



# Optimizing long-short-term memory models via metaheuristics for decomposition aided wind energy generation forecasting

Marijana Pavlov-Kagadejev<sup>1</sup> · Luka Jovanovic<sup>2</sup> · Nebojsa Bacanin<sup>3</sup> · Muhammet Deveci<sup>4,5,6</sup> · Miodrag Zivkovic<sup>3</sup> · Milan Tuba<sup>3</sup> · Ivana Strumberger<sup>3</sup> · Witold Pedrycz<sup>7,8,9</sup>

Accepted: 20 December 2023 / Published online: 12 February 2024  
© The Author(s) 2024

## Abstract

Power supply from renewable energy is an important part of modern power grids. Robust methods for predicting production are required to balance production and demand to avoid losses. This study proposed an approach that incorporates signal decomposition techniques with Long Short-Term Memory (LSTM) neural networks tuned via a modified metaheuristic algorithm used for wind power generation forecasting. LSTM networks perform notably well when addressing time-series prediction, and further hyperparameter tuning by a modified version of the reptile search algorithm (RSA) can help improve performance. The modified RSA was first evaluated against standard CEC2019 benchmark instances before being applied to the practical challenge. The proposed tuned LSTM model has been tested against two wind production datasets with hourly resolutions. The predictions were executed without and with decomposition for one, two, and three steps ahead. Simulation outcomes have been compared to LSTM networks tuned by other cutting-edge metaheuristics. It was observed that the introduced methodology notably exceed other contenders, as was later confirmed by the statistical analysis. Finally, this study also provides interpretations of the best-performing models on both observed datasets, accompanied by the analysis of the importance and impact each feature has on the predictions.

**Keywords** Long short-term memory networks · Metaheuristics optimization · Reptile search algorithm · Shapley additive explanations · Wind power generation

## 1 Introduction

Accurate power generation using renewable sources is crucial for several reasons. Renewable energy generation using wind power as well as photovoltaic sources is inherently variable (Li et al. 2023), which means that their output can fluctuate based on weather

---

Marijana Pavlov-Kagadejev, Luka Jovanovic, Nebojsa Bacanin, Muhammet Deveci, Miodrag Zivkovic, Milan Tuba, Ivana Strumberger, and Witold Pedrycz have contributed equally to this work.

Extended author information available on the last page of the article

conditions and other factors. Accurate power generation data enables grid operators and energy companies to better anticipate these fluctuations and manage the overall power supply to ensure a stable and reliable grid (Coppitters and Contino 2023). Accurate power generation data is essential for billing purposes, as it ensures that energy providers are properly compensated for the power they generate. Finally, accurate power generation data is important for policy-making and research (Awerbuch and Berger 2003), as it helps to inform decisions about energy infrastructure investment, environmental impact, and renewable energy technology development. In short, accurate power generation data is critical for ensuring the efficiency, reliability, and sustainability of our energy systems.

Historical power generation data could prove a useful indicator for future forecasting (Shi et al. 2012). However, several interconnected parameters affect power generation, all of which are prone to volatile changes. This makes prediction very difficult due to signal complexity. Decomposition techniques such as Variational Mode Decomposition (VMD) (Rehman and Aftab 2019) and Empirical Mode Decomposition (EMD) (Rehman and Mandic 2010) have the potential to deal with signal complexity by breaking down a complex signal into simpler, more easily analyzed components. These techniques are particularly useful for signals that exhibit non-stationary and nonlinear behavior. Both decompose a signal creating a set of intrinsic mode functions (IMFs) or variational modes, respectively, that capture the underlying frequency components of the signal. Each IMF or variational mode represents a distinct frequency component of the signal, with the highest frequency modes capturing the most rapid and transient changes, and the lowest frequency modes capturing the slower, more persistent changes. By analyzing the IMFs or variational modes separately, researchers can gain insights into the underlying patterns and dynamics of the signal that may be obscured by its complexity. Overall, decomposition techniques such as VMD and EMD offer a powerful approach to dealing with signal complexity, providing researchers with a deeper understanding of the underlying dynamics and behavior of complex signals.

Emerging artificial intelligence (AI) techniques have the potential for accurately forecasting production from wind sources in several ways. Models can be trained on large datasets of historical weather and wind data to better predict wind speeds and direction, which are critical factors in wind energy production. This can be leveraged to refine the accuracy of production forecasting and help energy providers to better anticipate fluctuations in wind power output. Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) neural networks are a type of recurrent neural network that are designed to model sequences and time series data. LSTMs are suitable for time series data because they can capture and remember long-term dependencies and patterns in the data over time, while also avoiding the vanishing gradient problem that can occur in traditional recurrent neural networks. LSTMs use a system of “gates” to moderate the transmission of data and control the memory of the network, allowing them to selectively forget or remember information from previous time steps. This makes LSTMs highly effective for modeling time series data with complex temporal dynamics. These types of networks offer a powerful tool for tackling time series data, allowing for accurate predictions and insights into the subtle patterns and connections in the data. However, like many algorithms, they present a set of hyperparameters that require adequate adjustment to attain desirable outcomes.

Hyperparameters are an essential aspect of forecasting AI models, as they can significantly influence model performance Probst et al. (2019). These parameters are not learned by the model during training but are set by the user before training. Choosing the right hyperparameters for a model is critical, as selecting the wrong values can lead to poor performance, slow convergence, or overfitting. Hyperparameter tuning involves selecting

optimal values for these hyperparameters so as to attain the best possible outcomes from the model. This is typically done through a combination of trial and error and automated techniques. The process involves training the model with different hyperparameter values and evaluating the performance on a validation set. The hyperparameter values that produce an optimized model are then selected.

