IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 November 2025, accepted 28 November 2025, date of publication 4 December 2025, date of current version 12 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3640434

RESEARCH ARTICLE

Enhanced Random Vector Functional Link Networks With Bayesian-Based Hyperparameter Optimization for Wind Speed Forecasting

LAIO ORIEL SEMAN[1], ANNE CAROLINA RODRIGUES KLAAR[2], MATHEUS HENRIQUE DAL MOLIN RIBEIRO[3], AND STEFANO FRIZZO STEFENON[4]

[1]Department of Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis 88040-900, Brazil
[2]Graduate Program in Education, University of Planalto Catarinense, Lages 88509-900, Brazil
[3]Industrial and Systems Engineering Graduate Program, Federal University of Technology—Paraná, Pato Branco 80230-901, Brazil
[4]Lisbon School of Engineering (ISEL), Polytechnic University of Lisbon (IPL), 1959-007 Lisbon, Portugal

Corresponding author: Laio Oriel Seman (laio.seman@ufsc.br)

This work was supported in part by the Coordination for the Improvement of Higher Education Personnel (CAPES Transformative Agreement); in part by the National Council for Scientific and Technological Development (CNPq), Brazil; and in part by the Araucária Foundation under Grant PRD2023361000550.

ABSTRACT Accurate short-term wind speed forecasting is essential for reliable and efficient wind energy integration. This paper introduces an enhanced Random Vector Functional Link (RVFL) network optimized through a Bayesian-based Neural Architecture Search (NAS) framework. The proposed RVFL-OptBayes model incorporates multi-scale feature generation, including kernel approximations, Nyström sampling, Fastfood transforms, wavelet scattering, and Neural Tangent Kernel embeddings with Principal Component Analysis (PCA)-aligned orthogonal initializations and spectral normalization to improve stability and feature diversity. Experiments were conducted on real-world Brazilian wind farm data to evaluate forecasting performance. Results show that RVFL-OptBayes outperforms conventional RVFL networks, deep learning models, and ensemble methods, achieving an R^2 above 0.99. The proposed framework demonstrates that lightweight randomized architectures, when combined with principled hyperparameter search, can rival or surpass complex deep learning models for time-series forecasting. The findings suggest strong potential for practical deployment in renewable energy systems, offering accurate and computationally efficient wind speed predictions to support operational planning, grid stability, and smart energy management.

INDEX TERMS Differentiable neural architecture, neural network architectures, predictive maintenance, vibration, forecasting, anomaly detection.

I. INTRODUCTION

Wind energy plays a key role in Brazil's ongoing transition toward sustainable power generation. With over 29GW of installed capacity, predominantly in the Northeast, and contributing more than 13% of national electricity output, the sector's growth has introduced challenges related to variability and forecasting accuracy. Traditional forecasting methods, including Numerical Weather Prediction (NWP), autoregressive models, and Kalman filters, are limited by assumptions of linearity and stationarity that do not hold for highly nonlinear and multiscale wind dynamics at turbine-level resolution. Meanwhile, purely data-driven neural models can capture nonlinearity but often disregard the valuable structural information embedded in physical processes [1].

Random Vector Functional Link (RVFL) networks offer a compelling alternative, combining the simplicity and speed of randomized hidden layers with robust approximation capabilities [2]. They have been successfully applied in energy forecasting tasks, such as wind power prediction with Capuchin Search-optimized RVFL [3] and outlier-robust ensemble deep RVFL for wind speed prediction [4].

Motivated by these strengths, this paper introduces an enhanced RVFL framework for short-term turbine-level wind speed forecasting. The proposed system is evaluated on

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

real Brazilian wind. Results demonstrate improvements over standard baselines. Key innovations include:

- Multi-component feature generation combining kernel-approximation, Nyström sampling, Fastfood transforms, and Neural Tangent Kernel (NTK) embeddings.
- Principal Component Analysis (PCA)-aligned orthogonal initializations and spectral normalization to improve feature diversity and stability.
- Bayesian hyperparameter optimization (via Optuna) to adaptively tune model complexity.
- Ridge/ADMM weighted solvers and ensemble-based structure to manage regularization and interpretability.

The remainder of the paper is structured as follows: Section II reviews relevant literature emphasizing RVFL configurations in time-series forecasting. Section III details the proposed model framework. Section III explains the preprocessing and the dataset characteristics. Section V presents experimental results, and Section VI discusses outcomes and outlines future directions.

II. RELATED WORKS

Wind speed forecasting has progressed from statistical approaches toward hybrid and deep learning models. Early studies employed autoregressive integrated moving average and Kalman filters to model wind speeds as stochastic processes [5]. Although these models are simple and interpretable, they rely on linearity and stationarity assumptions that do not adequately represent the nonlinear and time-varying nature of wind speed [6].

Recent research focuses on data-driven deep learning methods that capture nonlinear temporal dependencies [7]. Hybrid architectures that combine Convolutional Neural Networks (CNNs) and recurrent layers have been widely adopted. For instance, an adaptive spatial pyramid pooling CNN-Long Short-Term Memory (LSTM) with attention model was proposed for wind farm clusters under transitional weather conditions [8], demonstrating improved performance over simpler hybrids. In the context of ultra-short-term forecasting, Yu et al. [9] introduced an optimized CNN-Bidirectional LSTM (BiLSTM)-Attention model, validated on real-world power datasets, yielding notable accuracy gains. These results highlight the value of integrating bidirectional temporal modeling and attention mechanisms.

Spectral-domain information has also been incorporated into forecasting models. Shu et al. [10] introduced a hybrid model that applies fast Fourier transform and rank pooling to extract periodic and transient features, which are then processed by MultiLayer Perceptrons (MLPs) or LSTM networks, followed by linear regression. This approach improves multi-step forecasting by combining frequency-domain and temporal features.

Modeling spatial dependencies among turbines or stations has become another active direction. Wang et al. [11] proposed an adaptive-gated multi-graph attention network that learns behavioral similarities and directional causality between turbines, combined with recurrent layers for temporal modeling. Dong et al. [12] developed the dynamic graph embedding-based graph neural network-LSTM joint framework model for marine wind forecasting, using dynamic graph embeddings to represent spatial correlations among offshore measurement nodes. He [13] combined Graph Attention Networks with LSTM for multi-station forecasting, showing improvements over standalone LSTM and graph convolutional networks.

In addition to data-driven methods, hybrid approaches combining physical knowledge and machine learning are emerging. Large-scale Artificial Intelligence (AI) weather forecasting systems, such as DeepMind's GraphCast, have outperformed traditional NWP in global meteorological forecasts [14]. These developments suggest that physics-informed neural networks and AI-NWP hybrid systems can improve wind speed forecasting at regional and national scales.

Wang et al. [15] proposed a dynamic non-constraint ensemble model for probabilistic wind power and speed forecasting. Four deep Gaussian neural networks generate base forecasts, which are integrated using a weighted sum of Gaussian variables. Ensemble weights are dynamically learned via quantile functions, CNNs, and channel attention, with input length optimized by the maximal information coefficient. Experiments on four real-world datasets show that their model outperforms other approaches, reducing pinball loss by 4.93% and 16.64%, respectively, with hypothesis testing confirming the approach's effectiveness.

Zhang et al. [16] presented a hybrid adaptive decomposition denoising algorithm that mitigates unreasonable decomposition and residual noise. LSTM model parameters are optimized using a seagull algorithm enhanced with a chaotic system and a Cauchy operator. Interval predictions are generated via kernel density estimation with data enhancement. Validation on historical data from Sotavento (Spain) and Eman (China) wind farms shows point prediction errors of 2.87% and 8.01%, respectively, while the interval model achieves narrower intervals and higher average interval scores than benchmarks. The results demonstrate the system's superior stability, reliability, and practical value for wind farm management.

Accurate wind speed forecasting is essential for reliable wind energy production, yet existing models often fail to capture complex spatio-temporal patterns. Wu and Ling [17] introduced the spatio-temporal enhanced pre-trained Large Language Model (LLM), leveraging LLM reasoning for wind speed prediction. Wind series are decomposed into seasonal and trend components, while spatial and temporal prompts encode geographic and short-term temporal information. An autoregressive fine-tuning strategy enables patch-level representation learning, followed by a localized spatial module to capture turbine dependencies. Experiments on four public datasets demonstrate that their approach achieves superior forecasting performance.

Wu and Ling [18] proposed a Mixture Transformer with hierarchical context (Mixformer). Wind series are decomposed into seasonal and trend components, with an MLP predicting trends and an attention model predicting seasonal patterns. A spatio-temporal Gaussian mixture attention layer fuses periodic temporal and long-term spatial information, complemented by Dynamic Time Warping for global spatial features. Experiments on four benchmark datasets show Mixformer achieves the lowest errors, improving Mean Absolute Error (MAE) by 4.14 to 8.43% and Root Mean Square Error (RMSE) by 4.01 to 9.18% over state-of-the-art methods, demonstrating strong practical potential.

Bashir et al. [19] introduced two hybrid models: CNN-ABiLSTM, combining CNN with Attention-based Bidirectional LSTM, and CNN-Transformer-MLP, integrating CNN with Transformers and MLPs. CNN captures short-term patterns, while ABiLSTM and Transformer-MLP model long-term dependencies. Trained on quarter-hour real-time data, both hybrids outperform standalone CNN, BiLSTM, and Transformer models. CNN-Transformer-MLP excels in day- and week-ahead predictions, while CNN-ABiLSTM achieves superior month-ahead wind forecasts, demonstrating effectiveness across short- and long-term horizons.

Wang et al. [20] presented a novel forecasting framework using a two-stage data processing method. Singular spectrum analysis and variational mode decomposition are combined to decompose the trend and residual components, capturing inherent sequence characteristics. A multi-objective optimization strategy determines the weights of predictions from deep learning and improved extreme learning machine models. Experiments show the proposed framework outperforms benchmark models, providing an effective solution for wind speed forecasting and supporting power grid management.

Mohapatro et al. [21] developed a machine learning-based framework for fault prediction in wind turbines using data acquired from a supervisory control and data acquisition system. They focused on reducing the number of input features through hyperparameter tuning to improve computational efficiency without compromising predictive accuracy. To evaluate performance, the authors conducted numerical analyses showing that the optimized support vector machine model achieved F1-scores ranging from 58% to 94% and classification accuracy between 73% and 98%, depending on the fault type.

Tian et al. [22] developed a six-module wind power forecasting system for new wind farms with limited historical data, integrating a transformer network with a parameter-sharing transfer learning strategy and emphasizing model interpretability. A feature selection module combined with an attention mechanism identifies key input features and assigns importance weights. Three simulation experiments using ten multivariate datasets from two Chinese wind farms showed the system outperforms six benchmark models, achieving average improvements of 46.29% in MAE and 31.02% in RMSE compared to the worst-performing MLP. Transfer learning further enhanced performance, reducing MAE by 13.84% and RMSE by 7.77%.

Recent work emphasizes hybrid deep learning approaches that integrate temporal sequence learning [23], spatial dependency modeling [24], and frequency-domain feature extraction [25]. These methods address the nonlinear and multiscale nature of wind speed more effectively, enabling more accurate and robust forecasting frameworks [26].

III. DATASET

The dataset used in this study is regarding the wind speed (m/s) from Catetité, Bahia, Brazil. The timestamp is 10 minutes from March 2020 to May 2020. This database was retrieved from MERRA2 (''Modern Era Retrospective-analysis for Research and Applications'').

A. DATASET DESCRIPTION AND REGIONAL CONTEXT

For this investigation, operational wind turbine supervisory control and data acquisition measurements were obtained from a coastal wind farm located in northeastern Brazil. The dataset consists of approximately 13,248 samples recorded every 10 minutes ranging from 2020-03-01 to 2020-05-31, providing high-resolution information on wind speed, wind direction, and complementary operational parameters. Although these measurements are site-specific rather than covering a broader regional network, they are representative of the wind regime typical of Brazil's northeastern coastline, characterized by persistent Atlantic trade winds and minimal topographic complexity.

