Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

WiseLSTM: a depthwise CNN--LSTM framework with metaheuristic optimization for offshore wind power forecasting

Joaquín Toro
Pontificia Universidad Católica de Valparaíso
Tania Roa
Universidad Adolfo Ibáñez
Andrés Aránguiz
Federico Santa María Technical University
José Garcia
Pontificia Universidad Católica de Valparaíso
El-Ghazali Talbi
Université de Lille
Marcelo Becerra-Rozas
Pontificia Universidad Católica de Valparaíso
Emanuel Vega
emanuel.vega@pucv.cl

Pontificia Universidad Católica de Valparaíso

Article

Keywords: Wind power forecasting, Deep learning, Depthwise separable convolution, CNN--LSTM, Metaheuristic optimization, Green hydrogen

Posted Date: November 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11254587/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

WiseLSTM: a depthwise CNN–LSTM framework with metaheuristic optimization for offshore wind power forecasting

Joaquín Toro[1], Tania Roa[4], Andrés Aránguiz[3], José Garcia[2], El-Ghazali Talbi[5], Marcelo Becerra-Rozas[1*], Emanuel Vega[1*]

[1]Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2241, Valparaíso, 2362807, Chile.
[2]Escuela de Ingeniería de Construcción y Transporte, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2147, Valparaíso, 2362804, Chile.
[3]Departamento de Mecánica, Universidad Técnica Federico Santa María, Av. Federico Sta. María 6090, Quilpué, Chile.
[4]Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Padre Hurtado 750, Viña del Mar, 2520000, Chile.
[5]University of Lille & INRIA, UMR 9189 CRIStAL, Lille, France.

*Corresponding author(s). E-mail(s): marcelo.becerra@pucv.cl; emanuel.vega@pucv.cl;
Contributing authors: joaquin.toro.g@mail.pucv.cl; tania.roa@uai.cl; andres.aranguiz@usm.cl; jose.garcia@pucv.cl; el-ghazali.talbi@univ-lille.fr;

Abstract

Reliable one-hour-ahead wind-power forecasting can support scheduling and energy-management decisions in renewable-to-hydrogen systems. We propose WiseLSTM, a hybrid architecture combining two depthwise-separable temporal convolutional blocks with unidirectional Long Short-Term Memory (LSTM) modeling. Using 40 years of hourly ERA5-derived meteorological conditions and model-derived wind-farm power for Baie de Saint-Brieuc, evaluated with a five-fold blocked repeated-holdout protocol, WiseLSTM achieves an RMSE of $\mathbf{33.91 \pm 3.70}$ MW, MAE of $\mathbf{20.40 \pm 3.00}$ MW, and $\boldsymbol{R^2 = 0.9657 \pm 0.0076}$.

It reduces RMSE by approximately 5.0% relative to CNN–LSTM and 8.5% relative to one-hour persistence. At matched channel widths, the depthwise front-end uses 65.8% fewer convolutional kernel weights than standard convolutions. Particle Swarm Optimization (PSO) and Whale Optimization Algorithm (WOA) are further used to explore five mixed hyperparameters. The WOA-selected configuration achieves an RMSE of $\mathbf{33.67 \pm 1.24}$ MW and reduces the cross-fold RMSE standard deviation by 66.5% relative to untuned WiseLSTM. These results highlight the joint importance of efficient local feature extraction, recurrent temporal modeling, and configuration stability for offshore wind-power forecasting.

Keywords: Wind power forecasting, Deep learning, Depthwise separable convolution, CNN–LSTM, Metaheuristic optimization, Green hydrogen

1 Introduction

The transition toward low-carbon energy systems has increased interest in green hydrogen as an energy carrier for storage, industrial processes, and sector coupling. Hydrogen produced through water electrolysis powered by renewable energy can contribute to this transition, but the operation of such systems depends strongly on the availability and predictability of the renewable input [1–3]. Wind generation is particularly relevant because of its deployment potential and high capacity factors, yet its stochastic and non-stationary behavior introduces uncertainty into short-term operational planning. Accurate wind-power forecasts can therefore provide anticipatory information for energy allocation, electrolyzer scheduling, and power-to-hydrogen management [4, 5]. The present study focuses on this upstream forecasting component rather than on direct optimization of the hydrogen-production process.

Machine-learning and deep-learning methods have substantially expanded the range of models available for renewable-energy forecasting [6, 7]. Recurrent architectures such as LSTM networks are widely used to represent sequential dependencies, while convolutional networks are effective at extracting local temporal structures. Hybrid CNN–LSTM models combine these capabilities by first transforming short-scale patterns and then modeling their temporal evolution [8]. More recently, attention-based and transformer-based models have also shown strong forecasting performance [9, 10]. These developments reinforce an important design question: improved forecasting does not necessarily require increasing architectural complexity if local feature extraction and temporal modeling can be organized more efficiently.

Conventional temporal convolutions may become parameter-intensive as the number of input and output channels grows. Depthwise separable convolution offers an alternative factorization in which channel-wise temporal filtering is followed by pointwise channel mixing [11]. At a fixed channel width, this reduces the number of convolutional kernel weights while preserving a flexible feature-transformation stage. Such a design is attractive for wind forecasting because local fluctuations, ramps, and recurrent temporal structure coexist within the same input sequence.

A second challenge concerns model configuration. Deep-learning forecasting models remain sensitive to architectural and training hyperparameters, including hidden

dimension, number of recurrent layers, convolutional width, dropout, and learning rate. Their joint tuning forms a mixed discrete–continuous and computationally expensive search problem because evaluating a candidate configuration requires training a complete network. Population-based metaheuristics provide one practical mechanism for exploring this space [12–14].