Metaheuristic algorithms present a powerful class of optimization algorithms that may be applied to hyperparameter tuning in forecasting AI models Bacanin et al. (2023b). Iterative optimization is designed to traverse and analyze the search space of possible solutions, without making assumptions about the objective function or the structure of the problem. They are particularly well-suited to hyperparameter tuning, as they can handle high-dimensional and non-convex exploration spaces with many local optima. Metaheuristic algorithms offer a promising approach to hyperparameter tuning in forecasting AI models and can help to overcome some of the limitations of traditional optimization methods. By leveraging the power of iterative search and exploration, metaheuristic algorithms can help to identify optimal hyperparameter values and improve the accuracy and performance of forecasting AI models. Swarm intelligence metaheuristic algorithms are a group of optimization algorithms inspired by the collective behavior of social insects and animals. The concept of decentralized self-organization is at the core of these algorithms, in which a group of individuals mutually interact to reach a shared goal. By following simple sets of rules, complex behaviors emerge on a global scale. These algorithms have the ability to address non-deterministic polynomial-time hard (NP-hard) optimization problems spending time and resources reasonably, something often constituting a problem with traditional methods. This being said, in accordance with the no free lunch theorem (NFL) (Wolpert and Macready 1997), no single methodology is best for all problems, instead an individual approach is preferred. Therefore extensive investigation is needed to further improve techniques and methods.

A notably well-performing metaheuristic algorithm used in this research is the reptile search algorithm (RSA) (Abualigah et al. 2022). It is a meta-learning approach that adapts a model to new tasks quickly. It works by training the model on a set of tasks for a fixed number of iterations and then fine-tuning it on new tasks using only a few gradient updates. Reptile search algorithm aims to learn a good initialization of model parameters that can be quickly adapted to new tasks with minimal updates.

A motivation for the conducted research was to further explore and expand the understanding of the novel RSA and its potential for hyperparameter tuning. Additionally, a key motivator was to determine if this already admirably performing metaheuristic can be further improved through hybridization with other well-known powerful optimizers. Finally, this research hopes to introduce a robust AI-based method improved by the introduced metaheuristic in order to better address the pressing real-world issue of wind power generation forecasting.

With this in mind, this work proposes a novel method for forecasting power generated by wind farms based on a time series of meteorological and historical factors. To account for the complexities caused by the volatility associated with these predictors two signal decomposition techniques are utilized, VMD and EMD. The processed data is formulated as a time series and six input lags are utilized in order to train LSTM neural networks to create forecasts three steps ahead. With the goal of optimizing the performance of the models, metaheuristic algorithms are applied for hyperparameter selection. Several metaheuristics are evaluated and a new modified version of the RSA is introduced. New and modified algorithm performance is usually initially evaluated using a set of standardized benchmarking functions. Accordingly, the modified metaheuristic was evaluated using a wide range

of standard bound-constrained CEC2019 benchmarking functions before being applied to a real-world challenge. Following these evaluations, each metaheuristic optimized decomposition-aided LSMT approach is evaluated on two real-world data sets covering two wind power plants in different parts of the world to determine their performance. The best-performing models have been interpreted using SHapley Additive exPlanations (SHAP) (Lundberg and Lee 2017) methods to determine the factors that have the highest influence on model predictions.

The central contributions of the presented research work are summarised as:

- A proposal for a modified variation of the recently introduced RSA that better the commendable performance of the original
- An introduction of a decomposition-aided metaheuristic optimized methodology for wind power generation prognosis
- An interpretation of the best-performing models using SHapley SHAP analysis to better understand the factors that contribute the most to wind power generation

The remainder of the word follows the structure hereby presented: preceding research that lay out a foundation for this research is presented in the related works Sect. 2. The utilized methods and newly introduced metaheuristics are presented in Sect. 3. The capabilities of the introduced algorithm on bound-constrained functions are shown and discussed in Sect. 4. The experimental setup followed by the attained results on the two real-world datasets along with the achieved results' discussion are presented in Sects. 5 and 6, respectively. Finally, a conclusion of the work and proposals for future research are shown in Sect. 7.

## 2 Overview of research background and literature review

In research, interest in LSTM neural networks has been renewed for forecasting wind power generation. Several studies have demonstrated the effectiveness of LSTM-based models in accurately predicting wind power output over short-term and long-term horizons. One such study (Shahid et al. 2021) proposed an LSTM-based model for short-term wind power forecasting that incorporated both meteorological and power data. The authors demonstrated that the given model outperformed traditional time series models and other ML (ML) algorithms, achieving high accuracy and robustness across different locations and weather conditions.

Another study (Liu et al. 2019) focused on long-term wind power forecasting using LSTM networks. The authors introduced a hybrid model that combined LSTM and wavelet transform and principal component analysis to capture both the temporal and spatial variations in wind power data. The results showed that the proposed model did better than traditional statistical models and other ML algorithms, with improved accuracy and robustness over longer forecasting horizons.

In addition to LSTM neural networks, other techniques have been explored to help augment wind power forecasting precision, including the use of decomposition techniques such as VMD and EMD. Researchers (Zhang et al. 2016) proposed a VMD-based approach to upgrade the accuracy of wind power forecasting by decomposing the time series into different frequency components and applying separate models to each component. The authors demonstrated that the VMD-based approach outperformed traditional time series

models and other ML algorithms, achieving high accuracy and robustness across different locations and weather conditions. While other approaches for time-series forecasting exist their potential has been sufficiently explored in literature.

Hyperparameter tuning is a paramount aspect of ML and has been tested in the context of wind power forecasting using metaheuristic algorithms. Previous works (Shao et al. 2021) proposed a firework algorithm-based approach to optimize hyperparameters of LSTM neural networks for wind power forecasting. The authors demonstrated that the proposed approach achieved higher forecasting accuracy compared to other optimization techniques, indicating the importance of hyperparameter tuning for accurate wind power forecasting. Moreover, metaheuristics have been applied to optimization across several fields and have shown admirable results including crude oil price forecasting (Jovanovic et al. 2022a), and environmental sciences (Jovanovic et al. 2023a).

Another approach to augment the interoperability of wind electricity production forecasting models is the use of SHAP (Lundberg and Lee 2017) values. These values provide a way to interpret the reasoning behind ML model decisions, by determining contributions made by available features towards the final outcome. The use of SHAP values could be used to understand the factors that influence wind power forecasting accuracy using LSTM-based models. SHAP values provided a more comprehensive and intuitive understanding of model performance, compared to traditional evaluation metrics.

## 2.1 Motivation

Renewable energy plays a crucial role in addressing our planet's pressing environmental challenges (Akella et al. 2009; Dinçer et al., 2023; Yüksel et al., 2024). By harnessing sources like solar and wind we can significantly reduce greenhouse gas emissions and combat climate change (Razmjoo et al. 2021). Embracing renewable energy not only safeguards the environment but also promotes energy security, job creation, and a healthier future for generations to come.