The measurement site lies within a wind-rich corridor where the stable trade-wind system produces relatively narrow wind speed distributions favorable for energy production. Such atmospheric conditions maximize turbine availability and ensure a high degree of predictability. Figure 1 shows the empirical probability density of wind speeds, highlighting the concentration of values within the optimal operational range of modern utility-scale wind turbines. A statistical overview is provided in Figure 2, which visualizes the quartile range and outlier structure via a boxplot representation.

Meteorological characteristics extracted from the dataset reveal a robust and stable wind resource. Key statistical indicators include:

- **Mean wind speed:** 7.90 m/s over the entire measurement period.
- **Standard deviation:** 1.72 m/s, indicating moderate variability.
- **Percentile distribution:** $Q_{25} = 6.82$ m/s, $Q_{50} = 7.90$ m/s (median), and $Q_{75} = 9.05$ m/s.
- **Observed range:** from a minimum of 0.74 m/s to a maximum of 14.91 m/s.
- **Coefficient of variation:** 0.22, indicating a stable regime with relatively low relative variability.

These results confirm the site's suitability for wind energy generation, with a high frequency of wind speeds falling within the turbine's optimal operational envelope. Moreover,

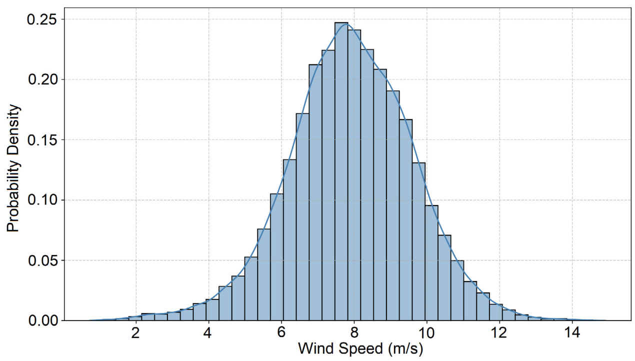


FIGURE 1. Empirical wind speed distribution for the wind dataset, showing histogram and kernel density estimate. Most observations are concentrated between 6.8 m/s and 9.0 m/s, confirming a stable and energy-rich wind regime.

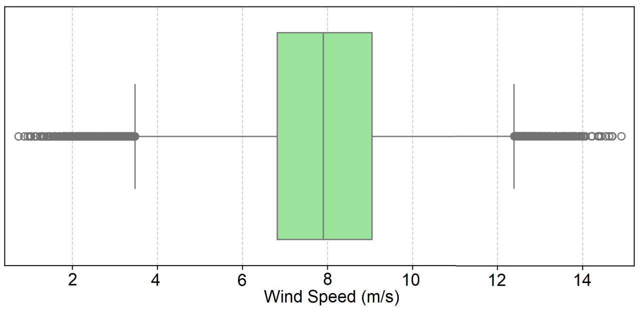


FIGURE 2. Boxplot representation of wind speeds from the wind dataset. The central quartile range is relatively narrow, reinforcing the regime's stability.

the relatively symmetric distribution reduces the incidence of prolonged low-wind events, enhancing forecast reliability and operational planning for energy dispatch. The present analysis focuses on wind speed magnitude as the primary predictor for short-term energy forecasting.

B. DATA IMPUTATION METHODOLOGY

Wind speed observations often contain gaps due to instrument malfunctions, maintenance periods, or adverse weather conditions that compromise measurement quality. To address these missing data challenges, we developed a four-stage imputation procedure that preserves the natural diurnal variability while maintaining temporal continuity across gap boundaries.

1) DIURNAL CLIMATOLOGY FRAMEWORK

The foundation of our approach rests on the construction of hourly climatological statistics from the available observations. Let $W(t)$ represent the observed wind speed time series, where missing values are denoted as NaN. For each hour of day $h \in \{0, 1, \ldots, 23\}$, we compute the climatological mean and standard deviation:

$$\bar{W}_h = \frac{1}{N_h} \sum_{t:H(t)=h, W(t)\neq \text{NaN}} W(t) \tag{1}$$

$$\sigma_h = \sqrt{\frac{1}{N_h - 1} \sum_{t:H(t)=h, W(t)\neq \text{NaN}} (W(t) - \bar{W}_h)^2} \tag{2}$$

where $H(t)$ extracts the hour of day from timestamp t, and N_h represents the number of valid observations for hour h. These statistics capture both the central tendency and natural variability observed at each hour across the measurement period.

2) STOCHASTIC IMPUTATION WITH TEMPORAL VARIABILITY

Rather than filling gaps with deterministic climatological means, our method incorporates realistic variability through stochastic imputation. For each missing observation at time t_{miss} where $H(t_{\text{miss}}) = h$, we generate an initial imputed value:

$$W_{\text{imp}}(t_{\text{miss}}) = \bar{W}_h + \epsilon_h \tag{3}$$

where $\epsilon_h \sim \mathcal{N}(0, \sigma_h^2)$ represents Gaussian noise with zero mean and variance matching the observed hourly variability. This approach prevents artificial variance reduction commonly associated with deterministic imputation methods while preserving the statistical characteristics of the original time series.

3) TEMPORAL CONTINUITY ENHANCEMENT

To mitigate abrupt transitions at gap boundaries, we apply temporal smoothing through a centered rolling mean operation. The smoothed series is computed as:

$$W_{\text{smooth}}(t) = \frac{1}{w} \sum_{i=-\lfloor w/2 \rfloor}^{\lfloor w/2 \rfloor} W_{\text{temp}}(t + i) \tag{4}$$

where $W_{\text{temp}}(t)$ represents the series after stochastic imputation, and $w = 6$ time steps defines the smoothing window.

This operation utilizes all available data points within the window, requiring a minimum of one observation to compute the local average. The smoothing effectively blends imputed values with neighboring observations, creating gradual transitions that better reflect natural wind speed evolution.

4) SELECTIVE INTEGRATION

The final step preserves measurement integrity by selectively applying imputed values only where data were originally missing. The complete imputed series is constructed as:

$$W_{\text{final}}(t) = \begin{cases} W(t) & \text{if } W(t) \neq \text{NaN} \\ W_{\text{smooth}}(t) & \text{if } W(t) = \text{NaN}. \end{cases} \tag{5}$$

This selective approach maintains the original observations unchanged while filling gaps with physically plausible values that respect both diurnal patterns and local variability characteristics.

5) METHOD PROPERTIES

The proposed imputation framework exhibits desirable characteristics for meteorological time series analysis. The climatological foundation ensures that filled values reflect expected diurnal behavior observed in historical data, while the stochastic component maintains realistic variance structure within each hour.

The temporal smoothing operation reduces discontinuities at gap edges without compromising the integrity of the original measurements. The method scales naturally to accommodate seasonal variations by computing time-dependent climatologies $\bar{W}_{h,s}$ and $\sigma_{h,s}$, where s denotes seasonal indices, enabling more nuanced imputation for datasets spanning multiple years or distinct meteorological regimes.

C. DATA FILTERING METHODOLOGY

Following imputation, the reconstructed wind speed series contains both measurement noise and imputation artifacts that require systematic removal to extract underlying meteorological signals. We implement a two-stage hybrid filtering approach that combines causal multiscale decomposition [27] with forward-only adaptive state-space filtering to achieve optimal noise reduction while strictly preserving temporal causality for forecasting applications.

1) CAUSAL MULTISCALE FIR FILTERING

The initial filtering stage employs cascaded Finite Impulse Response (FIR) low-pass filters applied causally through forward-only convolution [28]. For a given signal $W(t)$ of length N, we apply a sequence of M causal FIR filters with progressively decreasing cutoff frequencies to achieve multiscale smoothing:

$$W_{\mathrm{FIR}}(t) = \mathcal{L}_M \circ \mathcal{L}_{M-1} \circ \cdots \circ \mathcal{L}_1[W(t)] \tag{6}$$

where $\mathcal{L}_i$ represents the i-th causal filtering operation with cutoff frequency f_i in Nyquist-normalized units. We utilize linear-phase FIR filters designed via the window method with cutoff frequencies $\{f_1, f_2, f_3\} = \{0.25, 0.125, 0.0625\}$ and corresponding filter orders $\{n_1, n_2, n_3\} = \{31, 41, 51\}$ taps to capture meteorological variability from turbulent fluctuations to diurnal cycles.

The causal filter coefficients $\mathbf{b}_i$ for each stage are computed using the Hamming window method:

$$\mathbf{b}_i = \mathrm{firwin}(n_i, f_i), \quad i = 1, \ldots, M. \tag{7}$$

Each filtering step is applied using only forward linear convolution, ensuring that the filtered output at time t depends exclusively on the observations at that time, including:

$$y_i(t) = \sum_{k=0}^{n_i-1} b_i[k] \cdot y_{i-1}(t-k), \quad y_0(t) = W(t). \tag{8}$$

This cascade produces a group delay of approximately $(n_1 + n_2 + n_3 - 3)/2$ samples, which is acceptable in forecasting contexts and does not introduce non-causal information leakage.

2) FORWARD-ONLY ADAPTIVE KALMAN FILTERING

The second filtering stage applies forward-only Kalman filtering with past-dependent adaptive observation noise to address residual artifacts and imputation discontinuities [29]. We model the wind speed evolution as a random walk process with time-varying observation noise that adapts to historical signal characteristics:

$$x_{t+1} = x_t + w_t, \quad w_t \sim \mathcal{N}(0, Q) \tag{9}$$

$$y_t = x_t + v_t, \quad v_t \sim \mathcal{N}(0, R_t) \tag{10}$$

where x_t represents the true wind speed state, y_t denotes the FIR-filtered observations, and R_t captures time-varying measurement uncertainty. The adaptive observation covariance is estimated through strictly causal local variance computation:

$$R_t = \mathrm{clip}\left(\mathrm{Var}[\{y_\tau\}_{\tau=\max(0,t-w+1)}^{t}], R_{\min}, R_{95}\right) \tag{11}$$

where $w = \max(5, N/200)$ defines the past-looking estimation window using only observations up to time t, ensuring no future information leakage. The bounds $R_{\min} = 10^{-8}$ and R_{95} (computed from the empirical distribution of past-only variance estimates) prevent numerical instability. The process noise covariance is initialized as $Q = 0.05 \cdot \mathrm{median}(R_{1:w_{\mathrm{init}}})$ based on an early segment to maintain temporal continuity.

The forward Kalman filter provides causal state estimates through sequential prediction-update cycles without backward smoothing:

$$\text{Predict:} \quad \hat{x}_{t|t-1} = \hat{x}_{t-1|t-1}, \quad P_{t|t-1} = P_{t-1|t-1} + Q \tag{12}$$

$$\text{Update:} \quad K_t = P_{t|t-1}(P_{t|t-1} + R_t)^{-1} \tag{13}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - \hat{x}_{t|t-1}) \tag{14}$$

yielding the final filtered series $W_{\mathrm{final}}(t) = \hat{x}_{t|t}$ that maintains strict causality for forecasting applications.

3) FILTERING PERFORMANCE CHARACTERISTICS

This hybrid filtering approach addresses multiple noise sources while preserving temporal causality through its sequential architecture, as presented in Figure 3. The causal FIR cascade removes high-frequency measurement noise and sensor artifacts while preserving meteorological signals across turbulent to synoptic timescales without introducing non-causal smoothing.

The subsequent forward-only adaptive Kalman filtering eliminates imputation discontinuities and residual noise through probabilistic state estimation that adjusts to past signal characteristics without backward propagation. The combination achieves superior performance for forecasting applications compared to non-causal smoothing methods by ensuring that filtered values at time t depend exclusively on information available up to time t, eliminating data leakage while maintaining effective noise reduction through the complementary strengths of causal frequency-domain filtering and optimal forward state-space estimation.