Motivated by these considerations, this work proposes WiseLSTM, a hybrid depthwise-CNN–LSTM architecture for one-hour-ahead offshore wind-power forecasting. The study evaluates WiseLSTM against recurrent, convolutional, conventional hybrid, and persistence baselines using temporally ordered repeated holdouts. It further characterizes the behavior of PSO and WOA when used to configure the WiseLSTM hyperparameter space.

The main contributions are as follows:

- We propose WiseLSTM, a hybrid architecture that combines two depthwise–pointwise temporal convolutional blocks with unidirectional LSTM modeling. At the channel widths used in the proposed model, the separable front-end requires 65.8% fewer convolutional kernel weights than standard convolutions of equal width.
- We provide a temporally ordered evaluation over five blocked repeated holdouts spanning a 40-year offshore wind series and complement the neural comparison with a strong one-hour persistence benchmark. WiseLSTM reduces mean RMSE by approximately 5.0% relative to CNN–LSTM and 8.5% relative to persistence.
- We characterize a mixed discrete–continuous hyperparameter search using matched 30-agent PSO and WOA populations under the same nominal iteration budget. The WOA-selected configuration primarily improves cross-fold stability, reducing RMSE dispersion by 66.5% relative to the untuned WiseLSTM.
- We analyze the search-cost/generalization trade-off. In the reported executions, WOA required 36.2% more wall-clock search time than PSO but selected a structurally smaller configuration and exhibited a much smaller search-to-evaluation RMSE gap (approximately 1.3% versus 16.6%). These observations characterize the reported searches without implying universal optimizer superiority.

2 Related Work

Wind-power forecasting has been studied using statistical, machine-learning, and deep-learning approaches because of its importance for grid operation and renewable energy management [15, 16]. Classical autoregressive methods provide useful references but may struggle to represent strongly nonlinear relationships among meteorological variables. Machine-learning methods broaden this representation capacity, while deep neural networks can directly learn temporal features from multivariate sequences [17, 18].

Among deep-learning methods, recurrent networks remain common because wind generation is naturally sequential. LSTM models mitigate the vanishing-gradient problem and are able to retain information across an input window, whereas GRU networks provide a simpler gated alternative. Hybrid CNN–LSTM architectures augment this sequential modeling with a convolutional stage that extracts local temporal

features before recurrent processing. Such designs have reported competitive results in wind-power forecasting, including offshore settings [19].

Attention-based and transformer-oriented forecasting has further expanded the range of available temporal models, particularly when longer contexts or multivariate interactions must be represented [9, 10]. These developments provide an important contemporary reference point, while also motivating continued study of compact hybrid architectures when training cost, model size, and configuration stability are relevant design considerations.

Architectural design alone does not determine forecasting performance. Hyperparameter selection affects model capacity, regularization, optimization dynamics, and ultimately generalization. Population-based methods such as PSO and WOA have therefore been investigated as alternatives to manual or exhaustive search [12, 13]. Their appeal in the present context lies in their ability to operate directly over mixed search spaces without requiring differentiability of the hyperparameter objective.

The present study addresses the intersection of these two design dimensions. Rather than treating architectural efficiency and hyperparameter selection independently, WiseLSTM combines a depthwise-separable convolutional front-end with recurrent temporal modeling and subsequently examines how population-based search configures that architecture. The objective is not to establish universal superiority of one metaheuristic, but to characterize the predictive behavior of the architecture and the sensitivity of its performance to the hyperparameter configurations discovered under the available search budget.

3 Methods

Figure 1 summarizes the experimental workflow and the temporal evaluation protocol. Particular emphasis is placed on chronological separation between model fitting, validation, and final out-of-sample assessment.

3.1 Problem Formulation

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ denote a multivariate sequence of wind-related predictors, with $\mathbf{x}_t \in \mathbb{R}^d$. The forecasting task is defined as

$$\hat{y}_{t+h} = f(\mathbf{x}_{t-w+1}, \ldots, \mathbf{x}_t), \tag{1}$$

where w is the input-window length and h is the forecasting horizon. In this study, $w = 24$ hourly observations and $h = 1$ hour. Thus, the recurrent stage is used to model temporal dependencies within a 24-hour context rather than to claim seasonal-scale memory extending beyond the supplied input sequence.

3.2 Dataset Description

Experiments use the Baie de Saint-Brieuc series from the Europe's Wind Farms dataset [20, 21]. The source resource combines 40 years of hourly ERA5-derived meteorological conditions, spanning 1980–2019, with wind-farm generation obtained from turbine-specific technical characteristics and power-curve modeling. The resulting power signal