However, methods for generating energy in a renewable way are still developing. To facilitate large-scale adoption, many challenges need to be overcome (Züttel et al. 2022). One major challenge comes in the form of increased reliability. Being able to forecast the available power systems can improve reliability in the long run, increasing the viability of renewable systems.

The potential of metaheuristic algorithms for hyperparameter optimization is a well-established approach (Tayebi and El Kafhali 2022), however, it has not yet been explored when applied to wind power generation using LSTM networks. With novel and more powerful techniques constantly being developed (Mattos Neto et al. 2021; Belotti et al. 2020), evaluation and innovation are required in order to improve the body of work available to researchers tackling optimizations. The potential of the recently introduced RSA (Abualigah et al. 2022) has not yet been explored and implemented in energy forecasting. Furthermore, as a relatively recent approach, this algorithm has great potential for improvement through hybridization.

Decomposition techniques offer yet another technique that can improve model training, with new methods being developed experimentation is required to determine their potential for helping address this increasingly pressing problem. Model interpretation techniques (Dwivedi et al. 2023) are increasingly important to build more reliable models and robust systems. This work aims to address the observed research gap and improve the body of available techniques for renewable energy forecasting.

## 2.2 Variational mode decomposition (VMD)

Variational mode decomposition (VMD) (Rehman and Aftab 2019) is a methodology for decomposing signals into base components in a non-reducible way. The base of it is Wiener filtering and the Hilbert transform (Zhang et al. 2021). This adaptive signal decomposition method can decompose a given signal  $f(t)$  into several components signals  $u_k(t)$  within bandwidth constraints around a center frequency  $\omega_k$  according to Eq. (1) (Wang and Li 2023).

$$\min_{\{u_k\}, \{\omega_k\}} \left\{ \sum_{k=1}^K \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\}, \text{ s.t. } \sum_{k=1}^K u_k = f(t) \quad (1)$$

where  $K$  presents the count of decomposed modes,  $\{u_k\} = \{u_1, u_2, \dots, u_k\}$  are modal components with a center frequencies  $\{\omega_k\} = \{\omega_1, \omega_2, \dots, \omega_k\}$ ,  $\partial_t$  is partial derivative.  $\delta(t)$  represents the Dirac distribution,  $f(t)$  depicts the original input signal,  $u_k(t)$  represents  $k$ -th subsequence of  $f(t)$  and  $*$  marks a convolution operator.

Lagrange multiplication operator  $\lambda$  and the quadratic penalty factor  $\alpha$  are incorporated to upgrade the optimal solution of constrained variation as per the following:

$$\begin{aligned} L(\{u_k\}, \{\omega_k\}, \lambda) = & \alpha \sum_{k=1}^K \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ & + \left\| f(t) - \sum_{k=1}^K u_k(t) \right\|_2^2 + \left\langle \lambda(t), f(t) - \sum_{k=1}^K u_k(t) \right\rangle \end{aligned}$$

The Lagrange function is transformed from the time domain to the frequency domain. The alternate direction method of multipliers (ADMM) is utilized to minimize the optimization problem. The modes  $u_k$  and their center frequency  $\omega_k$  are calculated using the following equations respectively: Eqs. (2) and (3).

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i \neq k} \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2} \quad (2)$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega |\hat{u}_k(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k(\omega)|^2 d\omega} \quad (3)$$

in which  $n$  is a number of iteration,  $\lambda$  is Lagrange operator given by Eq. (4).

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau \left[ \hat{f}(\omega) - \sum_{k=1}^K \hat{u}_k^{n+1}(\omega) \right] \quad (4)$$

The iterative process will be executed until the condition given by Eq. (5) is met.

$$\frac{\sum_{k=1}^K \left\| \hat{u}_k^{n+1}(\omega) - \hat{u}_k^n(\omega) \right\|_2^2}{\left\| \hat{u}_k^n(\omega) \right\|_2^2} < \epsilon \quad (5)$$

### 2.3 Empirical mode decomposition (EMD)

Empirical mode decomposition (EMD) (Rehman and Mandic 2010) is a signal decomposition approach for reducing the amount of noise in non-stationary time data series (Devi et al. 2020). Fourier and Wavelet analysis are used. This technique gives good results for analyzing wind speed data series. By using the EMD algorithm, complex time series data can be decomposed into a limited number of Intrinsic Mode Functions (IMFs) (Wang et al. 2022b).

EMD process has the following procedure:

- (1) Determine the local maximum and minimum value for any processed signal  $x(t)$ . Record  $h_1(t)$  as difference between  $x(t)$  and mean value of upper and lower envelope  $m_1(t)$ , according to Eq. (6)

$$h_1(t) = x(t) - m_1(t) \quad (6)$$

- (2)  $h_1(t)$  filtered out of the original signal tends to contain the signal's highest frequency component. Difference signal,  $r_1(t)$ , is gained by separating  $h_1(t)$  from  $x(t)$ . That way, the high-frequency component is removed. Filtering steps are repeated, with  $r_1(t)$  as a new signal, with the goal of the residual signal in the  $n$ -th stage becoming a monotonic function. This is shown by Eq. (7).

$$\begin{cases} r_1(t) = x(t) - h_1(t) \\ r_2(t) = r_1(t) - h_2(t) \\ \vdots \\ r_n(t) = r_{n-1}(t) - h_n(t) \end{cases} \quad (7)$$

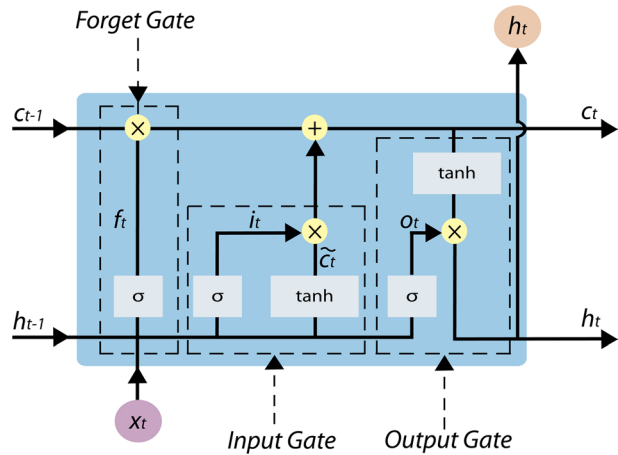
in which  $x(t)$  can be represented as a sum of  $n$  IMFs and a single residual according to Eq. (8).

$$x(t) = \sum_{j=1}^n h_j(t) + r_n(t) \quad (8)$$

where  $r_n(t)$  is the residual that denotes the signal average trend,  $h_j(t)$  is the  $j$ -th IMF,  $j = 1, 2, \dots, n$ , represents the various signal components in the direction from high to low frequencies.