IV. METHODOLOGY

A. RANDOM VECTOR FUNCTIONAL LINK NETWORKS WITH NEURAL ARCHITECTURE SEARCH

The proposed framework extends RVFL networks through the integration of Neural Architecture Search (NAS) [30],

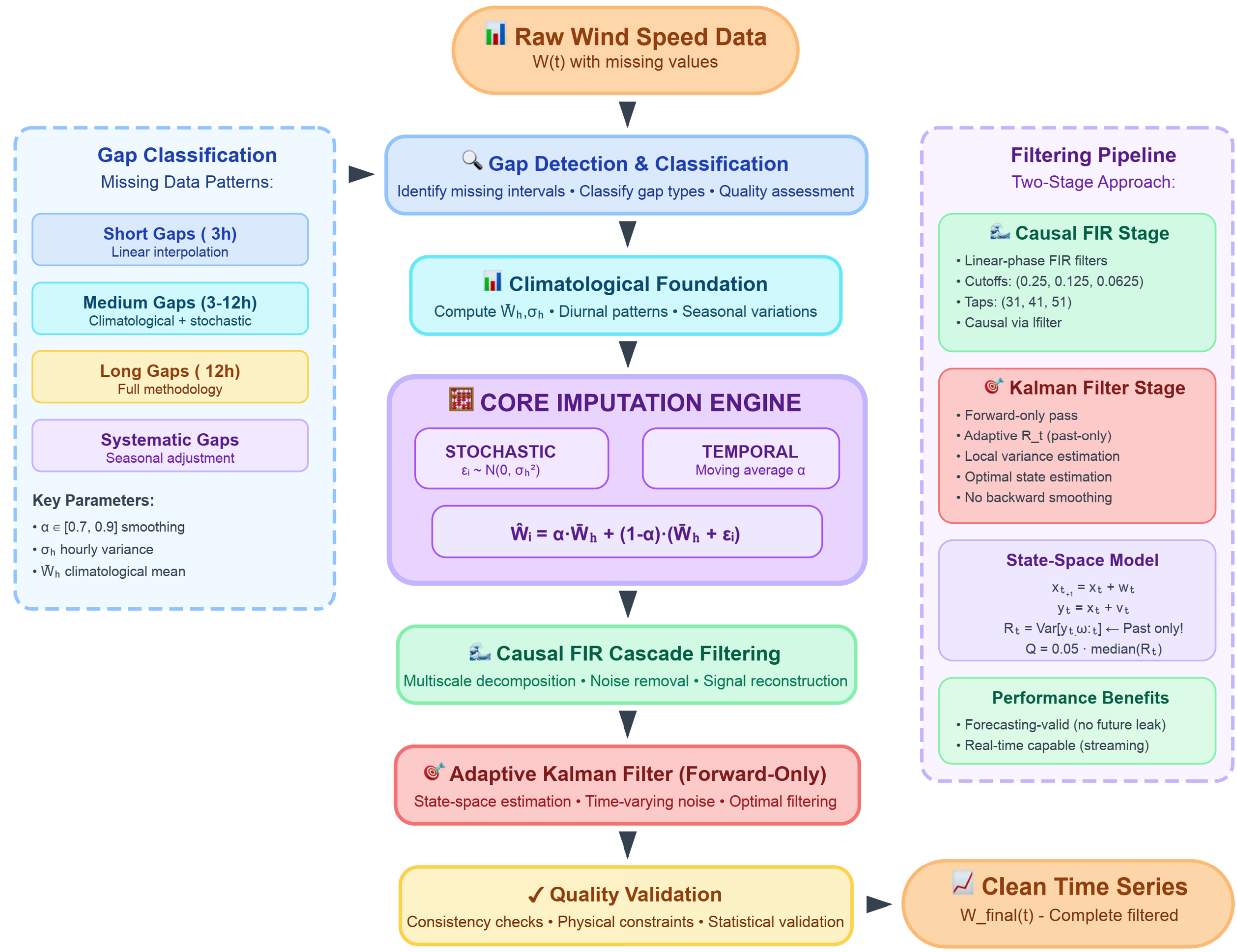


FIGURE 3. Flowchart of the proposed hybrid filtering approach.

multi-scale feature generation techniques, and adaptive weight initialization strategies. Unlike traditional neural networks that require iterative gradient-based training, RVFL networks employ a paradigm where random hidden layer weights remain fixed after initialization, and only output weights are determined through closed-form linear regression. This approach eliminates backpropagation and gradient computation, providing computational efficiency and theoretical guarantees for global optimality in the output layer.

1) PROBLEM FORMULATION

The wind speed forecasting problem is formulated as a supervised learning task where the objective is to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ from historical wind observations $\mathcal{X} \subseteq \mathbb{R}^{d_{\text{in}}}$ to future wind speeds $\mathcal{Y} \subseteq \mathbb{R}^{d_{\text{out}}}$ without backpropagation iterative training. The RVFL approach decomposes this problem into two distinct phases:

a: PHASE 1: FIXED RANDOM FEATURE GENERATION

Random transformations $\phi : \mathbb{R}^{d_{\text{in}}} \rightarrow \mathbb{R}^{d_{\text{total}}}$ map inputs to high-dimensional feature spaces using frozen parameters θ_{fixed}:

$$H = \phi(X; \theta_{\text{fixed}}) \in \mathbb{R}^{n \times d_{\text{total}}}. \tag{15}$$

b: PHASE 2: CLOSED-FORM OUTPUT WEIGHT COMPUTATION

Given the fixed feature representation, output weights are computed analytically through ridge regression:

$$\beta^* = \arg\min_{\beta} \|H\beta - y\|_F^2 + \lambda\|\beta\|_F^2. \tag{16}$$

The closed-form solution is:

$$\beta^* = (H^T H + \lambda I)^{-1} H^T y. \tag{17}$$

This formulation replaces gradient-based training with direct matrix operations. The expected risk minimization becomes:

$$\mathcal{R}(f) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(\phi(x; \theta_{\text{fixed}})^T \beta^*, y)] \tag{18}$$

where optimization focuses solely on architectural choices encoded in θ_{fixed}.

2) ARCHITECTURE PARADIGM

The RVFL architecture operates under a paradigm that fundamentally differs from conventional neural networks [31]:

- **Fixed Random Hidden Layer**: Weight matrices $W \in \mathbb{R}^{d_{\text{in}} \times d_{\text{hidden}}}$ and bias vectors $b \in \mathbb{R}^{d_{\text{hidden}}}$ are randomly initialized once and remain frozen throughout the entire process
- **Closed-Form Output Layer**: Output weights $\beta \in \mathbb{R}^{d_{\text{total}} \times d_{\text{out}}}$ are computed analytically through ridge regression
- **No Backpropagation**: The absence of gradient-based training eliminates the need for backpropagation and automatic differentiation

This approach provides several theoretical and computational advantages:

- **Global Optimality**: The linear regression formulation guarantees globally optimal output weights given the fixed random features
- **Computational Efficiency**: Elimination of backpropagation loops and gradient computations
- **Theoretical Tractability**: Well-defined mathematical properties and convergence guarantees
- **Hyperparameter Focus**: Optimization effort concentrates on architectural choices rather than training dynamics

B. NEURAL ARCHITECTURE SEARCH FRAMEWORK

1) NAS CONTROLLER DESIGN

The NAS controller employs Tree-structured Parzen Estimator (TPE) sampling combined with Hyperband pruning to optimize the configuration of fixed random transformations [32]. Unlike conventional NAS that optimizes trainable architectures, the proposed approach searches over the space of random feature generation strategies. For each optimization trial t, the controller generates a binary gating vector $\mathbf{g}^{(t)} \in \{0, 1\}^K$ where K represents the number of available feature families:

$$\mathbf{g}^{(t)} = [\text{gate}_{\text{ntk}}, \text{gate}_{\text{nystrom}}, \text{gate}_{\text{fastfood}}, \text{gate}_{\text{attention}}, \text{gate}_{\text{wavelets}}, \text{gate}_{\text{multi-kernel}}]^T. \quad (19)$$

The critical distinction from conventional NAS lies in the search space: rather than optimizing trainable network weights, the controller searches over random initialization strategies, activation function combinations, and feature transformation parameters that remain fixed once selected [33]. The controller maintains posterior distributions over architectural configurations using Bayesian updating:

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta) \quad (20)$$

where $\mathcal{D} = \{(\theta_i, f(\theta_i))\}_{i=1}^n$ represents the optimization history.

2) STATIC AND DYNAMIC FEATURE ARCHITECTURE

The feature generation strategy partitions transformations into static and dynamic categories based on computational dependencies, with all components remaining fixed after initialization. Static features $H_{\text{static}}(x)$ are precomputed once per fit using fixed random parameters and cached:

$$H_{\text{static}}(x) = [H_{\text{kernel}}(x); H_{\text{nystrom}}(x); H_{\text{fastfood-cached}}(x)]. \quad (21)$$

Dynamic features $H^{(t)}_{\text{dynamic}}(x)$ are computed per trial based on current fixed weight parameters $W^{(t)}, b^{(t)}$ determined by the trial's architectural choices, but these parameters remain frozen throughout the evaluation of that trial:

$$H^{(t)}_{\text{dynamic}}(x) = [\mathbf{g}^{(t)}_{\text{ntk}} \odot H_{\text{ntk}}(x, W^{(t)}, b^{(t)}); \mathbf{g}^{(t)}_{\text{attention}} \odot H_{\text{attention}}(x); \mathbf{g}^{(t)}_{\text{wavelets}} \odot H_{\text{wavelets}}(x)] \quad (22)$$

where $\odot$ denotes element-wise gating, and the specific feature components depend on trial-specific architectural decisions.

C. WEIGHT INITIALIZATION STRATEGIES

1) DATA-DRIVEN INITIALIZATION METHODS

The framework implements multiple strategies for generating fixed random weights that remain unchanged throughout the learning process. Unlike conventional neural networks, where initialization serves as a starting point for gradient-based optimization, RVFL initialization directly determines the final hidden layer transformation. For training data $X \in \mathbb{R}^{n \times d_{\text{in}}}$, the PCA-aligned orthogonal initialization begins with data centering:

$$X_{\text{centered}} = X - \frac{1}{n}\mathbf{1}\mathbf{1}^T X. \quad (23)$$

which subtracts the sample mean from every row of X because $(1/n)\mathbf{1}\mathbf{1}^T X$ replicates the mean vector across all rows.

Principal components are extracted using randomized SVD:

$$U, S, V^T = \text{rSVD}(X_{\text{centered}}, k = \min(d_{\text{hidden}}, \text{rank}(X_{\text{centered}}))). \quad (24)$$

The complete weight matrix combines data-driven directions with orthogonal random components:

$$W = [V_{\text{pca}}; \text{Orth}(\mathcal{N}(0, I)^{d_{\text{in}} \times (d_{\text{hidden}} - k)})] \quad (25)$$

where $\text{Orth}(\cdot)$ denotes QR-based orthogonalization against existing subspace directions.

2) ZCA WHITENING INITIALIZATION

For datasets exhibiting correlated input features, ZCA (Zero-phase Component Analysis) whitening provides an alternative strategy for generating fixed random projections. The ZCA transformation matrix is computed once and remains fixed:

$$W_{\text{ZCA}} = U\Lambda^{-1/2}U^T \quad (26)$$

where $(U, \Lambda) = \text{eig}(\text{Cov}(X) + \epsilon I)$ with regularization parameter $\epsilon = 10^{-5}$ for numerical stability.

3) SPECTRAL NORMALIZATION

Spectral normalization constrains the largest singular value of fixed-weight matrices to ensure bounded transformations:

$$W_{\text{normalized}} = \frac{\sigma_{\text{target}}}{\sigma_{\max}(W)} W \tag{27}$$

where $\sigma_{\max}(W)$ is approximated through power iteration:

$$\sigma_{\max}(W) \approx \lim_{k \to \infty} u_k^T W v_k \tag{28}$$

with iterative updates $u_{k+1} = \frac{W v_k}{\|W v_k\|_2}$ and $v_{k+1} = \frac{W^T u_k}{\|W^T u_k\|_2}$.

D. MULTI-SCALE FEATURE GENERATION

All feature generation techniques operate using fixed random parameters determined during initialization, eliminating the need for iterative parameter updates. The random parameters are drawn from appropriate distributions and remain frozen throughout the learning process.

1) RANDOM FOURIER FEATURES FOR KERNEL APPROXIMATION

The framework approximates RBF kernels through random Fourier features using fixed random frequencies and phases. The frequency parameters $\omega_i \sim \mathcal{N}(0, 2\gamma I)$ and phase shifts $b_i \sim \text{Uniform}[0, 2\pi]$ are sampled once during initialization and remain fixed. For kernel function $k(x, x') = \exp(-\gamma \|x - x'\|^2)$, the approximation using fixed parameters is:

$$k(x, x') \approx \frac{2}{m} \sum_{i=1}^{m} \cos(\omega_i^T x + b_i) \cos(\omega_i^T x' + b_i), \tag{29}$$

where x' denotes a second input vector, distinct from x, against which the kernel similarity is evaluated.