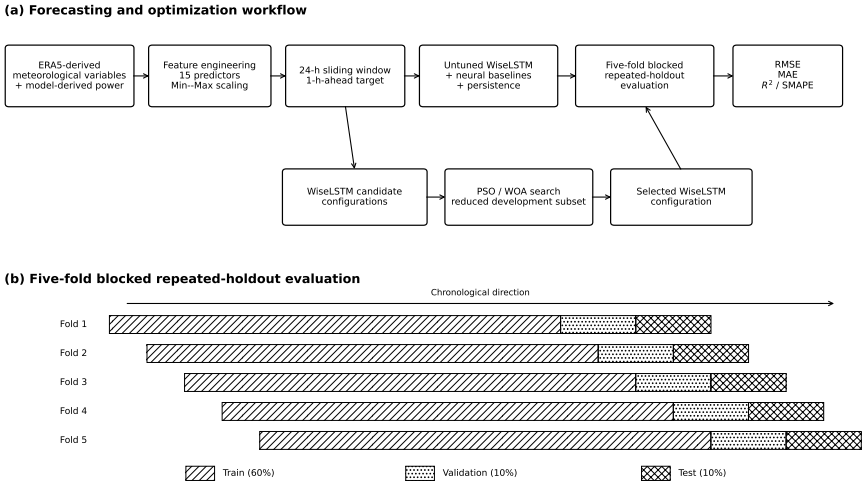


Fig. 1 Overview of the experimental framework and temporal evaluation protocol. **(a)** ERA5-derived meteorological variables and model-derived wind-farm power are transformed into supervised 24-hour input sequences for one-hour-ahead forecasting. WiseLSTM and the baseline architectures are evaluated under a common protocol, while PSO and WOA explore the WiseLSTM hyperparameter space. **(b)** Five-fold blocked repeated-holdout evaluation. Within each fold, training, validation, and test blocks remain contiguous and chronologically ordered, and the complete window is shifted forward between repetitions.

should therefore be interpreted as meteorology-driven and model-derived rather than as direct operational SCADA measurements. For Baie de Saint-Brieuc, the source dataset models a 496 MW farm; accordingly, absolute-error metrics are reported in MW [20].

The analyzed file contains 350,640 consecutive hourly observations. Inspection of the supplied series identified no missing values and no duplicated timestamps. This long and temporally consistent record exposes the models to a broad range of historical atmospheric conditions while maintaining a common generation model throughout the observation period. Operational effects not explicitly represented in the source dataset, such as maintenance interventions, grid curtailment, or farm-specific wake-control decisions, remain outside the scope of the forecasting task.

3.3 Feature Engineering and Preprocessing

The retained meteorological descriptors are the horizontal wind components u_{100} and v_{100}, forecast surface roughness, wind speed, hub-height-scaled wind speed, and wind direction. Calendar information is extracted from the timestamp as hour of day, weekday, and month. These three descriptors are retained and complemented with sine–cosine encodings,

$$x_{\sin} = \sin\left(\frac{2\pi x}{T}\right), \qquad x_{\cos} = \cos\left(\frac{2\pi x}{T}\right), \tag{2}$$

Table 1 Forecasting, preprocessing, and training configuration shared by the learning-based models.

Parameter	Value
Sequence length (w)	24 hours
Forecast horizon (h)	1 hour
Number of predictors (d)	15
Sampling frequency	1 hour
Input structure	Sliding window
Normalization	Min–Max scaling $[0, 1]$
Temporal encoding	Raw calendar + sine–cosine
Evaluation strategy	Five-fold blocked repeated holdout
Optimizer	Adam
Initial learning rate	1×10^{-3}
Learning-rate scheduler	None
Loss function	Huber loss
Batch size	256
Maximum epochs	100
Early stopping	30 epochs without validation improvement

where T denotes the corresponding period. This produces a 15-dimensional input representation: six meteorological variables, three calendar descriptors, and six cyclic components. Variables directly derived from power output or turbine shutdown state are excluded from the predictor set to prevent target leakage. In the supplied Baie de Saint-Brieuc file, the turbine-specific power signal directly reproduces the target power series and is therefore not used as an input feature. Also, all predictors are normalized using Min–Max scaling. Scaling parameters are estimated exclusively from the training block of each temporal fold and are then applied without refitting to the associated validation and test blocks.

3.4 Forecasting Architectures

The recurrent comparison includes one-layer LSTM, BiLSTM, and GRU baselines with hidden size 64. The convolutional baseline uses two depthwise-separable blocks with output widths 64 and 128 followed by global temporal pooling. The conventional CNN–LSTM uses standard Conv1D layers with 32 and 64 output channels before a one-layer LSTM with hidden size 64. All learning-based models use the same 24-hour input window, feature representation, temporal folds, training objective, and evaluation metrics. The layer-wise specification of WiseLSTM is provided in Table 2.

3.5 Proposed WiseLSTM Architecture

WiseLSTM combines two depthwise-separable temporal convolutional blocks with a unidirectional LSTM. The first stage applies a depthwise convolution independently

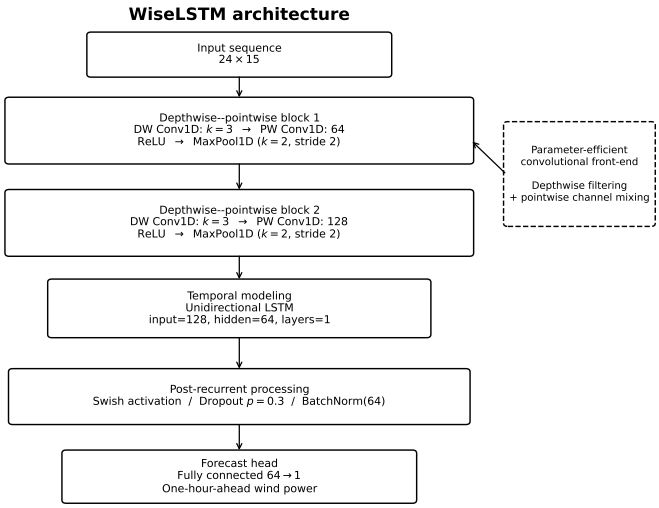


Fig. 2 WiseLSTM architecture. Two depthwise–pointwise temporal convolutional blocks extract local patterns before a unidirectional LSTM models dependencies across the transformed sequence. The recurrent representation is subsequently refined using Swish activation, dropout, and batch normalization before the final one-hour-ahead regression head.

to each input channel and subsequently mixes channels using a 1×1 pointwise convolution. The same principle is repeated in the second block before recurrent processing. Figure 2 illustrates the complete pipeline.