Standard deviation (SD) is given by Eq. (9). It is mainly set from 0.2 to 0.3.

$$SD = \sum_{j=1}^n \frac{|h_j(t) - h_{j-1}(t)|^2}{|h_{j-1}(t)|^2} \quad (9)$$

**Fig. 1** The architecture of LSTM cell

## 2.4 Long short-term memory (LSTM)

Recurrent neural networks (RNN) (Amalou et al. 2022), represent network architecture specialized for processing sequential data. However, RNNs have a problem of gradient disappearance or gradient explosion. To overcome this, researchers proposed the Long Short-Term Memory Neural Network (LSTM) (Hochreiter and Schmidhuber 1997). This type of RNN can learn long-term dependent information (Liu et al. 2020). It can remember the relationship of the current information with the long-term information in the time sequence. The hidden level of traditional RNN has been replaced with memory cell (Wang et al. 2022a). It is comprised of a forget gate, input, and output gate, as shown in Fig. 1 (Liu et al. 2020; Wang et al. 2022a).

The basic structure in an LSTM is a memory block. This unit contains a cell used to store data as a system consisting of three control gates: forget, input, and output. All gate units have the same structure and consist of the same sequence: sigmoid activation function and multiplication in a range  $[0, 1]$ , and determines the amount of information passing through. The output of activation function  $\tanh$  is in the range  $[-1, 1]$  (Wang et al. 2022b).

LSTM input is made up of previous sequences  $h_{t-1}$  as well as the ongoing input  $x_t$ . The forget gate determines values in the cell state  $C_{t-1}$  to be discarded, as it is defined by Eq. (10) (Fu et al. 2019).

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (10)$$

where  $f_t$  is an output vector with values in the range  $[0, 1]$ ,  $\sigma$  is the sigmoid function,  $W_f$  and  $U_f$  are the weight matrices and  $b_f$  is the bias vector. The input gate updates the information using the result of the sigmoid layer  $i_t$  according to Eq. (11).

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (11)$$

where  $W_i$ ,  $U_i$  are the weight matrices and  $b_i$  is the bias vector.

The new potential values of cell state vector  $\tilde{C}_t$  are generated by  $\tanh$  function, determined by Eq. (12).

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (12)$$



$C_t$  is obtained by multiplying the old cell state  $C_{t-1}$  with  $f_t$  (to forget the unwanted information) and adding new potential information  $i_t \otimes \tilde{C}_t$ , as it is given by Eq. (13)

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \quad (13)$$

with the  $\otimes$  indicating an element-wise multiplication (Wang et al. 2022a; Fan et al. 2020). The output gates value  $o_t$  is obtained according to Eq. (14)

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (14)$$

where  $W_o, U_o$  are the weight matrices and  $b_o$  is the bias vector. The value of output  $h_t$  is calculated using Eq. (15).

$$h_t = o_t \otimes \tanh(C_t) \quad (15)$$

With the  $\tanh$  function, cell state  $C_t$  is scaled in the range  $[-1, 1]$ , and multiplying it with the output of output gate  $o_t$ , a new output value  $h_t$  has been calculated.

## 2.5 Metaheuristics optimization

Stochastic algorithms, including metaheuristics, are often necessary for computer science to tackle NP-hard challenges because deterministic algorithms are not practical. Metaheuristics algorithms may be classified into categories emulating natural processes to lead the search method. For instance, some methods are inspired by evolution, natural selection, or birds and insects' collective behavior (Stegherr et al. 2020; Emmerich et al. 2018; Fausto et al. 2020). The most prominent groups of metaheuristics approaches include nature-inspired algorithms, consisting of genetic algorithms and swarm intelligence, as well as algorithms based on some physical phenomena (e.g., storm, gravitational and electromagnetic fields). Other approaches include those that mimic facets of human behavior such as teaching and learning, brainstorming, or social media activity and those that were derived from fundamental mathematical laws that pilot the search, e.g. through the use of trigonometric function oscillations.

Swarm intelligence methods have been based on groups' cooperative actions composed of relatively simple individuals, such as swarms of insects or flocks of birds, which can exhibit astonishingly coordinated and sophisticated behavior patterns while performing fundamental survival tasks such as hunting, foraging, mating, or migrating (Beni 2020; Abraham et al. 2006). These methods have demonstrated significant potential when tackling real-world NP-hard problems. However, they have been sometimes known to fail. Several popular swarm intelligence algorithms include particle swarm optimization (PSO) (Kennedy and Eberhart 1995), ant colony optimization (ACO) (Dorigo et al. 2006), firefly algorithm (FA) (Yang 2009), and bat algorithm (BA) (Yang 2010; Yang and Gandomi 2012). In the past few years, a highly effective group of metaheuristics has been developed based on mathematical functions and their behavior patterns to guide the search process, where the most notable samples are the sine-cosine algorithm (SCA) (Mirjalili 2016) and the arithmetic optimization algorithm (AOA) (Abualigah et al. 2021).

The NFL is the main cause of such a variety of optimization methodologies existing. The NFL states that there is no single algorithm that can be universally superior for every optimization task. Therefore, while one algorithm may perform well on a particular problem, it may fall short entirely on another, highlighting the necessity for diverse

metaheuristic methods and the need to select the most appropriate method for each individual optimization task.