The corresponding feature transformation uses the frozen frequency and phase parameters:

$$H_{\text{kernel}}(x) = \sqrt{\frac{2}{m}} [\cos(\Omega x + b); \sin(\Omega x + b)]. \tag{30}$$

2) MULTI-KERNEL INTEGRATION

Meteorological phenomena exhibit patterns across multiple scales, motivating the use of multiple kernel types. The multi-kernel approach combines kernel approximations:

$$H_{\text{multi-kernel}}(x) = [H_{\text{rbf}}(x); H_{\text{matern}}(x); H_{\text{laplace}}(x)] \tag{31}$$

where each kernel type captures different aspects of temporal correlations with fixed characteristic scales.

3) NYSTRÖM KERNEL APPROXIMATION

The Nyström method constructs kernel approximations using landmark points $\{x_j^{(\ell)}\}_{j=1}^m$ selected through random sampling and fixed after initialization [34]. The approximation decomposes the kernel matrix as $K \approx K_{nm} K_{mm}^{-1} K_{mn}$, where $K_{nm}[i, j] = k(x_i, x_j^{(\ell)})$ and $K_{mm}[i, j] = k(x_i^{(\ell)}, x_j^{(\ell)})$.

The Nyström features are computed using these frozen components:

$$H_{\text{nystrom}}(x) = \sqrt{m} \cdot K_{xm} K_{mm}^{-1/2} \tag{32}$$

where $K_{mm}^{-1/2} = U \Lambda^{-1/2} U^T$ with eigendecomposition $(U, \Lambda) = \text{eig}(K_{mm})$ and eigenvalue filtering $\lambda_i > 10^{-8}$.

4) STRUCTURED RANDOM PROJECTIONS VIA FASTFOOD TRANSFORM

The Fastfood transform provides $O(d \log d)$ computational complexity through structured matrix operations using fixed random components:

$$H_{\text{fastfood}}(x) = \frac{1}{\sqrt{d}} S \odot \text{FFT}(B \odot \Pi(x)) \tag{33}$$

where FFT is the Fast Fourier Transform operation. All transformation components are fixed during initialization, $B \in \{-1, +1\}^{d_{\text{in}}}$ fixed Rademacher random variables, Π fixed random permutation matrix, and $S \sim \mathcal{N}(0, 1)^{d_{\text{in}}}$ fixed Gaussian scaling vector.

5) NEURAL TANGENT KERNEL FEATURES

NTK features approximate the infinite-width neural network limit through gradient-based transformations using fixed random weights [35]. For fixed activation function σ and frozen pre-activation parameters $z = Wx + b$, the NTK features are:

$$H_{\text{ntk}}(x) = \alpha_{\text{ntk}} \cdot \text{diag}(\sigma'(z)) W^T x. \tag{34}$$

For Gaussian Error Linear Units (GELUs) activation, the derivative is computed as:

$$\begin{aligned}\sigma'_{\text{gelu}}(z) &= \frac{1}{2}(1 + \tanh(\Phi(z))) \\ &\quad + \frac{z}{2}(1 - \tanh^2(\Phi(z)))\, \phi\, (1 + 3 \cdot 0.044715\, z^2)\end{aligned} \tag{35}$$

where $\Phi(z) = \sqrt{\frac{2}{\pi}}(z + 0.044715\, z^3)$ and $\phi = \sqrt{\frac{2}{\pi}}$. The constant 0.044715 is not arbitrary; it was obtained empirically when fitting a `tanh`-based approximation to the exact Gaussian cumulative distribution function.

6) SELF-ATTENTION MECHANISMS

The framework incorporates lightweight self-attention to capture long-range dependencies using fixed random projection matrices. For input $x \in \mathbb{R}^{d_{\text{in}}}$, the attention mechanism computes transformations using frozen parameters:

$$H_{\text{attention}}(x) = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{36}$$

where $Q = xW_Q$, $K = xW_K$, $V = xW_V$ with fixed random projection matrices $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{in}} \times d_k}$ sampled once during initialization [36].

7) WAVELET SCATTERING TRANSFORM

Wavelet scattering transforms provide translation-invariant representations using predetermined wavelet bases and scales. The transform uses fixed wavelet parameters ψ_j at predetermined scales and fixed averaging filter ϕ_J:

$$S_J[p]f = |f \star \phi_J| \quad \text{and}$$
$$S_J[p,j]f = ||f \star \psi_{j_1}| \star \psi_{j_2} \cdots \star \psi_{j_p}| \star \phi_J. \tag{37}$$

The wavelet basis functions and scale parameters are fixed during initialization, with the maximum scale $J = \lfloor \log_2(d_{\text{in}}) \rfloor$ determined automatically based on input dimensionality.

E. ENHANCED DIRECT LINK ARCHITECTURE

The direct link component employs fixed linear transformations to capture input-output relationships, which are determined during initialization:

$$H_{\text{direct}}(x) = (xU)V^T + b_{\text{direct}} \tag{38}$$

where $U \in \mathbb{R}^{d_{\text{in}} \times r}$, $V \in \mathbb{R}^{d_{\text{out}} \times r}$ with rank $r = \min(32, d_{\text{in}})$, and $b_{\text{direct}} \in \mathbb{R}^{d_{\text{out}}}$ are all randomly initialized once and remain frozen.

To ensure balanced contributions across feature types, the framework applies Root Mean Square (RMS)-based normalization:

$$\text{RMS}(H) = \sqrt{\frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d} H_{ij}^2}. \tag{39}$$

The normalization scaling factor is computed as:

$$\alpha_{\text{scale}} = \frac{\text{RMS}(H_{\text{hidden}})}{\text{RMS}(H_{\text{direct}}) + \epsilon} \cdot \gamma_{\text{direct}} \tag{40}$$

where $\epsilon = 10^{-8}$ prevents division by zero and γ_{direct} is the direct link scaling parameter.

F. BLOCK NORMALIZATION AND FEATURE INTEGRATION

Feature blocks are normalized to prevent dominance of high-magnitude transformations. For feature blocks $\{H^{(k)}\}_{k=1}^{K}$, the global RMS reference is:

$$\text{RMS}_{\text{global}} = \frac{1}{K}\sum_{k=1}^{K} \text{RMS}(H^{(k)}). \tag{41}$$

Each block is rescaled as:

$$\tilde{H}^{(k)} = H^{(k)} \cdot \text{clamp}\left(\frac{\text{RMS}_{\text{global}}}{\text{RMS}(H^{(k)}) + \epsilon}, 0.5, 2.0\right). \tag{42}$$

A stochastic soft gating mechanism is applied to each normalized block:

$$H_{\text{gated}}^{(k)} = \tilde{H}^{(k)} \cdot (0.9 + 0.2 \cdot \sigma(\xi^{(k)})) \tag{43}$$

where $\xi^{(k)} \sim \mathcal{N}(0, 1)$ provides stochastic gating and σ denotes the sigmoid function.

G. OUTPUT WEIGHT COMPUTATION

The core advantage of the RVFL paradigm lies in the analytical determination of output weights through closed-form linear regression, completely eliminating backpropagation optimization procedures. Given the fixed feature representation $H \in \mathbb{R}^{n \times d_{\text{total}}}$ constructed from frozen random transformations, the output weights are computed analytically. The ridge regression objective:

$$\beta^* = \arg\min_{\beta} \|H\beta - y\|_F^2 + \lambda\|\beta\|_F^2 \tag{44}$$

admits the direct closed-form solution:

$$\beta^* = (H^T H + \lambda I)^{-1} H^T y. \tag{45}$$

This formulation guarantees optimality for the output layer given the fixed random features. The framework implements Graphics Processing Unit (GPU)-accelerated computation of the closed-form solution through adaptive matrix decomposition methods. The solver selection criterion is:

$$\text{Method} = \begin{cases} \text{Normal Equations} & \text{if } n \geq d_{\text{total}} \\ \text{SVD Direct} & \text{if } n < d_{\text{total}}. \end{cases} \tag{46}$$

For the SVD-based approach, the analytical solution uses singular value decomposition:

$$\beta^* = V \text{diag}\left(\frac{s_i}{s_i^2 + \lambda}\right) U^T y \tag{47}$$

where $(U, s, V^T) = \text{SVD}(H)$ decomposes the fixed feature matrix H.

For large-scale problems, the closed-form computation implements chunked accumulation while maintaining the analytical property. The normal equations are accumulated without iterative parameter updates:

$$H^T H_{\text{acc}} = \sum_{i=1}^{C} H_i^T H_i \tag{48}$$

$$H^T y_{\text{acc}} = \sum_{i=1}^{C} H_i^T y_i \tag{49}$$

where C denotes the number of chunks. The final solution is computed directly:

$$\beta^* = (H^T H_{\text{acc}} + \lambda I)^{-1} H^T y_{\text{acc}}. \tag{50}$$

When non-smooth regularization is required, the Alternating Direction Method of Multipliers (ADMM) solver maintains the analytical property by avoiding gradient-based updates of the random features. The algorithm optimizes only the output weights:

$$\beta^{(k+1)} = (H^T H + \rho I)^{-1}(H^T y + \rho(z^{(k)} - u^{(k)})) \tag{51}$$
$$z^{(k+1)} = \text{prox}_{\lambda/\rho}(\beta^{(k+1)} + u^{(k)}) \tag{52}$$
$$u^{(k+1)} = u^{(k)} + \beta^{(k+1)} - z^{(k+1)} \tag{53}$$

where the feature matrix H remains fixed throughout all ADMM iterations.

H. AUTOMATIC MIXED PRECISION OPTIMIZATION

The framework employs Automatic Mixed Precision (AMP) to accelerate the closed-form computations while preserving numerical stability. Feature generation from fixed random transformations uses half-precision for forward computations:

$$H_{\text{fp16}} = \phi(\text{FP16}(x) \cdot \text{FP16}(W) + \text{FP16}(b)). \tag{54}$$

The closed-form ridge regression maintains full precision for numerical stability:

$$\beta^* = \text{FP32}((H^T H + \lambda I)^{-1} H^T y). \tag{55}$$

The numbers in FP16 and FP32 come directly from the number of bits used to represent each floating point value in hardware. Since no gradient computation or backpropagation occurs, the AMP implementation focuses entirely on accelerating matrix operations for feature generation and analytical solution computation.

I. HYPERPARAMETER OPTIMIZATION FRAMEWORK

The hyperparameter optimization focuses exclusively on architectural choices for the fixed random transformations, since no backpropagation training parameters exist. The hyperparameter optimization encompasses only architectural and initialization parameters since no training dynamics exist. Table 1 presents the complete search space for the NAS optimization. All parameters determine the configuration of fixed random transformations rather than trainable model weights.

The TPE optimizes the configuration of fixed random architectures rather than training trajectories [37]. For each trial, TPE evaluates the performance of a complete fixed architecture without any iterative parameter updates [38]. The algorithm maintains separate distributions for architectures yielding objective values below and above a threshold:

$$\ell(\theta_{\text{arch}}) = p(\theta_{\text{arch}} | f(\theta_{\text{arch}}) < f^*) \tag{56}$$
$$g(\theta_{\text{arch}}) = p(\theta_{\text{arch}} | f(\theta_{\text{arch}}) \geq f^*) \tag{57}$$

where θ_{arch} represents the complete specification of the fixed random architecture and f^* is the γ-quantile threshold [39].

The acquisition function guides the search over architectural configurations:

$$\text{EI}(\theta_{\text{arch}}) = \frac{\ell(\theta_{\text{arch}})}{g(\theta_{\text{arch}})}. \tag{58}$$

Early termination of underperforming fixed architectures employs Hyperband's successive halving without requiring training checkpoints [40]. Since each trial involves only closed-form computation, pruning decisions are based on validation performance after complete feature generation and ridge regression solution:

$$\mathcal{A}^{(r+1)} = \text{TopK}\left(\mathcal{A}^{(r)}, \left\lfloor \frac{|\mathcal{A}^{(r)}|}{\eta} \right\rfloor\right) \tag{59}$$

where $\mathcal{A}^{(r)}$ represents the set of architectural configurations at evaluation round r, $\eta = 3$ denotes the elimination ratio, and TopK selects architectures with lowest validation loss.