Given an input with C_{in} channels, a standard one-dimensional convolution with C_{out} output channels and kernel size k contains

$$N_{\text{std}} = kC_{\text{in}}C_{\text{out}} \tag{3}$$

kernel weights. The corresponding depthwise–pointwise factorization contains

$$N_{\text{sep}} = kC_{\text{in}} + C_{\text{in}}C_{\text{out}}. \tag{4}$$

For the two WiseLSTM convolutional stages ($15 \to 64$ and $64 \to 128$, $k = 3$), the standard-convolution counterpart would require 27,456 convolutional kernel weights, whereas the depthwise-separable formulation uses 9,389, a reduction of approximately 65.8% at the same channel widths. This comparison concerns the convolutional feature extractor and is not intended to imply that WiseLSTM has fewer total parameters than every benchmark model.

3.6 Five-Fold Temporal Evaluation

Because wind-power time series are non-stationary, the evaluation protocol must preserve chronological order. Following the principles discussed by Cerqueira et al. [22], a five-fold blocked repeated-holdout scheme is used. Each fold contains three contiguous blocks arranged as

Table 2 Layer-wise configuration of the proposed WiseLSTM architecture.

Layer	Configuration	Output
Input	Sequence length 24, features 15	24 time steps × 15 features
Depthwise Conv1	Kernel 3, padding 1, groups 15	24 × 15
Pointwise Conv1	Kernel 1, output 64	24 × 64
ReLU + MaxPool1D	Kernel 2, stride 2	12 × 64
Depthwise Conv2	Kernel 3, padding 1, groups 64	12 × 64
Pointwise Conv2	Kernel 1, output 128	12 × 128
ReLU + MaxPool1D	Kernel 2, stride 2	6 × 128
LSTM	Input 128, hidden 64, one layer	64-dimensional representation
Post-recurrent processing	Swish; dropout $p = 0.3$; BatchNorm(64)	64-dimensional representation
Fully connected	$64 \to 1$	One-hour forecast

$$\text{Training} \longrightarrow \text{Validation} \longrightarrow \text{Test}. \tag{5}$$

The documented fold configuration uses 210,384 observations for training (approximately 60% of the complete series), 35,064 for validation (10%), and 35,064 for testing (10%). The starting index is shifted forward by 17,520 observations between successive folds. The 24-hour input requirement reduces the number of supervised sequences relative to the number of raw observations but does not alter the chronological order of the partitions. Validation is used for early stopping and model-selection decisions associated with the corresponding training process, whereas the subsequent test block is reserved for out-of-sample forecasting performance.

The repeated windows intentionally overlap, so fold-level mean and standard deviation are used as descriptive summaries of performance across temporal segments rather than as independent-replicate estimates for inferential testing.

3.7 Training Configuration

All neural architectures are trained using Adam with an initial learning rate of 10^{-3} and Huber loss. The maximum training horizon is 100 epochs, with a batch size of 256 and early stopping after 30 consecutive epochs without validation improvement. The same training configuration is used for the architectural comparison.

The main experiments were executed in the Kaggle computing environment configured with approximately 30 GB of system RAM and two NVIDIA T4 accelerators with 16 GB of VRAM each. Supplementary development runs were also conducted in Google Colab Pro with an NVIDIA T4 GPU. These hardware descriptions document the experimental environment; wall-clock comparisons are interpreted only for the reported executions.

3.8 Metaheuristic-Based Hyperparameter Search

WiseLSTM is further configured using PSO and WOA. The search targets five hyperparameters spanning architecture, regularization, and optimization: the number of LSTM layers, LSTM hidden size, convolutional output width, dropout probability, and learning rate. The corresponding ranges are 1–5 layers, 32–512 hidden units,

32–512 convolutional output channels, dropout in $[0.1, 0.5]$, and learning rate in $[10^{-5}, 10^{-2}]$. Integer-valued architectural variables are rounded to valid configurations before evaluation.

Both algorithms use a population of 30 candidate solutions and the same nominal maximum of 20 iterations, with early stopping after five consecutive iterations without improvement. For PSO, the inertia weight is $w = 0.7$ and the cognitive and social coefficients are $c_1 = c_2 = 1.5$. For WOA, the control coefficient a decreases linearly from 2 to 0 over the search. Candidate configurations are trained on a reduced chronological development subset to limit computational cost, and fitness is defined by validation RMSE. The held-out test blocks used for the final five-fold assessment are not used to rank candidate configurations. After the search, the selected configurations are evaluated using the complete five-fold forecasting protocol. Consequently, the search objective measures configuration quality on the reduced development setting, whereas the final fold results characterize transfer to broader temporal segments.

3.9 Evaluation Metrics

Forecasting performance is assessed using RMSE, MAE, R^2, and SMAPE. RMSE and MAE are treated as the primary absolute-error measures,

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{6}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{7}$$

while the coefficient of determination is

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{8}$$

SMAPE is retained as a complementary relative-error measure,

$$\mathrm{SMAPE} = \frac{100}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{\max\left((|y_i| + |\hat{y}_i|)/2, \epsilon\right)}. \tag{9}$$

where $\epsilon > 0$ denotes the numerical stabilizer used to avoid division by zero. Because the target contains extended near-zero and zero-generation periods, percentage-based errors are interpreted together with RMSE and MAE rather than in isolation.

