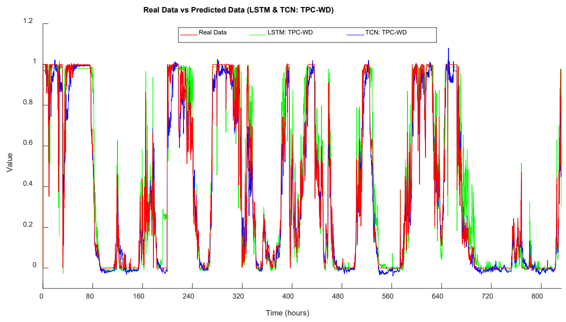

Fig. 3. Real Data vs LSTM & TCN Predicted Data (TPC-WD).

For the TPC–WS (Fig. 4) and WD–WS (Fig. 5) cases, the opposite behavior is observed: the LSTM predictions adhere more closely to the real data, capturing peak values with less overshooting than the TCN. This visual trend is consistent with the higher R^2 values obtained by the recurrent model in these scenarios.

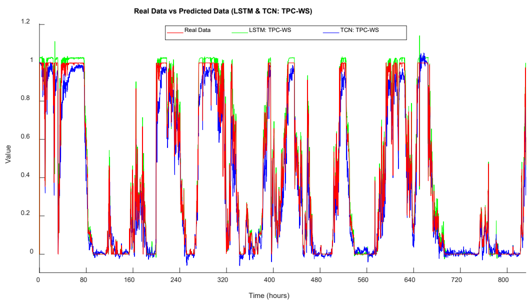

Fig. 4. Real Data vs LSTM & TCN Predicted Data (TPC-WS).

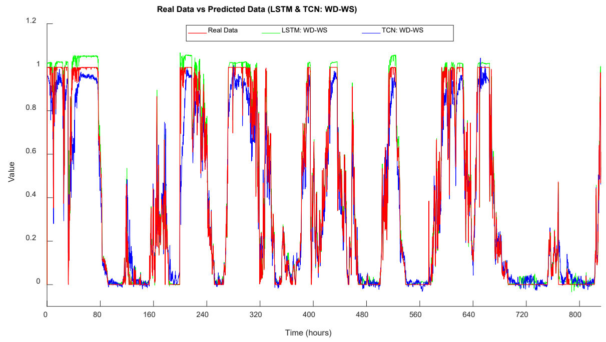

Fig. 5. Real Data vs LSTM & TCN Predicted Data (WD-WS).

When all three variables are combined (TPC–WD–WS), the difference between the models becomes less pronounced. Both curves align well with the real signal, and discrepancies are mostly confined to brief fluctuations near peak transitions (Fig. 6).

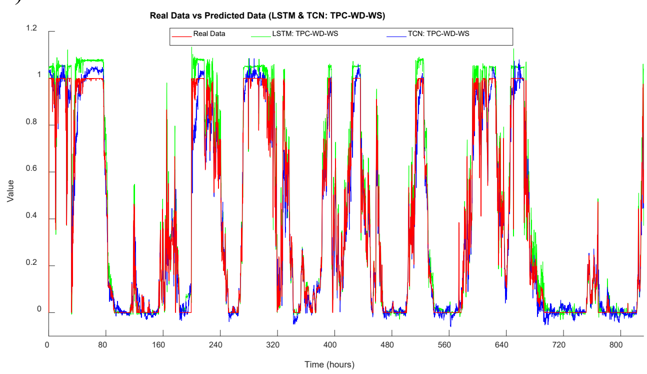

Fig. 6. Real Data vs LSTM & TCN Predicted Data (TPC-WD-WS).

These figures confirm that the visual behavior of predictions is fully aligned with the quantitative accuracy metrics reported in Table 1.

5.3. Error distributions

The error histograms shown in Fig. 7 provide further insight into the models' behavior. For each input scenario, the LSTM and TCN error distributions are compared directly.

The TPC–WD histogram displays a sharper, more concentrated error distribution for the TCN, reflecting its superior performance in this configuration. In contrast, the LSTM shows a slightly wider spread, indicating larger deviations around the mean error.

Conversely, in the TPC–WS and WD–WS cases, the LSTM histograms are more peaked and narrow than those of the TCN, consistent with the lower MAE and RMSE values of the recurrent model. The TCN distributions in these scenarios show heavier tails, suggesting that the convolutional architecture occasionally produces larger instantaneous errors.

In the combined-feature case (TPC–WD–WS), both models exhibit very similar distributions, with only small differences in tail behavior. This reinforces the conclusion that, when provided with full input information, both architectures achieve comparable accuracy.

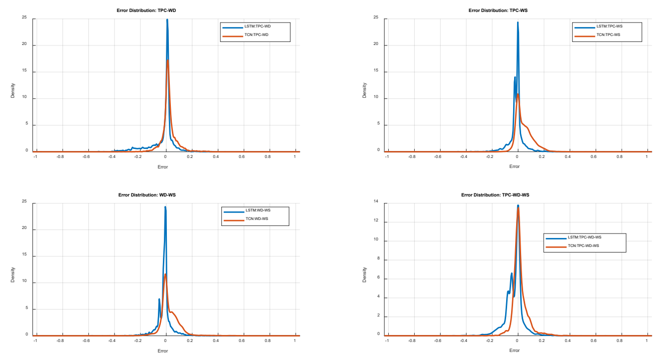

Fig. 7. Error Distributions.