J. PERFORMANCE EVALUATION AND ANALYSIS

Model performance is evaluated using multiple regression metrics: MAE, RMSE, Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2), given by:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{60}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{61}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{62}$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}. \tag{63}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the total number of observations [41].

The contribution of different feature families is quantified through output weight analysis:

$$\text{Importance}_k = \frac{\|\beta_k\|_F}{\sum_{j=1}^{K} \|\beta_j\|_F} \tag{64}$$

where β_k represents the weight submatrix corresponding to feature family k and $\| \cdot \|_F$ denotes the Frobenius norm.

K. COMPUTATIONAL COMPLEXITY ANALYSIS

The algorithmic complexity analysis highlights the efficiency of the proposed approach, primarily attributed to the elimination of backpropagation training procedures. From a time complexity perspective, the computational cost is significantly reduced due to the use of fixed transformations and closed-form solutions. The feature generation process requires only a single pass, with a complexity of $O(nd\_ind\_total)$, where n denotes the number of samples, d_in the input dimensionality, and d_total the total number of generated features.

The model fitting step relies on a closed-form solution, incurring a cost of $O(\min(nd\_total^2, n^2 d\_total) + d\_total^3)$ due to matrix operations involved in pseudo-inverse computation and related procedures. Additionally, during the architecture search phase, the total cost is given by $O(T \cdot C\_fixed)$, where T represents the number of optimization trials and C_fixed denotes the constant cost of evaluating each fixed architecture.

In terms of space complexity, the algorithm maintains a predictable and bounded memory footprint. The transformed features are stored in a matrix of size $O(nd\_total)$, while the fixed random weights used for transformation are stored with a complexity of $O(d\_ind\_hidden)$. A static cache of size $O(n \cdot d\_static)$ is also maintained to store precomputed features, further improving runtime efficiency. Importantly,

TABLE 1. Hyperparameter search space for NAS-based RVFL optimization.

Category	Parameter	Symbol	Distribution
Architecture Parameters	Weight initialization	-	Categorical: {uniform, orthogonal, pca_orthogonal, zca_whitened, he, xavier}
	Random scaling factor	s	LogUniform(0.1, 2.0)
	Ridge regularization	λ	LogUniform(10^{-6}, 10^{-2})
	Feature dropout rate	p_{drop}	Uniform(0.0, 0.5)
Feature Family Gates	NTK feature inclusion	g_{ntk}	Categorical({0, 1})
	Attention feature inclusion	$g_{\text{attention}}$	Categorical({0, 1})
	Wavelet feature inclusion	g_{wavelets}	Categorical({0, 1})
	Nyström feature inclusion	g_{nystrom}	Categorical({0, 1})
Feature-Specific Parameters	NTK scaling coefficient	α_{ntk}	LogUniform(0.1, 10.0) if $g_{\text{ntk}} = 1$
	Attention projection dim	d_k	DiscreteUniform(16, 128) if $g_{\text{attention}} = 1$
	Nyström landmark count	m	DiscreteUniform(20, 150) if $g_{\text{nystrom}} = 1$

the paradigm avoids the computational overhead typically associated with gradient-based learning. Specifically, there is no need for gradient computation and backpropagation. Likewise, convergence monitoring is entirely absent from the process, further underscoring the minimal computational demands of the method.

The static feature caching mechanism reduces the computational cost per optimization trial from $O(nd_{\text{total}})$ to $O(nd_{\text{dynamic}})$ where $d_{\text{dynamic}} \ll d_{\text{total}}$, providing computational speedup ratios of approximately 10-20x compared to conventional neural architecture search approaches that require full training for each candidate architecture.

L. IMPLEMENTATION AND NUMERICAL STABILITY

To ensure robust performance across diverse datasets, several numerical stability measures were incorporated into the implementation. Eigenvalue thresholding was applied with a lower bound of $\epsilon = 10^{-8}$ during matrix decompositions to avoid computational issues arising from near-singular values. This approach helps maintain numerical precision when dealing with low-rank or nearly degenerate matrices. The condition number of feature matrices was continuously monitored to detect and address potential issues stemming from ill-conditioning, which could adversely affect the convergence and accuracy of the model.

To further mitigate such risks, regularization parameters were carefully bounded within predefined limits, ensuring that model fitting procedures remained stable across different input distributions. Matrix scaling techniques were also employed prior to decomposition operations to improve matrix conditioning, thereby enhancing the reliability and accuracy of the resulting computations. Together, these strategies contribute to a stable and generalizable learning process, even in scenarios involving challenging numerical characteristics.

M. MULTI-ACTIVATION STRATEGY

To enhance the expressiveness of random projections, the framework supports diverse activation functions across hidden units. In multi-activation mode, different hidden neurons employ distinct activation functions:

$$H_{\text{std}}(x)[i] = \sigma_{\text{act}[i]}((Wx + b)[i]) \tag{65}$$

where act$[i]$ is randomly sampled from the activation set {GELU, Swish, Mish, Tanh, LeakyReLU}, providing diverse nonlinear transformations that capture different aspects of the input patterns.

Figure 4 illustrates the complete methodology framework, showing the integration of NAS-controlled selection of fixed random architectures, static caching optimization for frozen transformations, and closed-form solution computation without backpropagation. This plot shows the static feature caching of fixed random transformations, dynamic feature generation with NAS gating, block normalization, and a GPU-accelerated closed-form ridge regression solution for wind speed forecasting applications. The diagram emphasizes the absence of backpropagation training loops and the direct analytical computation of output weights.

An overview of the high-level data flow pipeline is presented in Figure 5.

V. RESULTS AND DISCUSSION

The experimental evaluation of the proposed RVFL-OptBayes architecture was conducted on a wind speed forecasting task using a univariate time series dataset. The forecasting problem was formulated as a supervised learning task with 6 lagged input features $(x_{t-23}, x_{t-22}, \ldots, x_t)$ predicting one step ahead (x_{t+1}). This configuration provides the model with approximately one day of historical context to predict the subsequent time step, enabling the capture of both short-term fluctuations and daily periodicities inherent in wind speed patterns.

The dataset was partitioned into training and validation sets using a temporal split, with approximately 10,600 time steps (from 2020-03-01 00:00:00 to 2020-05-13 14:10:00) allocated for training and the remaining steps (from 2020-05-13 14:20:00 to 2020-05-31 23:50:00) reserved for evaluation, with a sampling time of 10 min. This split ensures that the model's forecasting performance is assessed on future unseen data, maintaining the temporal integrity essential for time series validation. All input features were standardized to zero mean and unit variance to ensure numerical stability across the diverse feature transformations within the RVFL architecture. Notice that the metrics presented are all calculated on scaled data.

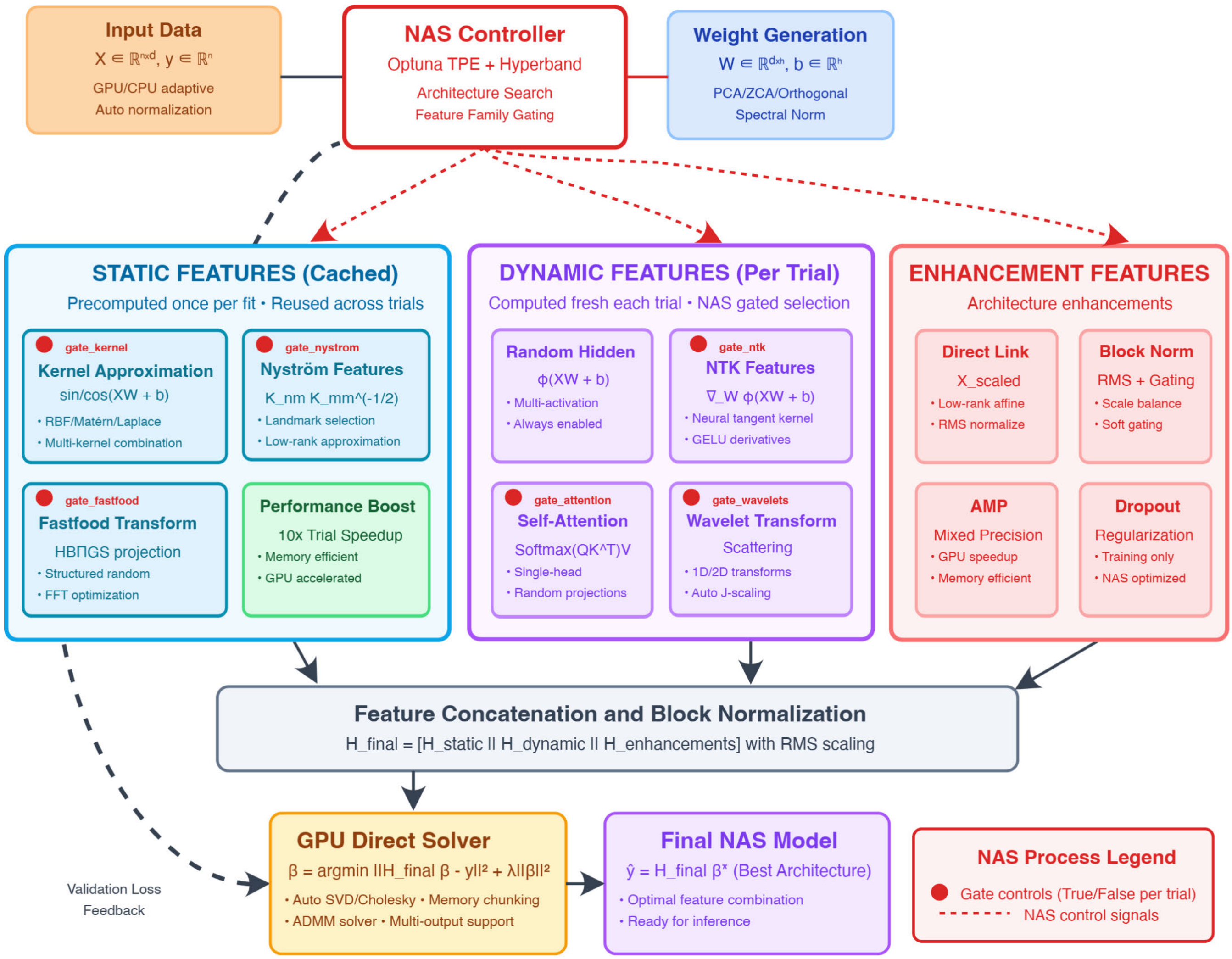


FIGURE 4. Architecture diagram of the RVFL framework with neural architecture search integration.

The experimental analysis encompasses four key aspects: an ablation study to quantify the contribution of individual architectural components, a Bayesian optimization analysis to understand hyperparameter sensitivity, a time series forecasting evaluation demonstrating practical performance, and a comprehensive model comparison against established baseline approaches. These experiments collectively validate the effectiveness of the proposed hybrid architecture and provide insights into the mechanisms driving its performance.

A. ABLATION STUDY

To evaluate the contribution of individual components within the proposed RVFL-OptBayes architecture, we conducted an ablation study by systematically removing each feature type while maintaining all other components. The baseline configuration includes all features: direct link connections, attention-based features, wavelet features, kernel approximations, Nyström features, spectral normalization, NTK features, Fastfood transforms, and multi-kernel approximations. Each ablation experiment removes one component and measures the resulting performance degradation.

Table 2 presents the ablation study results ranked by performance degradation (ΔMSE). The baseline model achieves an MSE of 0.000292, RMSE of 0.017094, MAE of 0.013322, and R^2 of 0.9996. The direct link connection shows the largest impact on model performance, with its removal resulting in $\Delta\text{MSE} = 0.002294$ and R^2 decreasing from 0.9996 to 0.9960. This indicates that the direct connection between input and output layers provides linear modeling capacity that complements the nonlinear transformations in the hidden layer.