3.10 Persistence Benchmark

For the one-hour-ahead task, persistence is included as a non-parametric temporal benchmark,

$$\hat{y}_{t+1} = y_t. \tag{10}$$

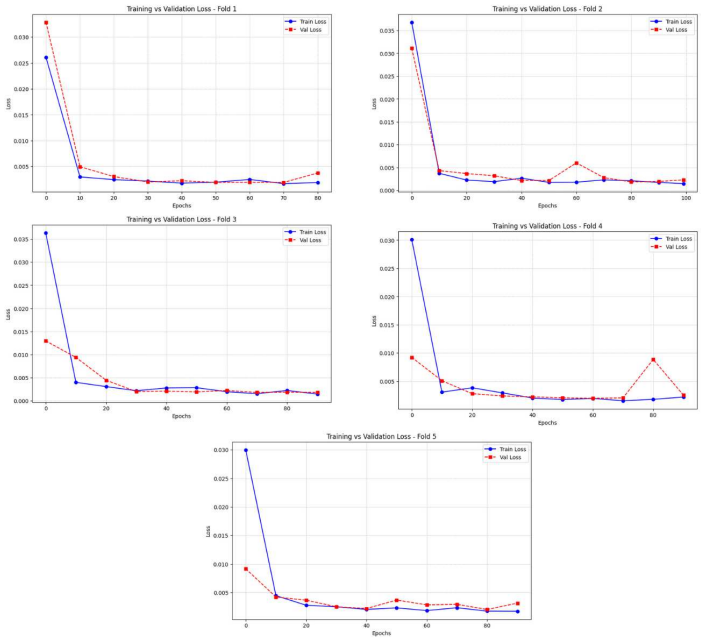


Fig. 3 WiseLSTM training and validation loss across the five blocked repeated-holdout folds.

This reference is intentionally simple but demanding at short horizons because adjacent wind-power observations are strongly temporally related. Persistence is evaluated over the same five held-out temporal test intervals used for the learning-based comparison.

3.11 AI-Assisted Manuscript Preparation

Generative AI tools were used during manuscript preparation for language editing and structural revision. All scientific claims, numerical calculations, methodological descriptions, interpretations, and the final manuscript were reviewed and approved by the authors. Also, no generative AI system is listed as an author.

4 Results

4.1 Training Behavior

Figure 3 shows the training and validation loss trajectories for WiseLSTM across the five temporal folds. The curves exhibit rapid error reduction during the early training stages followed by a slower refinement phase. Training and validation trajectories remain broadly aligned in the reported runs, indicating that the early-stopping configuration provides a practical control on overfitting across the different temporal windows.

Table 3 Predictive performance across the five temporal folds (mean ± standard deviation). Persistence is evaluated over the same five held-out temporal intervals used for the learning-based comparison.

Model	RMSE mean (MW)	RMSE SD (MW)	MAE mean (MW)	MAE SD (MW)	R^2 mean	R^2 SD	SMAPE mean	SMAPE SD
Persistence	37.0689	0.4036	22.2389	0.1721	0.9594	0.0006	26.7899	1.0016
BiLSTM	49.9416	14.9086	39.1912	14.3462	0.9209	0.0502	56.2774	15.1853
LSTM	48.3104	9.1155	38.0752	9.3365	0.9290	0.0250	52.4581	12.5446
DWC	42.7024	8.2401	30.9544	7.5786	0.9445	0.0204	47.8111	11.9518
GRU	39.3669	7.4646	28.6424	7.7513	0.9529	0.0179	43.1315	9.1443
CNN–LSTM	35.7076	3.5132	22.8745	3.0047	0.9621	0.0074	36.7563	2.8455
WiseLSTM	**33.9143**	3.7023	**20.4030**	3.0034	**0.9657**	0.0076	34.3249	2.7679

4.2 Forecasting Performance

Table 3 summarizes the predictive performance of the evaluated models. Among the learned architectures, WiseLSTM achieves the lowest mean RMSE and MAE and the highest mean R^2, with RMSE $= 33.91 \pm 3.70$ MW, MAE $= 20.40 \pm 3.00$ MW, and $R^2 = 0.9657 \pm 0.0076$.

The conventional CNN–LSTM is the closest neural benchmark (RMSE $= 35.71 \pm 3.51$ MW). Relative to CNN–LSTM, WiseLSTM reduces mean RMSE by approximately 5.0% and mean MAE by 10.8%. The result is consistent with the intended architectural role of the proposed front-end: local temporal transformations are performed before recurrent processing, while the depthwise–pointwise factorization allows a wider feature representation with a substantially lower convolutional kernel cost than a standard convolution at the same widths.

The persistence model provides an additional perspective. It achieves RMSE $= 37.07 \pm 0.40$ MW, outperforming several neural baselines despite having no trainable parameters. WiseLSTM nevertheless reduces mean RMSE by approximately 8.5% and MAE by 8.3% relative to persistence and increases R^2 from 0.9594 to 0.9657. This comparison is particularly relevant because a one-hour persistence forecast is a strong reference for a highly autocorrelated power series.

Persistence obtains a lower SMAPE than the neural models. Inspection of the target series shows that 24,262 observations (6.92%) correspond to exactly zero power. Percentage-based metrics become highly sensitive in this regime: persistence frequently reproduces consecutive zero-output periods exactly, whereas small non-zero deviations can receive a comparatively large relative penalty. Accordingly, RMSE and MAE are treated as the primary error metrics and SMAPE as a complementary indicator.

The fold-level standard deviations in Table 3 should be interpreted descriptively. WiseLSTM has lower mean errors than CNN–LSTM, but its RMSE and R^2 dispersion are slightly higher in the untuned configuration. The substantial stability improvement emerges after WOA-based configuration, as shown below.