Population-based algorithms have lately become a usual choice for addressing different real-world problems. These algorithms are useful for many fields, such as prediction of COVID-19 cases (Zivkovic et al. 2021a, b), organizing on demand computational services (Bacanin et al. 2019; Bezdan et al. 2020a, b; Zivkovic et al. 2021c), optimizing wireless sensors and IoT (Zivkovic et al. 2020, 2021d), feature selection (Bezdan et al. 2021; Bacanin et al. 2023a), processing and classifying medical images (Bezdan et al. 2020c; Zivkovic et al. 2022), addressing global optimization problems (Strumberger et al. 2019; Preuss et al. 2011), identifying credit card fraud (Jovanovic et al. 2022b; Petrovic et al. 2022), monitoring and forecasting air pollution (Bacanin et al. 2022a; Jovanovic et al. 2023a), detecting network and computer system intrusions (Bacanin et al. 2022b; Stankovic et al. 2022), predicting power generation and energy load (Bacanin et al. 2023b; Stoean et al. 2023), and optimizing different ML models (Salb et al. 2022; Milosevic et al. 2021; Gajic et al. 2021; Bacanin et al. 2022c, d; Jovanovic et al. 2022a, 2023b; Bukumira et al. 2022).

## 2.6 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) (Lundberg and Lee 2017) is a method for interpreting the output of ML models, particularly those that are a black-box or difficult to interpret. The methodology of SHAP is the computation of the Shapley value concept rooted in cooperative game theory to explain the role of the available feature and its impact on decisions. In SHAP, the Shapley value for a feature represents the average influence of that feature on the model's output across all possible subsets of features.

To calculate the Shapley value for a feature, we first define a reference value for that feature. This reference value could be the average value of that feature in the dataset, or it could be a user-defined value. We then create a set of all possible feature subsets that include the feature we are interested in. For example, if we are interested in the Shapley value of feature A, we might create subsets that include just A, or A and B, or A and C, and so on.

For each subset, we calculate the contribution of the feature to the model's output relative to the reference value. This contribution might be positive or negative, causing the feature's value in the subset to increase or decrease the model's output. We then calculate the average contribution of the feature across all possible subsets, which gives us the Shapley value.

Mathematically, the Shapley value for a feature  $j$  is determined by:

$$\phi_j = \sum_{S \subseteq M \setminus j} \frac{|S|!(|M| - |S| - 1)!}{|M|!} (f(S \cup j) - f(S)) \quad (16)$$

Here,  $M$  is the set of all features, and  $f(S)$  is the model's output for a given subset of features  $S$ . The term inside the summation calculates the marginal contribution of feature  $j$  to the subset  $S$ , and the summation calculates the average contribution across all possible subsets.

In practice, we can estimate the Shapley values for a model using a technique called Kernel SHAP. This involves generating a set of "background" instances that are representative of the dataset and then using these instances to estimate the expected model output for

each subset of features. The Shapley values can then be calculated using these expected outputs.

Once we have the Shapley values for all features, we can use them to create a SHAP summary plot, which shows the contribution of each feature to the model's output. This plot can help us identify which features are most important for the model's predictions, and how each feature contributes to those predictions.

### 3 Methods

#### 3.1 The original reptile search algorithm (RSA)

The RSA (Abualigah et al. 2022) is an optimization algorithm that mathematically simulates the predatory techniques of Crocodiles. This algorithm simulates two main techniques: encircling and hunting. The details of both are outlined in the original work (Abualigah et al. 2022).

#### 3.2 Initialization phase

Optimization initialized with a stochastically generated candidates ( $X$ ) shown in Eq. (17)

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots \\ \vdots & & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad (17)$$

$$x_{i,j} = rand \times (UB - LB) + LB, j = 1, 2, \dots, n \quad (18)$$

in which  $x_{i,j}$  represents the  $j_{th}$  location of the  $i_{th}$  agent,  $N$  is the count of potential agent, and  $n$  represents the dimensions for the given challenge. In Eq. (18),  $rand$  is a random value from a uniform distribution,  $LB$  is lower and  $UB$  is the upper bound of the given challenge. During independent testing, multiple distributions were considered, and it has been deduced that using a uniform distribution yielded the best results.

##### 3.2.1 Encircling phase (exploration)

During the encircling phase, Crocodiles have two kinds of movements: high walking and belly walking. The RSA can alternate two search phases: encircling (exploration) and hunting (exploitation). The change is made according to four criteria and is determined by the current iteration. The position updating for the exploration phase is presented by Eq. (19).

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times -\eta_{(i,j)}(t) \times \beta - R_{(i,j)}(t) \times rand, & t \leq \frac{T}{4} \\ Best_j(t) \times x_{(r_1,j)} \times ES(t) \times rand, & t \leq 2\frac{T}{4} \text{ and } t > \frac{T}{4} \end{cases} \quad (19)$$

in which  $Best_j(t)$  represents the  $j_{th}$  location in the current optimal agent,  $rand$  signifies an arbitrary number in range  $[0, 1]$ ,  $t$  represents the ongoing iterative count, and  $T$  maximum

iterations. The value of  $\eta_{(i,j)}$  is calculated by Eq. (20), and describes hunting operator for the  $j_{th}$  position in the  $i_{th}$  solution. The value of  $\beta$  is used to control sensitivity.  $R_{(i,j)}$  is a reduced function (a value used to reduce the search area) presented by Eq. (21). The value of  $r_1$  is arbitrarily selecting from  $[1, N]$ , variable  $x_{(r_1,j)}$  denotes the  $i_{th}$  solutions arbitrary location. Evolutionary Sense, denoted as  $ES(t)$ , is a probability ratio with randomly decreasing values between  $[2, -2]$ , which is calculated using Eq. (22).

$$\eta_{(i,j)} = Best_j(t) \times P_{(i,j)} \quad (20)$$

$$R_{(i,j)} = \frac{Best_j(t) - x_{(r_2,j)}}{Best_j(t) + \epsilon} \quad (21)$$

$$ES(t) = 2 \times r_3 \times \left(1 - \frac{1}{T}\right) \quad (22)$$

in which  $P_{(i,j)}$  defines the dissimilarity in percent of the  $j_{th}$  and best attained  $j_{th}$  solution location determined via Eq. (23). Variable  $\epsilon$  stores a minimal value used to avoid division with zero. Value  $r_2$  and  $r_3$  are arbitrary values between  $[1, N]$  and  $[-1, 1]$  respectively.

$$P_{(i,j)} = \alpha + \frac{x_{(i,j)} - M(x_i)}{Best_j(t) \times (UB_{(j)} - LB_{(j)}) + \epsilon} \quad (23)$$

in which  $M(x_i)$  represents the mean location of the  $i_{th}$  agent, determined per Eq. (24).  $LB(j)$  and  $UB(j)$  are the lower and upper constraints for the  $j_{th}$  location, variable  $\alpha$  is used to regulate sensitivity.

$$M(x_i) = \frac{1}{n} \sum_{j=1}^n x_{(i,j)} \quad (24)$$

### 3.2.2 Hunting phase (exploitation)

During the hunting phase, the exploitation mechanisms of RSA are presented. During the hunting process, crocodiles perform either hunting coordination or cooperation. According to this, the position updating for the exploitation phase is presented by Eq. (25).