Fastfood features represent the second most affected component, with removal causing $\Delta\text{MSE} = 0.001012$ and R^2 decreasing to 0.9980. The Fastfood transform provides efficient random feature approximations that contribute substantially to the model's representational capacity. Wavelet features produce $\Delta\text{MSE} = 0.000455$, providing time-frequency domain representations that enhance the model's ability to capture temporal patterns in the input data.

NTK features contribute $\Delta\text{MSE} = 0.000114$, capturing neural network behavior in the infinite-width limit within the RVFL framework. Attention-based features show

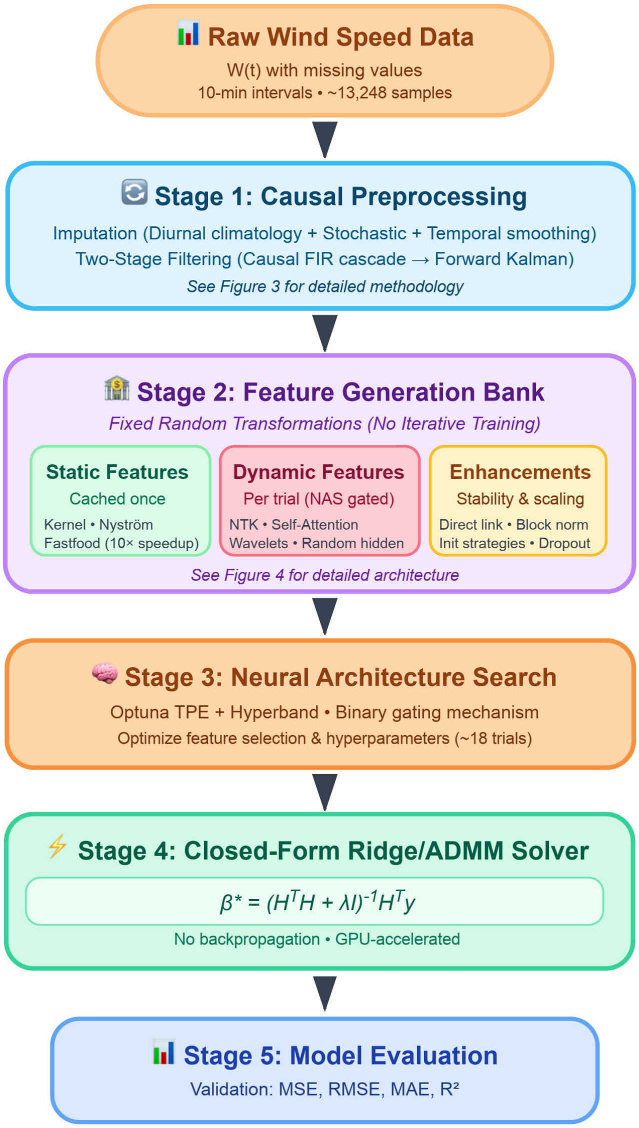


FIGURE 5. Architecture diagram of the RVFL framework with neural architecture search integration.

ΔMSE $= 0.000075$, with the attention mechanism selectively weighting input features to enhance representational capacity. Spectral normalization (ΔMSE $= 0.000029$) provides measurable improvements to model stability.

Multi-kernel approximations (ΔMSE $= 0.000021$) and Nyström features (ΔMSE $= 0.000019$) contribute smaller but consistent improvements to performance. The kernel approximation features exhibit minimal impact (ΔMSE $= 0.000005$), indicating that for this task, the basic kernel approximations provide limited additional representational capacity beyond the other feature types.

The computational overhead is relatively consistent across components, with training times ranging from 0.3 to 0.9 seconds. This analysis reveals a hierarchy of feature importance, with direct connections providing the dominant performance contribution, Fastfood transforms and wavelet-based features contributing substantial improvements, NTK and attention mechanisms providing moderate refinements, and spectral normalization along with kernel-based features offering smaller but measurable enhancements.

B. BAYESIAN OPTIMIZATION SUBPROBLEM

The hyperparameter optimization process employed Optuna based on TPE to explore the hyperparameter space for the RVFL-OptBayes model. Figure 6 shows the convergence behavior of the Bayesian optimization process over 18 trials.

The optimization process begins with an initial trial achieving an objective value of approximately 2.5×10^{-4}. The most significant convergence occurs within the first trial, establishing a baseline performance level. Throughout the search process, the optimization exhibits exploration with individual trials spanning a range of objective values from approximately 2.3×10^{-4} to 1.8×10^{-3}, demonstrating the TPE algorithm's exploration-exploitation balance in navigating the hyperparameter landscape. Exploration peaks occur at trials 8 and 18, where the algorithm tests configurations with higher objective values. The best objective value improves over the course of optimization, converging to approximately 2.3×10^{-4} by the final trials, with the red curve showing refinement of the optimal configuration.

Figure 7 presents the relative importance of each hyperparameter in determining model performance. The scale_direct parameter shows the highest importance with a fraction of approximately 0.20, indicating that direct link scaling has the largest impact on model performance. The dropout parameter ranks second with importance $\approx$ 0.18, followed by L2 regularization ($\approx$ 0.17) and gate_ntk ($\approx$ 0.12). The gate_nystrom parameter shows importance $\approx$ 0.10, while the general scale parameter contributes $\approx$ 0.08. The method selection exhibits importance $\approx$ 0.06, gate_fastfood shows $\approx$ 0.04, and gate_wavelets contributes $\approx$ 0.03. The remaining hyperparameters, use_multi_kernel and gate_attention, each contribute less than 0.02 to the overall importance.

The hyperparameter importance analysis reveals that direct link scaling dominates the optimization landscape, with regularization parameters (dropout and L2) providing secondary contributions. The prominence of scale_direct aligns with the ablation study findings that direct connections provide the most critical performance contribution, and proper scaling of this component influences optimal model behavior. The importance of dropout (0.18) and L2 regularization (0.17) indicates that regularization strength influences model generalization. The importance of gate_ntk (0.12) and gate_nystrom (0.10) suggests these feature-gating mechanisms contribute to performance optimization. Unlike the ablation study, where wavelet features showed impact when removed, the relatively low importance of gate_wavelets (0.03) in hyperparameter optimization suggests the model performance is less sensitive to the specific gating strength of wavelets within the explored range. The minimal importance of use_multi_kernel ($<$ 0.02) and gate_attention ($<$ 0.02) indicates the model is robust to these hyperparameter choices.

TABLE 2. Ablation study results showing performance degradation when individual components are removed from the baseline RVFL-OptBayes model. Baseline performance: MSE = 0.000292, RMSE = 0.017094, MAE = 0.013322, R^2 = 0.9996. Results are ranked by MSE degradation (ΔMSE).

Removed Feature	MSE	ΔMSE	RMSE	ΔRMSE	R^2	Fit Time (s)
Direct Link	0.002587	0.002294	0.050859	0.033764	0.9960	0.6
Fastfood Features	0.001304	0.001012	0.036115	0.019020	0.9980	0.5
Wavelet Features	0.000747	0.000455	0.027326	0.010232	0.9989	0.9
NTK Features	0.000406	0.000114	0.020161	0.003067	0.9994	0.5
Attention Features	0.000367	0.000075	0.019162	0.002067	0.9994	0.6
Spectral Normalization	0.000321	0.000029	0.017918	0.000824	0.9995	0.4
Multi-Kernel Approximation	0.000313	0.000021	0.017687	0.000593	0.9995	0.7
Nyström Features	0.000311	0.000019	0.017646	0.000551	0.9995	0.4
Kernel Approximation	0.000297	0.000005	0.017241	0.000146	0.9995	0.3

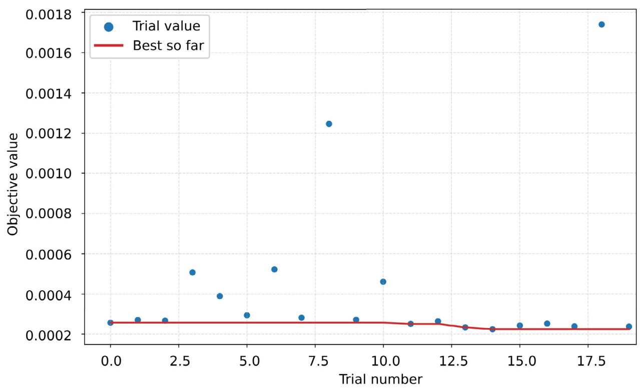


FIGURE 6. Bayesian optimization convergence showing objective value progression over 18 trials. The blue dots represent individual trial values, while the red line indicates the best objective value achieved up to each trial. The optimization exhibits exploration with trials ranging from 0.00023 to 0.0018, converging to an optimal MSE of approximately 0.00023.

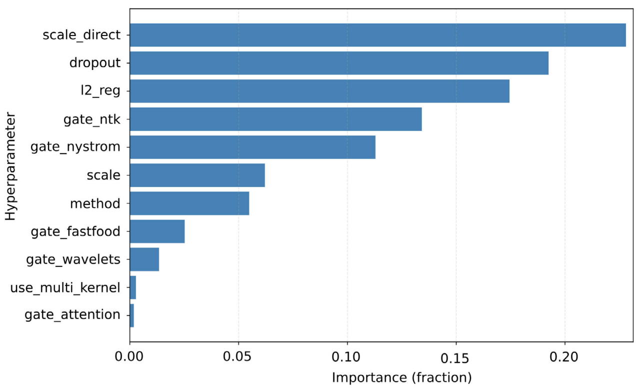


FIGURE 7. Hyperparameter importance analysis showing the relative contribution of each parameter to optimization performance. The scale_direct parameter exhibits the highest importance (≈0.20), followed by dropout (≈0.18) and L2 regularization (≈0.17), indicating that direct link scaling and regularization strength are the most critical factors for model performance.

The convergence pattern in Figure 6 and the hyperparameter importance distribution in Figure 7 demonstrate that TPE identified direct link scaling and regularization parameters as the critical optimization targets while maintaining exploration across the remaining hyperparameter space.

C. IMPORTANCE-ABLATION CONCORDANCE ANALYSIS

We aim to verify whether the *hyperparameter importances* obtained from Bayesian optimization (Optuna, fANOVA)

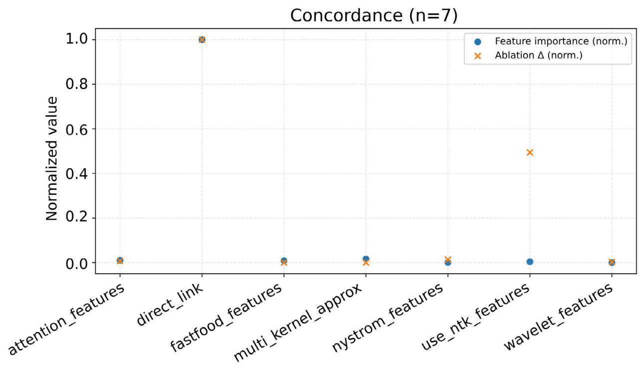


FIGURE 8. Normalized feature-level Optuna importance ($\hat{I}$) versus normalized ablation degradation ($\hat{\Delta}$). Bars are paired per feature; legend indicates which bars correspond to each source. Reported Spearman and Kendall coefficients summarize rank-level concordance over the common feature set.

are consistent with the *functional importance* revealed by a retrain ablation in which each feature family is disabled, and the model is refit. Intuitively, if a component matters to generalization, then (i) tuning its associated hyperparameters should be impactful (high fANOVA importance), and (ii) removing that component should hurt performance (large positive Δ on a minimize metric such as MSE).

We then *aggregate* parameter importances into feature-level scores by

$$\tilde{I}_f ;=; \sum_{p,:,f\in m(p)} I_p \qquad \text{for } f \in \mathcal{F}, \tag{66}$$

followed by min-max normalization over the common set $\mathcal{F}*\cap$ used in both analyses:

$$\hat{I}_f ;=; \frac{\tilde{I}_f - \min *g \in \mathcal{F}*\cap \tilde{I}_g}{\max *g \in \mathcal{F}*\cap \tilde{I}_g - \min *g \in \mathcal{F}*\cap \tilde{I}_g}, \tag{67}$$

$$\hat{\Delta}_f ;=; \frac{\Delta_f^{(+)} - \min *g \in \mathcal{F}*\cap \Delta_g^{(+)}}{\max *g \in \mathcal{F}*\cap \Delta_g^{(+)} - \min *g \in \mathcal{F}*\cap \Delta_g^{(+)}}. \tag{68}$$

Figure 8 juxtaposes normalized feature-level importances ($\hat{I}$) and normalized ablation deltas ($\hat{\Delta}$) and reports ρ_s and τ_b. In the present run we obtained limited rank agreement between the two signals, with NTK features presenting the highest disagreement.