4.3 Metaheuristic Search Behavior

Figure 4 and Table 4 summarize the two reported hyperparameter-search executions. PSO reduces its objective rapidly during the early search and subsequently reaches a plateau, terminating after 13 iterations. The selected solution is relatively deep and wide, with four LSTM layers, hidden size 113, 420 convolutional output channels, dropout 0.1, and a learning rate of 10^{-5}.

WOA continues to identify improved candidates through later iterations and terminates after 15 iterations. Its selected configuration contains two LSTM layers, hidden size 71, 142 convolutional output channels, dropout 0.1474, and a learning rate of 4×10^{-4}. Relative to the PSO-selected solution, this corresponds to 50% fewer recurrent layers, a 37.2% smaller hidden dimension, and 66.2% fewer convolutional output channels. These percentages describe the selected hyperparameter dimensions and should not be interpreted as an exact whole-model parameter-count reduction. The two searches also expose a computational trade-off. PSO requires 550.94 min of wall-clock search time, whereas WOA requires 750.12 min, an additional 199.18 min or approximately 36.2%. The greater search time is therefore not presented as an efficiency advantage. Rather, the reported WOA execution trades additional search cost for a configuration that transfers more consistently to the full temporal evaluation. This transfer can be described by comparing each best search fitness with the mean RMSE obtained after the selected configuration is evaluated under the full five-fold protocol. For PSO, RMSE increases from 43.1120 during search to 50.2768 during final evaluation, a relative gap of approximately 16.6%. For WOA, the corresponding values are 33.2270 and 33.6660, a gap of approximately 1.3%. Because search fitness and final RMSE are measured on different temporal partitions, these gaps are descriptive rather than inferential. Nevertheless, the much smaller WOA gap is consistent with better transfer of the selected configuration from the reduced development setting to the broader temporal evaluation.

The principal benefit of the WOA-selected configuration is likewise observed in stability rather than in a large shift in mean accuracy. Relative to untuned WiseLSTM, mean RMSE changes from 33.9143 to 33.6660 (approximately 0.7%), while RMSE standard deviation decreases by 66.5%, MAE standard deviation by 70.5%, and R^2 standard deviation by 68.4%. By contrast, the PSO-selected configuration reaches 50.2768 ± 11.8464 RMSE under the same final evaluation protocol.

The results therefore support two conclusions that remain within the evidence. First, hyperparameter selection can materially alter both accuracy and temporal stability even when the underlying WiseLSTM architecture is unchanged. Second, the reported WOA execution identifies a more transferable and cross-fold-stable configuration than the reported PSO execution, despite requiring more search time. The trajectories characterize these specific stochastic executions; establishing optimizer-level superiority would require repeated independent searches under matched budgets.

5 Discussion

The results support the value of combining local temporal feature extraction with recurrent sequence modeling. WiseLSTM improves mean RMSE and MAE relative to the conventional CNN–LSTM benchmark and also outperforms one-hour persistence, which proved stronger than several neural baselines. This ordering is informative because it places the proposed architecture above both a competitive short-horizon time-series heuristic and a conventional hybrid deep-learning model.

The depthwise-separable front-end should be interpreted primarily as an efficient factorization of local feature extraction rather than as evidence that the complete

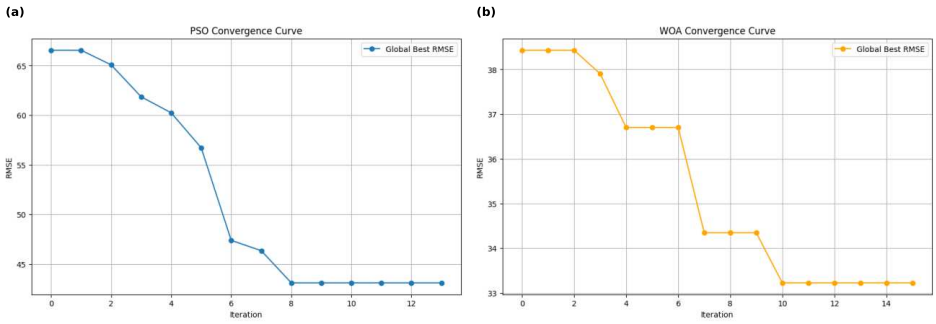


Fig. 4 Best objective value observed during the reported metaheuristic executions. **(a)** PSO improves rapidly during the early search and subsequently reaches a plateau. **(b)** WOA continues to identify improved configurations through later iterations before stabilization. The curves characterize these executions and are not interpreted as repeated-run evidence of algorithm-level superiority.

Table 4 Metaheuristic search outcomes, selected hyperparameters, and final predictive performance. Search fitness is the best RMSE observed during hyperparameter selection; final metrics are mean ± standard deviation across the five blocked repeated-holdout test folds.

(a) Search outcome and selected configuration

Metric / Parameter	**PSO**	**WOA**
Population size	30	30
Maximum iterations	20	20
Iterations reached	13	15
Elapsed time (min)	550.94	750.12
Best search fitness, RMSE (MW)	43.1120	33.2270
Search-to-evaluation RMSE gap	16.6%	1.3%
LSTM layers	4	2
LSTM hidden size	113	71
CNN output channels	420	142
Dropout probability	0.1000	0.1474
Learning rate	1×10^{-5}	4×10^{-4}

(b) Final forecasting performance

Model	RMSE mean (MW)	RMSE SD (MW)	MAE mean (MW)	MAE SD (MW)	R^2 mean	R^2 SD	SMAPE mean	SMAPE SD
WiseLSTM	33.9143	3.7023	20.4030	3.0034	0.9657	0.0076	34.3249	2.7679
WiseLSTM–PSO	50.2768	11.8464	37.1325	11.0197	0.9219	0.0358	45.3250	6.0574
WiseLSTM–WOA	**33.6660**	**1.2409**	**20.2262**	**0.8849**	**0.9665**	**0.0024**	**34.0384**	**1.4170**

WiseLSTM network is universally smaller than all comparison models. At matched channel widths, the two depthwise–pointwise blocks require 65.8% fewer convolutional kernel weights than standard convolutions. This efficiency allows local temporal representations to be expanded before recurrent modeling without incurring the kernel cost of an equally wide conventional convolutional front-end.