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times -P_{(i,j)}(t) \times rand, & t \leq 3\frac{T}{4} \text{ and } t > 2\frac{T}{4} \\ Best_j(t) - \eta_{(i,j)}(t) \times \epsilon - R_{(i,j)}(t) \times rand, & t \leq T \text{ and } t > 3\frac{T}{4} \end{cases} \quad (25)$$

in which  $Best_j(t)$  represents the  $j_{th}$  location for the best-obtained agent thus far,  $\epsilon$  a small value,  $P_{(i,j)}$ ,  $\eta_{(i,j)}$  and  $R_{(i,j)}$  are given by Eqs. (23), (20) and (21), respectively.

The Pseudocode for the described RSA is shown in 1.

**Algorithm 1** Pseudo-code of the RSA

---

```

1: Set parameter values
2: Produce a population  $X$ 
3: while ( $t < T$ ) do
4:   Determine fitness of all agents and store the best
5:   Recalculate  $ES$ 
6:   for ( Agent in population ) do
7:     for ( Agent in sub-population ) do
8:       Determine  $\eta, R, P$ 
9:       if ( $t \leq \frac{T}{4}$ ) then
10:        Utilize high walking strategy
11:       else if ( $t \leq 2\frac{T}{4}$  and  $t > \frac{T}{4}$ ) then
12:        Utilize belly walking strategy
13:       else if ( $t \leq 3\frac{T}{4}$  and  $t > 2\frac{T}{4}$ ) then
14:        Utilize Hunting coordination strategy
15:       else
16:        Utilize Hunting cooperation strategy
17:       end if
18:     end for
19:   end for
20:    $t = t + 1$ 
21: end while
22: Return best-performing agent

```

---

**3.3 Proposed modified RSA approach**

The prioritization of exploitation over exploration in RSA leads to a lack of variety among the population and early convergence. This implies that the starting positions of the solutions have a significant influence on the final outcomes. The objective of this research is to enhance the RSA algorithm by tackling the problem of limited exploration by ensuring adequate population diversity during initialization and during execution. To accomplish this, two adjustments are implemented in the elementary RSA metaheuristics: a new approach to initialization and a mechanism for preserving diverse solutions during the execution of the algorithm.

**3.3.1 New initialization scheme**

The method introduced in this study utilizes a traditional initialization equation to produce the set of individuals in the initial population:

$$X_{ij} = lb_j + \psi \cdot (ub_j - lb_j), \quad (26)$$

in which  $X_{ij}$  denotes the  $j$ -th item of the  $i$ -th solution,  $lb_j$  and  $ub_j$  represent the lower and upper constraints of the component  $j$ , and  $\psi$  is an arbitrary value drawn from the normal distribution within limits  $[0, 1]$ .

Still, the study by Rahnamayan et al. (2007) has shown that incorporation of the quasi-reflection-based learning (QRL) (Rahnamayan et al. 2007) approach to the population produced by Eq. (26) could allow the exploration of the wider search area. Consequently, for every component  $j$  belonging to the solution ( $X_j$ ), a quasi-reflective-opposite component ( $X_j^{qr}$ ) is produced in the following way:

$$X_j^{qr} = \text{rnd}\left(\frac{lb_j + ub_j}{2}, x_j\right), \quad (27)$$

where *rnd* allows to select an arbitrary number within  $\left[\frac{lb_j + ub_j}{2}, x_j\right]$  limits.

According to the QRL procedure, the proposed initialization approach doesn't affect the complexity of the algorithm with respect to *FFEs* as it produces only half of the entire population ( $NP/2$ ). The initialization procedure used in this research is outlined in Algorithm 2.

**Algorithm 2** Proposed initialization procedure that incorporates the QRL method

---

Step 1: Produce a starting population  $P^{init}$  of  $NP/2$  individuals by using Equation (26).  
 Step 2: Create QRL population  $P^{qr}$  based on  $P^{init}$  according to Eq. (27)  
 Step 3: Formulate the entire initial population  $P$  by merging  $P^{init}$  and  $P^{qr}$  ( $P \cup P^{qr}$ )  
 Step 4: Calculate the fitness of each individual that belongs to  $P$   
 Step 5: Order the agents from  $P$  according to their evaluation outcomes

---

The intensive experiments have demonstrated that this initialization scheme exhibits two important advantages. First, it enhances the diversification of the starting population, which can improve the outcomes of the algorithm at the beginning of the run. Second, it enables the algorithm to cover a wider search area with the same size of the population, allowing the initial boost to the search procedure as well.

### 3.3.2 Procedure to keep the diversity of population

To evaluate whether the algorithm's search mechanism is converging or diverging, one method is to assess the diversity of the population, which is explained in Cheng and Shi (2011). The study employs a new definition of measuring population diversity, specifically using the  $L1$  norm. This norm considers diversities resulting from two factors: the solutions generated by the population and the problem's dimensionality.

Furthermore, Cheng and Shi (2011) emphasizes the importance of data obtained from the dimension-wise element of the  $L1$  norm, which can be utilized to evaluate the search mechanism of the algorithm being studied.