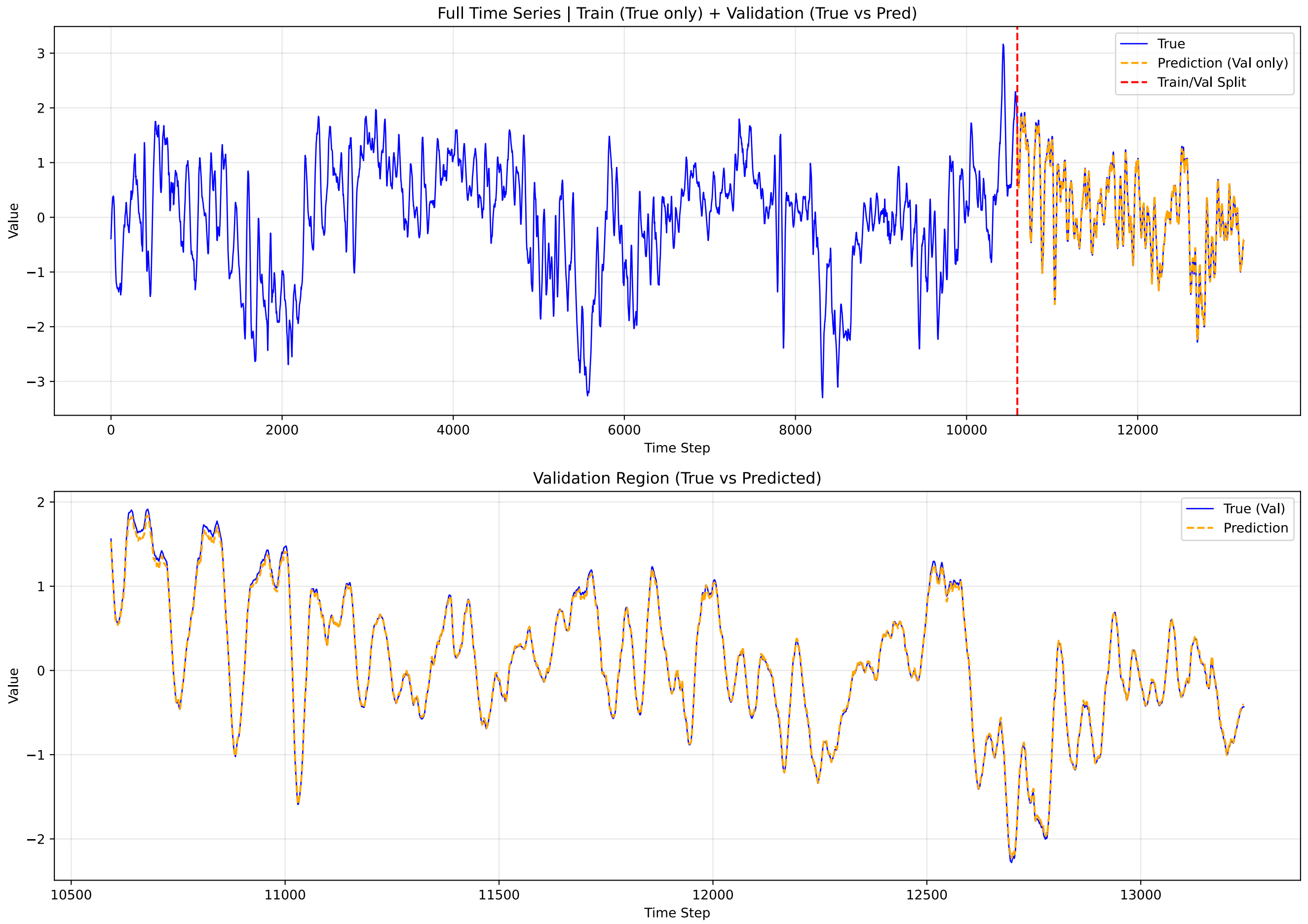


FIGURE 9. Time series forecasting results showing the complete dataset (upper panel) and detailed validation region (lower panel). The model achieves $R^2 = 0.996$ on the validation set, with predictions closely tracking the true time series values across the evaluation period. The vertical dashed line in the upper panel indicates the train-validation split.

This is not unexpected: (i) gates/scales probe *marginal sensitivity within the manifold of feasible models*, whereas ablation probes *counterfactual removal* (a harder intervention); (ii) some highly ranked Optuna parameters (e.g., regularization like dropout/L2) have no direct "off" analogue in the ablation; they can, however, be used to compensate a missing feature; (iii) search-range effects can inflate/deflate fANOVA contributions independently of removal sensitivity; and (iv) measurement noise from short retraining can blur Δ estimates. Despite these caveats, components with clear structural necessity (e.g., `direct_link`/`scale_direct`) tend to show stronger $\hat{\Delta}$ and non-trivial $\hat{I}$, aligning with the qualitative picture from the ablation section.

D. TIME SERIES FORECASTING PERFORMANCE

Figure 9 presents the model's performance on time series forecasting using the optimal configuration identified through the ablation study. The upper panel shows the complete time series with the train-validation split indicated by the vertical dashed line, while the lower panel provides a detailed view of the validation region.

The model achieves $R^2 = 0.996$ on the validation set, indicating that the predicted values capture 99.6% of the variance in the true time series. Visual inspection of the validation region reveals that the model accurately tracks both the amplitude and phase of the time series oscillations. The predictions follow the underlying temporal patterns, including periodic components and transient variations present in the data.

The close alignment between predicted and true values across the validation period demonstrates the model's ability to learn complex temporal dependencies. The forecasting accuracy remains consistent throughout the validation window, without apparent degradation at longer prediction horizons. This performance indicates that the combined feature set, particularly the direct link connections and attention mechanisms identified in the ablation study, provides effective temporal modeling capabilities for this time series.

E. MODEL COMPARISON

To evaluate the performance of our proposed RVFL-OptBayes architecture against established baselines, we conducted a comprehensive comparison study including deep learning

TABLE 3. Model comparison results across different architectures. Results are ranked by MSE in ascending order.

Model	MSE	MAE
RVFL-OptBayes (Ours)	0.000292	0.013322
LSTM_MHA	0.000352	0.014514
LSTM	0.000380	0.015160
LSTM_Attn	0.000385	0.015242
ExtraTrees	0.000437	0.016073
RandomForest	0.000455	0.016439
LightGBM	0.000539	0.017820
ElasticNet	0.000645	0.019629
XGBoost	0.000776	0.021522
DecisionTree	0.000824	0.022085
CatBoost	0.000892	0.023088
KNN_5	0.001031	0.023963
GradBoost	0.001038	0.024684
Transformer	0.003263	0.041456
AdaBoost	0.004693	0.054018

models, ensemble methods, and traditional machine learning approaches. The comparison encompasses transformer networks, LSTM-based architectures with various attention mechanisms, tree-based ensemble methods, gradient boosting variants, linear models, and nearest neighbor approaches.

Table 3 presents the comparative performance results ranked by MSE. Our RVFL-OptBayes model achieves the best overall performance with an MSE of 0.000292 and MAE of 0.013322, outperforming all baseline approaches. Among the deep learning models, the LSTM with multi-head attention (LSTM_MHA) shows the strongest performance with an MSE of 0.000352, followed by the standard LSTM (MSE = 0.000380) and LSTM with attention (MSE = 0.000385). The transformer architecture exhibits weaker performance with an MSE of 0.003263, indicating that attention mechanisms alone may not capture the temporal dependencies as well as LSTM-based approaches for this task.

The ensemble methods demonstrate competitive performance, with ExtraTrees achieving an MSE of 0.000437 and RandomForest reaching an MSE of 0.000455. These tree-based methods provide strong nonlinear modeling capacity but fall short of the proposed RVFL-OptBayes architecture. Among gradient boosting variants, LightGBM achieves the best performance with an MSE of 0.000539, followed by XGBoost (MSE = 0.000776), CatBoost (MSE = 0.000892), and GradBoost (MSE = 0.001038). The linear ElasticNet model shows an MSE of 0.000645, suggesting that the underlying data contains linear relationships that this model can capture.

The DecisionTree baseline achieves an MSE of 0.000824, while the K-nearest neighbors approach (KNN_5) demonstrates an MSE of 0.001031. AdaBoost shows degraded performance with an MSE of 0.004693, indicating limited capacity to improve upon weak learners for this regression task.

The superior performance of RVFL-OptBayes can be attributed to its hybrid architecture that combines the linear modeling capacity of direct connections with diverse nonlinear feature transformations, optimized through hyperparameter tuning. The results demonstrate that the proposed approach leverages both linear and nonlinear patterns in the data while maintaining computational efficiency compared to deep learning alternatives.

VI. CONCLUSION

This work proposed an enhanced RVFL network, integrated with Bayesian-based hyperparameter optimization, for short-term wind speed forecasting. By combining multi-scale feature generation, advanced weight initialization strategies, spectral normalization, and ensemble-based solvers, the framework successfully addressed key challenges in modeling nonlinear and multiscale wind dynamics. The adoption of a NAS strategy further enabled adaptive selection of feature representations, while Bayesian optimization ensured effective tuning of architectural configurations without reliance on backpropagation training.

Experimental results on real-world Brazilian wind farm data demonstrated that the proposed RVFL-OptBayes architecture significantly outperformed conventional RVFL networks, state-of-the-art deep learning approaches, and traditional ensemble methods. The ablation study confirmed the critical role of direct link connections, attention mechanisms, and wavelet-based features, while the optimization analysis highlighted the dominant influence of attention gating in model performance. With an achieved R^2 above 0.99 and consistent improvements across multiple evaluation metrics, the framework proved robust, accurate, and computationally efficient.

Beyond methodological advances, this study underscores the value of lightweight yet expressive architectures for renewable energy forecasting, particularly in contexts demanding real-time decision support. Future research may extend the proposed framework toward probabilistic forecasting, physics-informed feature integration, and multi-turbine spatiotemporal modeling, thereby enhancing its applicability for large-scale wind energy management and smart grid operations. Also, further exploring the architecture in a multi-step prediction environment is desired.

ACKNOWLEDGMENT

The authors thank CAPES for enabling open access through its transformative agreement with IEEE. Coordination for the Improvement of Higher Education Personnel—CAPES (ROR identifier: 00c0ma614).

REFERENCES

[1] M. H. D. M. Ribeiro, R. G. da Silva, S. R. Moreno, C. Canton, J. H. K. Larcher, S. F. Stefenon, V. C. Mariani, and L. D. S. Coelho, "Variational mode decomposition and bagging extreme learning machine with multi-objective optimization for wind power forecasting," *Int. J. Speech Technol.*, vol. 54, no. 4, pp. 3119–3134, Feb. 2024.

[2] A. K. Malik, R. Gao, M. A. Ganaie, M. Tanveer, and P. N. Suganthan, "Random vector functional link network: Recent developments, applications, and future directions," *Appl. Soft Comput.*, vol. 143, Aug. 2023, Art. no. 110377.

[3] M. A. A. Al-Qaness, A. A. Ewees, H. Fan, L. Abualigah, A. H. Elsheikh, and M. A. Elaziz, "Wind power prediction using random vector functional link network with capuchin search algorithm," *Ain Shams Eng. J.*, vol. 14, no. 9, Sep. 2023, Art. no. 102095.

[4] C. Zhang, Z. Li, Y. Ge, Q. Liu, L. Suo, S. Song, and T. Peng, "Enhancing short-term wind speed prediction based on an outlier-robust ensemble deep random vector functional link network with AOA-optimized VMD," *Energy*, vol. 296, Jun. 2024, Art. no. 131173.