The metaheuristic results reveal a second aspect of the problem. Hyperparameter selection changes not only average accuracy but also the consistency of the model across temporal windows. The reported searches also reveal a practical cost–transfer

trade-off. WOA requires 36.2% more wall-clock search time than PSO, but it selects a configuration that is structurally smaller along the optimized architecture dimensions and whose validation-search fitness remains much closer to its final five-fold RMSE. This pattern is consistent with a search that expends additional computational effort to identify a configuration that transfers more faithfully beyond the reduced optimization subset. Because only one stochastic execution of each optimizer is available, this interpretation is restricted to the observed runs and does not constitute an optimizer-level ranking. The WOA-selected configuration produces only a small additional reduction in mean RMSE relative to untuned WiseLSTM, yet its fold-level dispersion is markedly lower. The more compact recurrent configuration identified in that execution suggests that additional depth and width are not automatically beneficial for this forecasting task. Conversely, the PSO-selected configuration shows substantially poorer performance under the final evaluation protocol than under its internal search objective. This contrast highlights the importance of evaluating selected hyperparameters outside the optimization trajectory.

Regarding the hydrogen context it should be interpreted at the forecasting layer of the renewable-to-hydrogen chain. WiseLSTM does not directly optimize electrolyzer operation, hydrogen yield, or production cost. Instead, more accurate and stable estimates of near-term wind-power availability can reduce uncertainty in the information supplied to downstream scheduling and energy-management models. Quantifying the operational value of those forecasts within a coupled forecasting–optimization framework remains a natural extension of the present study.

6 Conclusion

This study introduced WiseLSTM, a hybrid architecture that combines depthwise-separable temporal convolutions with LSTM modeling for one-hour-ahead offshore wind-power forecasting. Evaluated over five chronologically ordered repeated holdouts, WiseLSTM achieved the best mean RMSE, MAE, and R^2 among the evaluated learning architectures. It also outperformed one-hour persistence, reducing mean RMSE by approximately 8.5%, and improved upon the conventional CNN–LSTM benchmark by approximately 5.0% in RMSE and 10.8% in MAE.

The architecture provides a parameter-efficient mechanism for local feature extraction: at the channel widths used by WiseLSTM, depthwise–pointwise factorization reduces convolutional kernel weights by approximately 65.8% relative to standard convolutions of equal width. The results therefore support the use of efficient local filtering before recurrent processing in a highly variable short-horizon forecasting setting.

Hyperparameter optimization further showed that configuration choice can strongly affect generalization. The WOA-selected model produced only a modest improvement in mean error but substantially reduced cross-fold dispersion. It also exhibited a search-to-evaluation RMSE gap of approximately 1.3%, compared with 16.6% for the PSO-selected configuration, despite requiring 36.2% more search time. These observations emphasize that the practical value of hyperparameter search lies not only in obtaining a lower internal objective but also in identifying configurations that remain stable under temporally distinct evaluation windows.

In this work, WiseLSTM provided a compact methodological framework for studying the interaction between local temporal representation, recurrent forecasting, and metaheuristic configuration. Its forecasts may support downstream renewable-to-hydrogen scheduling, while direct quantification of hydrogen-system benefits requires explicit integration with electrolyzer and operational optimization models.

7 Future Work

Several directions follow naturally from the present results. First, the architecture should be evaluated across additional wind farms and geographical regions to determine how well its gains transfer beyond the Baie de Saint-Brieuc series. Multi-site and probabilistic formulations would be particularly useful for characterizing uncertainty under heterogeneous atmospheric regimes [23].

Second, future work may compare WiseLSTM with modern transformer-based and foundation-model approaches using the same temporal folds and forecasting horizon. Such an extension would clarify the trade-off between forecast accuracy, architectural complexity, and computational requirements [24].

Third, the hyperparameter study can be strengthened through multiple independent runs of each optimizer under matched computational budgets and through additional search strategies such as Bayesian optimization [25]. Repeated runs would make it possible to separate the behavior of a particular stochastic execution from systematic differences between optimizers.

Finally, the forecasting model can be incorporated into a downstream optimization framework for electrolyzer scheduling and hydrogen production. Such integration would enable the value of improved wind forecasts to be quantified directly in terms of operational decisions, hydrogen yield, and economic performance [26].

Acknowledgements

The authors extend their gratitude to the PUCV Library for providing access to essential documents and resources.

Funding declaration

Emanuel Vega was supported by Proyecto de Investigación Asociativa Interdisciplinaria PUCV 2025 (DI-PUCV) under Grant 2025/pucv/039.774/2025, and by the Agencia Nacional de Investigación y Desarrollo (ANID) under FONDECYT Initiation Project 11261348.

Author contributions

Joaquín Toro: Methodology, Software, Investigation, Data curation, Writing – original draft.
Tania Roa: Conceptualization, Formal analysis, Investigation.
Andrés Aránguiz: Writing – review & editing, Conceptualization.

José Garcia: Validation, Investigation, Conceptualization.
El-Ghazali Talbi: Validation, Writing – review & editing.
Marcelo Becerra-Rozas: Methodology, Formal analysis, Validation, Writing – review & editing.
Emanuel Vega: Formal analysis, Validation, Supervision, Project administration, Methodology, Conceptualization.