Suppose  $m$  represents the number of solutions in the population, and  $n$  denotes the problem's dimensionality. The  $L1$  norm can be calculated as presented in Eqs. (28) to (30):

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij} \quad (28)$$

$$\Theta_j^p = \frac{1}{m} \sum_{i=1}^m \left| x_{ij} - \bar{x}_j \right| \quad (29)$$

$$\Theta^p = \frac{1}{n} \sum_{j=1}^n \Theta_j^p, \quad (30)$$

In this context,  $\bar{x}$  is referring to the array containing the average positions of the solutions across each dimension, while  $\Theta_j^p$  represents the array of position diversities of the

individuals, calculated using the  $L1$  norm.  $\Theta^P$  denotes the overall diversity value of the population as a scalar.

During the initial rounds of the algorithm's execution, the population's diversity should be high since the solutions are generated using the standard initialization equation (26). Nevertheless, while the method is converging to the optimal or sub-optimal solution in later rounds, the diversity should decrease dynamically. In order to tackle this, the enhanced RSA algorithm proposed in this study makes use of the  $L1$  norm for regulating the population's diversity during the entire run. This is achieved through a dynamic diversity threshold control parameter, represented by  $\Theta_t$ .

A technique has been suggested to preserve variety within a population by introducing an extra control factor, referred to as *nrs*, that specifies the number of individuals to be substituted. The approach functions in the following way: at the outset of the algorithm, the dynamic threshold for diversity, labeled as  $\Theta_{t0}$ , is established. During each round of execution, the latest population diversity, represented as  $\Theta^P$ , is assessed and compared to the dynamic diversity threshold,  $\Theta_t$ . If the condition  $\Theta^P < \Theta_t$  is met, suggesting that the population's diversity is insufficient, the worst *nrs* individuals are replaced with random solutions created utilizing a method comparable to the one used to initialize the population.

After conducting empirical simulations and theoretical analysis, the formula for computing  $\Theta_{t0}$  can be expressed in the following manner.

$$\Theta_{t0} = \sum_{j=1}^{NP} \frac{(ub_j - lb_j)}{2 \cdot NP} \quad (31)$$

As the algorithm progresses, it is anticipated that the population will gradually approach the optimal search area. Thus, the dynamic diversity threshold,  $\Theta_t$ , must be lowered from its starting value,  $\Theta_{t0}$ , which is calculated by applying the Eq. (31). To accomplish this reduction in  $\Theta_t$ , a linear decreasing function can be employed, as shown in Eq. (32), with  $T$  denoting the maximum number of rounds and  $\Theta_{t0}$  representing the initial diversity threshold.

$$\Theta_{t+1} = \Theta_t - \Theta_t \cdot \frac{t}{T}, \quad (32)$$

Here,  $t$  and  $t + 1$  represent the current as well as the next rounds. Additionally,  $T$  represents the iterative maximum. As the algorithm continues, the dynamic reduction of  $\Theta_t$  occurs, and eventually, the mechanism described will no longer be utilized, disregarding  $\Theta^P$ .

### 3.3.3 Inner workings of the suggested algorithm

Since the introduced modified RSA algorithm improved on the admirable performance of the basic RSA, it was therefore named the enhanced RSA - ERSA, and its internal structure is provided in Algorithm 3. While looking at the proposed pseudo-code, it is possible to note that the suggested modifications have been integrated into the basic variant of the RSA algorithm described in Algorithm 1 (Abualigah et al. 2022).

**Algorithm 3** ERSA pseudocode

---

```

Create a starting set of solutions ( $X$ ) by utilizing the Algorithm 2
Determine the values of  $\Theta_{t0}$  and  $\Theta_t$ 
while  $t < \text{maximum count of rounds}$  do
    Evaluate agent fitness
    Store the best agent found thus far ( $P=X^*$ )
    Update the values of  $\eta$ ,  $R$ ,  $P$  by utilizing Eq. (20), (21) and (23)
    Update the locations of solutions by using the procedure given in 1
    Determine  $\Theta^P$ 
    if ( $\Theta^P < \Theta_t$ ) then
        Replace poorest  $nrs$  individuals with agents produced as in (26)
    end if
    Update  $\Theta_t$ 
end while
Return the best-discovered agent as the global optimum
Return overall and detailed statistical outcomes and visualize

```

---

**3.4 Evaluation metrics**

The observed models' simulation outcomes have been validated by applying the collection of traditional ML measurements, namely mean squared error (MSE) calculated by Eq. (33), root mean squared error (RMSE) that can be obtained by Eq. (34), mean absolute error (MAE) specified by Eq. (36), and finally the coefficient of determination (R2) that can be determined with Eq. (36).

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{p}_i - p_i)^2 \quad (33)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{p}_i - p_i)^2} \quad (34)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{p}_i - p_i| \quad (35)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (p_i - \hat{p}_i)^2}{\sum_{i=1}^n (p_i - \bar{p})^2}, \quad (36)$$

where  $p_i$  and  $\hat{p}_i$  mark arrays that consist of the observed and predicted values, both containing  $N$  entries. This research employs MSE as the objective function with the goal to minimize it.



**Table 1** Review of CEC2019 benchmark function problems details

No	Functions	$F_i^* = F_i(x^*)$	Dim	Search range
1	Storn's Chebyshev polynomial fitting problem	1	9	$[-8192, 8192]$
2	Inverse Hilbert matrix problem	1	16	$[-16384, 16384]$
3	Lennard-Jones minimum energy cluster	1	18	$[-4, 4]$
4	Rastrigin's function	1	10	$[-100, 100]$
5	Griewangk's function	1	10	$[-100, 100]$
6	Weierstrass function	1	10	$[-100, 100]$
7	Modified Schwefel's function	1	10	$[-100, 100]$
8	Expanded Schaffer's F6 function	1	10	$[-100, 100]$
9	Happy cat function	1	10	$[-100, 100]$
10	Ackley function	1	10	$[-100, 100]$

## 4 Experiments with standard bound-constrained functions

Before evaluating the performance of ERSA metaheuristics on practical RNN tuning for wind energy time-series forecasting, in compliance with well-established practices from the modern literature, the proposed method was first tested against standard bound-constrained (unconstrained) benchmarks.

The CEC2019 test suite was chosen for this purpose due to dual reasons: this set of functions is more challenging and complex than other benchmarking suites (e.g. standard test instances and CEC2017); the basic RSA was also evaluated on them when this approach was introduced for the first time (Abualigah et al. 2022). This package consists of ten functions and its details (name, dimension, search space constraints, global optimum) have been demonstrated in Table 1.