[5] C. V. Zuege, S. F. Stefenon, C. K. Yamaguchi, V. C. Mariani, G. V. Gonzalez, and L. D. S. Coelho, "Wind speed forecasting approach using conformal prediction and feature importance selection," *Int. J. Electr. Power Energy Syst.*, vol. 168, Jul. 2025, Art. no. 110700.

[6] E. A. Tuncar, Ş. Sağlam, and B. Oral, "A review of short-term wind power generation forecasting methods in recent technological trends," *Energy Rep.*, vol. 12, pp. 197–209, Dec. 2024.

[7] L. S. Aquino, L. O. Seman, V. C. Mariani, L. D. S. Coelho, S. F. Stefenon, and G. V. González, "Spatiotemporal wind energy forecasting: A comprehensive survey and a deep equilibrium-based case study with StemGNN," *IEEE Access*, vol. 13, pp. 131461–131482, 2025.

[8] G. Ding, G. Yan, Z. Wang, B. Kang, Z. Xu, X. Zhang, H. Xiao, and W. He, "Adaptive SPP–CNN–LSTM–ATT wind farm cluster short-term power prediction model based on transitional weather classification," *Frontiers Energy Res.*, vol. 11, Dec. 2023, Art. no. 1253712.

[9] W. Yu, S. Li, H. Zhang, Y. Kang, H. Li, and H. Dong, "Ultra-short-term wind-power forecasting based on an optimized CNN-BILSTM-attention model," *iEnergy*, vol. 3, no. 4, pp. 268–282, Dec. 2024.

[10] H. Shu, W. Song, Z. Song, H. Guo, C. Li, and Y. Wang, "Multistep short-term wind speed prediction with rank pooling and fast Fourier transformation," *Wind Energy*, vol. 27, no. 7, pp. 667–694, 2024.

[11] Y. Wang, Z. Yang, J. Ma, and Q. Jin, "A wind speed forecasting framework for multiple turbines based on adaptive gate mechanism enhanced multi-graph attention networks," *Appl. Energy*, vol. 372, Oct. 2024, Art. no. 123777.

[12] D. Dong, S. Wang, Q. Guo, Y. Ding, X. Li, and Z. You, "Short-term marine wind speed forecasting based on dynamic graph embedding and spatiotemporal information," *J. Mar. Sci. Eng.*, vol. 12, no. 3, p. 502, Mar. 2024.

[13] Q. He, "Combined wind speed prediction model based on GAT-LSTM," *Int. Core J. Eng.*, vol. 10, no. 9, pp. 61–67, 2024.

[14] S. Dewitte, J. P. Cornelis, R. Müller, and A. Munteanu, "Artificial intelligence revolutionises weather forecast, climate monitoring and decadal prediction," *Remote Sens.*, vol. 13, no. 16, p. 3209, Aug. 2021.

[15] Y. Wang, H. Xu, R. Zou, F. Zhang, and Q. Hu, "Dynamic non-constraint ensemble model for probabilistic wind power and wind speed forecasting," *Renew. Sustain. Energy Rev.*, vol. 204, Oct. 2024, Art. no. 114781.

[16] Y. Zhang, X. Kong, J. Wang, H. Wang, and X. Cheng, "Wind power forecasting system with data enhancement and algorithm improvement," *Renew. Sustain. Energy Rev.*, vol. 196, May 2024, Art. no. 114349.

[17] T. Wu and Q. Ling, "STELLM: Spatio-temporal enhanced pre-trained large language model for wind speed forecasting," *Appl. Energy*, vol. 375, Dec. 2024, Art. no. 124034.

[18] T. Wu and Q. Ling, "Mixformer: Mixture transformer with hierarchical context for spatio-temporal wind speed forecasting," *Energy Convers. Manage.*, vol. 299, Jan. 2024, Art. no. 117896.

[19] T. Bashir, H. Wang, M. Tahir, and Y. Zhang, "Wind and solar power forecasting based on hybrid CNN-ABiLSTM, CNN-transformer-MLP models," *Renew. Energy*, vol. 239, Feb. 2025, Art. no. 122055.

[20] J. Wang, X. Niu, L. Zhang, Z. Liu, and X. Huang, "A wind speed forecasting system for the construction of a smart grid with two-stage data processing based on improved ELM and deep learning strategies," *Expert Syst. Appl.*, vol. 241, May 2024, Art. no. 122487.

[21] S. R. Mohapatro, M. Mulchandani, N. Mohanty, M. R. Sethi, S. Sahoo, and A. Samad, "Efficient wind turbine fault diagnosis using machine learning technique and hyper-parameter tuning," in *Proc. Int. Conf. Adv. Data-Driven Comput. Intell. Syst.* Cham, Switzerland: Springer, 2024, pp. 163–174.

[22] C. Tian, T. Niu, and T. Li, "Developing an interpretable wind power forecasting system using a transformer network and transfer learning," *Energy Convers. Manage.*, vol. 323, Jan. 2025, Art. no. 119155.

[23] M. A. Hossain, R. K. Chakrabortty, S. Elsawah, and M. J. Ryan, "Very short-term forecasting of wind power generation using hybrid deep learning model," *J. Cleaner Prod.*, vol. 296, May 2021, Art. no. 126564.

[24] Y. Ren, Z. Li, L. Xu, and J. Yu, "The data-based adaptive graph learning network for analysis and prediction of offshore wind speed," *Energy*, vol. 267, Mar. 2023, Art. no. 126590.

[25] Y. Liao, Z. Gao, and X. Li, "Wind farm meteorological prediction model based on frequency domain feature extraction fusion mechanism," *IEEE Access*, vol. 13, pp. 57426–57441, 2025.

[26] S. Stipa, A. Ajay, D. Allaerts, and J. Brinkerhoff, "The multi-scale coupled model: A new framework capturing wind farm–atmosphere interaction and global blockage effects," *Wind Energy Sci.*, vol. 9, no. 5, pp. 1123–1152, May 2024.

[27] H. R. Sezavar, S. Hasanzadeh, and N. Fahimi, "A novel hybrid mathematical deep learning technique for early warning of flashover in composite insulators," *Sci. Rep.*, vol. 15, no. 1, p. 33448, Sep. 2025.

[28] B. Zhao, H. Liu, T. Bi, and S. Xu, "Synchrophasor measurement method based on cascaded infinite impulse response and dual finite impulse response filters," *J. Modern Power Syst. Clean Energy*, vol. 12, no. 5, pp. 1345–1356, 2024.

[29] T. Kruse, T. Griebel, and K. Graichen, "Adaptive Kalman filtering: Measurement and process noise covariance estimation using Kalman smoothing," *IEEE Access*, vol. 13, pp. 11863–11875, 2025.

[30] K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. K. Somani, "Neural architecture search benchmarks: Insights and survey," *IEEE Access*, vol. 11, pp. 25217–25236, 2023.

[31] R. M. Adnan, R. R. Mostafa, M. Wang, K. S. Parmar, O. Kisi, and M. Zounemat-Kermani, "Improved random vector functional link network with an enhanced remora optimization algorithm for predicting monthly streamflow," *J. Hydrol.*, vol. 650, Apr. 2025, Art. no. 132496.

[32] L. O. Seman, L. S. Aquino, S. F. Stefenon, K.-C. Yow, V. C. Mariani, and L. D. S. Coelho, "Simultaneously anomaly detection and forecasting for predictive maintenance using a zero-cost differentiable architecture search-based network," *Comput. Ind. Eng.*, vol. 208, Oct. 2025, Art. no. 111412.

[33] Y. Feng, Y. Sun, G. G. Yen, and K. C. Tan, "REP: An interpretable robustness enhanced plugin for differentiable neural architecture search," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 5, pp. 2888–2902, May 2025.

[34] J. Á. Martín-Baos, R. García-Ródenas, L. Rodriguez-Benitez, and M. Bierlaire, "Scalable kernel logistic regression with nyström approximation: Theoretical analysis and application to discrete choice modelling," *Neurocomputing*, vol. 617, Feb. 2025, Art. no. 128975.

[35] M. Mallik, J. Schampheleer, L. Clavier, and M. Deruyck, "Fast field strength prediction using modern machine learning in European cities from few RF-EMF measurements: A neural tangent kernel perspective," *IEEE Access*, vol. 13, pp. 131003–131014, 2025.

[36] Q. Snyder, Q. Jiang, and E. Tripp, "Integrating self-attention mechanisms in deep learning: A novel dual-head ensemble transformer with its application to bearing fault diagnosis," *Signal Process.*, vol. 227, Feb. 2025, Art. no. 109683.

[37] R. N. Muniz, S. F. Stefenon, W. G. Buratto, A. Nied, R. Cardoso, C. K. Yamaguchi, and K.-C. Yow, "Time series forecasting based on multi-criteria optimization for model and filter selection applied to hydroelectric power plants," *Energy*, vol. 337, Nov. 2025, Art. no. 138688.

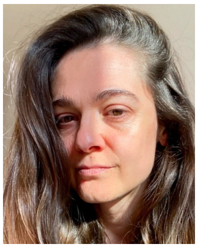
[38] K. K. Wong, "Ensemble machine learning and tree-structured Parzen estimator to predict early-stage pancreatic cancer," *Biomed. Signal Process. Control*, vol. 108, Oct. 2025, Art. no. 107867.

[39] J.-X. Liu and J.-S. Leu, "LARSI-TPE-XGB: Short-term load forecasting by load-adaptive relative strength index and fusion of tree-structured Parzen estimator and XGBoost," *IEEE Trans. Power Del.*, vol. 40, no. 3, pp. 1318–1330, Jun. 2025.

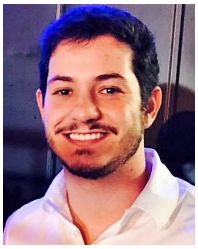
[40] A. Bhardwaj, V. Mangat, and R. Vig, "Hyperband tuned deep neural network with well posed stacked sparse AutoEncoder for detection of DDoS attacks in cloud," *IEEE Access*, vol. 8, pp. 181916–181929, 2020.

[41] W. G. Buratto, R. N. Muniz, R. Cardoso, A. Nied, C. T. da Costa, and G. V. Gonzalez, "Hybrid group method of data handling for time-series forecasting of thermal generation dispatch in electrical power systems," *Electr. Eng.*, vol. 107, no. 10, pp. 13929–13945, Oct. 2025.

LAIO ORIEL SEMAN received the Ph.D. degree in electrical engineering from the Federal University of Santa Catarina, in 2017. His research interests include strategies for static and dynamic optimization, along with applications in traffic control, cyber-physical systems, and oil and gas production systems.



ANNE CAROLINA RODRIGUES KLAAR received the B. in Fashion (Industrial Design) degree from the Regional University Foundation of Blumenau, and the M.E. degree in education from the University of Planalto Catarinense (UNIPLAC), Brazil. She was an Instructor with a Specialist III Degree with the National Industrial Learning Service, from 2013 to 2021, teaching oral and written communication and project development methodology. She is currently a member with the Center for Studies and Research on Philosophy Teaching and Philosophical Education, UNIPLAC.



MATHEUS HENRIQUE DAL MOLIN RIBEIRO received the degree in mathematics from the Federal University of Technology—Paraná, in 2013, the M.Sc. degree in biostatistics from the State University of Maringá, in 2015, and the Ph.D. degree from the Graduate Program in Environmental Engineering, Pontifical Catholic University of Paraná, Curitiba, Brazil. He is currently an Adjunct Professor with the Federal University of Technology—Paraná. His research interests include analysis of repeated measures, longitudinal data, statistical models for longitudinal data, time series forecasting, machine learning, and mono and multi-objective optimization.

STEFANO FRIZZO STEFENON received the Ph.D. degree in electrical engineering from the State University of Santa Catarina (UDESC), Brazil, in 2021, and the Ph.D. degree in computer science and artificial intelligence from the Università degli Studi di Udine (UNIUD), Italy, in 2025.

He was a Post-Doctoral Fellow Researcher with the Faculty of Engineering and Applied Science, University of Regina (UofR), Saskatchewan, Canada. He is currently a Professor (Adjunct) with Lisbon School of Engineering (ISEL), Polytechnic University of Lisbon (IPL), Portugal. His research interests include electrical power systems, deep learning, computer vision, and time series forecasting.

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614