Data availability. The dataset analyzed in this study is publicly available through Figshare (DOI: 10.6084/m9.figshare.19139648) [21]. The Baie de Saint-Brieuc series was selected for the experiments reported here.

Competing interests

The authors declare no competing interests.

References

[1] Gayen, D., Chatterjee, R. & Roy, S. A review on environmental impacts of renewable energy for sustainable development. *International Journal of Environmental Science and Technology* **21**, 5285–5310 (2024).

[2] Horri, B. A. & Ozcan, H. Green hydrogen production by water electrolysis: Current status and challenges. *Current Opinion in Green and Sustainable Chemistry* **47**, 100932 (2024).

[3] Staffell, I. *et al.* The role of hydrogen and fuel cells in the global energy system. *Energy & Environmental Science* **12**, 463–491 (2019).

[4] Foley, A. M., Leahy, P. G., Marvuglia, A. & McKeogh, E. J. Current methods and advances in forecasting of wind power generation. *Renewable Energy* **37**, 1–8 (2012).

[5] Giebel, G. & Kariniotakis, G. Wind power forecasting—a review of the state of the art. *Renewable energy forecasting* 59–109 (2017).

[6] Lai, J. P., Chang, Y. M., Chen, C. H. & Pai, P. F. A survey of machine learning models in renewable energy predictions. *Applied Sciences* **10**, 5975 (2020).

[7] Rana, A. & Pathak, N. K. Deep learning for renewable energy forecasting: A review. *Asian Journal of Multidimensional Research* **10**, 366–372 (2021).

[8] Hong, Y.-Y. & Rioflorido, C. L. P. P. A hybrid deep learning-based neural network for 24-h ahead wind power forecasting. *Applied Energy* **250**, 530–539 (2019).

[9] Saravanan, H. K., Dwivedi, S. & Arjunan, P. Poster abstract: Analyzing the performance of time series foundation models for short-term load forecasting.

Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys 2024) 237–238 (2024).

[10] Hornek, T., Sartipi, A., Tchappi, I. & Fridgen, G. Benchmarking pre-trained time series models for electricity price forecasting (2025). URL https://arxiv.org/abs/2506.08113. arXiv:2506.08113.

[11] Chollet, F. Xception: Deep learning with depthwise separable convolutions (2017). URL https://arxiv.org/abs/1610.02357. arXiv:1610.02357.

[12] Jain, M., Saihjpal, V., Singh, N. & Singh, S. B. An overview of variants and advancements of pso algorithm. *Applied Sciences* **12**, 8392 (2022).

[13] Mohammed, H. M., Umar, S. U. & Rashid, T. A. A systematic and meta-analysis survey of whale optimization algorithm. *Computational intelligence and neuroscience* **2019**, 8718571 (2019).

[14] Kaveh, M. & Mesgari, M. S. Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review. *Neural Processing Letters* **55**, 4519–4622 (2023).

[15] Alazemi, T., Darwish, M. & Radi, M. Renewable energy sources integration via machine learning modelling: A systematic literature review. *Heliyon* **10** (2024).

[16] Demolli, H., Dokuz, A. S., Ecemis, A. & Gokcek, M. Wind power forecasting based on daily wind speed data using machine learning algorithms. *Energy Conversion and Management* **198**, 111823 (2019).

[17] Dubey, A. K., Kumar, A., Segovia Ramirez, I. & Garcia Marquez, F. P. Xu, J., Altiparmak, F., Hassan, M. H. A., García Márquez, F. P. & Hajiyev, A. (eds) *A review of intelligent systems for the prediction of wind energy using machine learning.* (eds Xu, J., Altiparmak, F., Hassan, M. H. A., García Márquez, F. P. & Hajiyev, A.) *Proceedings of the Sixteenth International Conference on Management Science and Engineering Management – Volume 1*, Vol. 144 of *Lecture Notes on Data Engineering and Communications Technologies*, 476–491 (Springer, Cham, 2022). URL https://doi.org/10.1007/978-3-031-10388-9_35.

[18] Forootan, M. M., Larki, I., Zahedi, R. & Ahmadi, A. Machine learning and deep learning in energy systems: A review. *Sustainability* **14**, 4832 (2022).

[19] Sun, Y. *et al.* CNN–LSTM–AM: A power prediction model for offshore wind turbines. *Ocean Engineering* **301**, 117598 (2024).

[20] Grothe, O., Kächele, F. & Watermeyer, M. Analyzing Europe's biggest offshore wind farms: A data set with 40 years of hourly wind speeds and electricity production. *Energies* **15**, 1700 (2022).

[21] Kächele, F., Grothe, O. & Watermeyer, M. Dataset for the paper: "analyzing europe's biggest offshore wind farms: a data set with 40 years of hourly wind speeds and electricity production". Dataset (2022). URL https://doi.org/10.6084/m9.figshare.19139648.v1.

[22] Cerqueira, V., Torgo, L. & Mozetič, I. Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning* **109**, 1997–2028 (2020).

[23] Hong, T. *et al.* Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond (2016).

[24] Zhou, H. *et al.* Informer: Beyond efficient transformer for long sequence time-series forecasting (2021). URL https://arxiv.org/abs/2012.07436. arXiv:2012.07436.

[25] Snoek, J., Larochelle, H. & Adams, R. P. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems* **25** (2012).

[26] Glenk, G. & Reichelstein, S. Economics of converting renewable power to hydrogen. *Nature Energy* **4**, 216–222 (2019).