5.4. Inference speed

The last part of the evaluation focuses on inference speed. Table 2 presents the mean number of predictions per second and the average inference time per sample for each scenario.

TABLE II. MEAN PREDICTIONS PER SECOND AND INFERENCE

Mean Prediction per Second and Inference			
Scenario	***Method***	***Mean Predictions per Second***	***Mean Inference Time per Sample (s)***
TPC-WD	LSTM	336.31	2.97
	TCN	3049.7	0.33
TPC-WS	LSTM	1197.67	0.84
	TCN	3219.95	0.31
WD-WS	LSTM	1224.65	0.8166
	TCN	3327.78	0.3
TPC-WD-WS	LSTM	985.82	1.01
	TCN	3221.52	0.31

The differences between the architectures are substantial. Across all feature combinations, the TCN achieves between 3,000 and 3,330 predictions per second, whereas the LSTM model ranges from 336 to 1,224 predictions per second. This represents an improvement of approximately one order of magnitude in computational performance.

Similarly, the mean inference time per sample decreases dramatically with the TCN, reaching values around 0.30–0.33 seconds, compared with 0.84–2.97 seconds for the LSTM.

These results highlight the key advantage of TCNs: they deliver rapid inference due to their fully convolutional design, enabling parallel computation and eliminating the sequential bottleneck inherent to recurrent networks.

6. Conclusions and future work

This study has presented a comprehensive comparison between Temporal Convolutional Networks (TCN) and Long Short-Term Memory (LSTM) architectures for short-term wind power forecasting using real SCADA data. The results demonstrate that both models are capable of accurately reproducing the dynamic behavior of the wind turbine, although their performance varies depending on the chosen input features. TCN achieves superior accuracy in scenarios where wind direction plays a significant role, while LSTM remains more effective when wind speed dominates the forecasting task. Most notably, the TCN provides a substantial improvement in computational efficiency, offering up to an order of magnitude faster inference times across all experiments. These findings position TCN as a compelling alternative for real-time or resource-constrained forecasting applications and highlight the importance of selecting model architectures according to the characteristics of the input data.

Future research will focus on extending this analysis in two directions. First, additional hybrid or attention-based architectures may be investigated to combine the strengths of convolutional and recurrent models, potentially improving forecasting robustness under varying meteorological conditions. Second, incorporating exogenous variables—such as atmospheric pressure, turbulence intensity, or numerical weather predictions—could further enhance model accuracy.

Referencias

Behera, U. S., Sangwai, J. S., & Byun, H. S. (2025). A comprehensive review on the recent advances in applications of nanofluids for effective utilization of renewable energy. Renewable and Sustainable Energy Reviews, 207, 114901.

Liu, S., Zhang, L., Lu, J., Zhang, X., Wang, K., Gan, Z., ... & Wang, H. (2025). Advances in urban wind resource development and wind energy harvesters. Renewable and Sustainable Energy Reviews, 207, 114943.

Serrano, C., Sierra-Garcia, J. E., & Santos, M. (2022). Hybrid optimized fuzzy pitch controller of a floating wind turbine with fatigue analysis. Journal of Marine Science and Engineering, 10(11), 1769.

Sierra-García, J. E., & Santos, M. (2020). Exploring reward strategies for wind turbine pitch control by reinforcement learning. Applied Sciences, 10(21), 7462.

Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. Computers in Industry, 131, 103498.

Buestán-Andrade, P. A., Santos, M., Sierra-García, J. E., & Pazmiño-Piedra, J. P. (2023, August). Comparison of LSTM, GRU and transformer neural network architecture for prediction of wind turbine variables. In International conference on soft computing models in industrial and environmental applications (pp. 334-343). Cham: Springer Nature Switzerland.

Wu, Z., Luo, G., Yang, Z., Guo, Y., Li, K., & Xue, Y. (2022). A comprehensive review on deep learning approaches in wind forecasting applications. CAAI Transactions on Intelligence Technology, 7(2), 129-143.

Telicko, J., Krumins, A., & Nikitenko, A. (2025). Development and Evaluation of Neural Network Architectures for Model Predictive Control of Building Thermal Systems. Buildings, 15(15), 2702.

Zuo, C., Wang, J., Liu, M., Deng, S., & Wang, Q. (2023). An ensemble framework for short-term load forecasting based on TimesNet and TCN. Energies, 16(14), 5330.

Peñacoba, M., Buestán, P., Sierra-García, J. E., Santos, M., & Ruano, A. (2024, October). Affordable wind power forecasting: Implementing LSTM Networks on low-cost hardware platforms for enhanced energy management. In International Conference on Soft Computing Models in Industrial and Environmental Applications (pp. 270-278). Cham: Springer Nature Switzerland.

Torres, J. F., Martínez-Álvarez, F., & Troncoso, A. (2022). A deep LSTM network for the Spanish electricity consumption forecasting. Neural Computing and Applications, 34(13), 10533-10545.

Buestán-Andrade, P. A., Peñacoba-Yagüe, M., Sierra-García, J. E., & Santos, M. (2024). Wind power forecasting with machine learning algorithms in low-cost devices. Electronics, 13(8), 1541.

Yagüe, M. P., García, J. E. S., & Penas, M. S. (2026). Human-intelligent trajectory optimization for robotic manipulators with hybrid PSO-PS algorithm. Advanced Engineering Informatics, 69, 103941.