The CEC2019 simulations were conducted with both, the introduced ERSA and the original RSA algorithms. Additionally, to make a comparative analysis more widespread, other cutting-edge metaheuristics were also considered in comparative analysis: particle swarm optimization (PSO) (Kennedy and Eberhart 1995), artificial bee colony (ABC) (Karaboga and Basturk 2008), firefly algorithm (FA) (Yang 2009), harris hawks optimization (HHO) (Heidari et al. 2019), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016) and chimpanzee swarm optimization (ChOA) (Khishe and Mosavi 2020). This particular set of algorithms was chosen to make a balance between traditional methods like PSO and more recent ones, e.g. ChOA.

Similar experimental conditions as in Abualigah et al. (2022), where maximum iterations ( $T$ ) and the agents in population count ( $N$ ) were adjusted to 500 and 30, respectively, were set for the purpose of this research as well. Additionally, due to the methods' stochastic behavior, experiments are repeated 30 independent times ( $runtime = 30$ ), with the best, worst, mean, average, and standard deviation metrics captured.

All evaluated methods were implemented specifically for this research work and results for the approaches tested also in Abualigah et al. (2022) were not taken. However, it should be pointed out that very similar results for the basic RSA, WOA, and PSO as those reported in Abualigah et al. (2022) were obtained in simulations conducted for the purpose of this work.

The performance metrics comparison between the competitive algorithms is given in Table 2, with the best-attained outcomes highlighted in bold text.

**Table 2** Results of the RSA using the CEC2019 test functions

Function	Measure	ERSA	RSA	PSO	ABC	FA	HHO	WOA	ChOA
1	Worst	<b>4.38E+04</b>	6.33E+04	2.23E+13	4.66E+10	3.15E+08	1.15E+10	1.22E+11	1.01E+09
	Average	<b>3.71E+04</b>	5.65E+04	8.12E+12	2.41E+10	1.01E+08	4.35E+09	5.81E+10	5.12E+08
	Best	<b>3.22E+04</b>	5.06E+04	8.26E+11	4.33E+09	5.03E+05	1.29E+09	1.55E+09	2.57E+08
	STD	<b>4.65E+03</b>	4.93E+03	8.60E+12	2.16E+10	1.45E+08	4.19E+09	4.73E+10	2.76E+08
	Rank	<b>1</b>	2	8	7	3	5	6	4
2	Worst	1.73E+01	1.73E+01	1.63E+04	1.73E+01	1.73E+01	1.75E+01	1.74E+01	1.35E+02
	Average	1.73E+01	1.73E+01	1.27E+04	1.73E+01	1.73E+01	1.75E+01	1.74E+01	8.34E+01
	Best	1.73E+01	1.73E+01	8.32E+03	1.73E+01	1.73E+01	1.74E+01	1.73E+01	4.35E+01
	STD	<b>4.56E-08</b>	1.26E-07	3.13E+03	2.41E-04	3.45E-05	4.68E-02	6.19E-03	3.06E+01
	Rank	1	1	5	1	1	3	2	4
3	Worst	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01
	Average	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01
	Best	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01
	STD	0.00E+00	5.43E-05	0.00E+00	<b>5.64E-17</b>	4.29E-06	3.42E-05	7.43E-07	5.23E-05
	Rank	1	1	1	1	1	1	1	1
4	Worst	<b>1.32E+01</b>	2.71E+01	2.36E+01	1.15E+02	1.65E+02	3.61E+03	8.64E+02	2.72E+01
	Average	<b>9.13E+00</b>	2.45E+01	1.73E+01	5.63E+01	7.54E+01	1.95E+03	5.86E+02	2.43E+01
	Best	5.02E+00	2.13E+01	<b>4.96E+00</b>	1.11E+01	4.88E+01	1.08E+03	2.25E+02	2.07E+01
	STD	3.41E+00	2.44E+00	6.53E+00	3.73E+01	5.13E+01	9.55E+02	2.63E+02	<b>2.40E+00</b>
	Rank	<b>1</b>	4	2	5	6	8	7	3
5	Worst	1.11E+00	1.09E+00	1.66E+00	1.09E+00	2.68E+00	2.68E+00	1.55E+00	1.71E+00
	Average	1.06E+00	1.05E+00	1.45E+00	1.05E+00	2.41E+00	2.37E+00	1.26E+00	1.54E+00
	Best	1.00E+00	1.00E+00	1.25E+00	1.00E+00	2.17E+00	2.17E+00	1.12E+00	1.31E+00
	STD	5.79E-02	4.66E-02	1.63E-01	4.23E-02	2.57E-01	1.99E-01	<b>1.59E-01</b>	1.66E-01
	Rank	2	1	4	1	7	6	3	5
6	Worst	8.76E+00	<b>8.54E+00</b>	1.25E+01	1.21E+01	1.21E+01	1.16E+01	1.13E+01	1.14E+01
	Average	<b>5.21E+00</b>	5.32E+00	9.85E+00	1.14E+01	1.16E+01	1.10E+01	9.62E+00	1.06E+01
	Best	<b>2.94E+00</b>	3.05E+00	8.96E+00	1.05E+01	1.14E+01	9.69E+00	8.27E+00	9.15E+00
	STD	2.66E-01	<b>2.31E-01</b>	9.48E-01	4.72E-01	2.38E-01	7.46E-01	1.12E+00	8.05E-01
	Rank	<b>1</b>	2	4	7	8	6	3	5
7	Worst	<b>1.44E+02</b>	6.51E+02	4.13E+02	1.41E+03	6.63E+02	1.05E+03	1.15E+03	6.06E+02
	Average	<b>4.69E+01</b>	4.48E+02	2.88E+02	6.55E+02	3.55E+02	8.52E+02	8.13E+02	2.57E+02
	Best	<b>8.84E+01</b>	2.79E+02	3.43E+02	1.85E+02	1.26E+02	4.79E+02	4.64E+02	2.91E+01
	STD	<b>9.73E+01</b>	1.53E+02	1.73E+02	4.45E+02	2.31E+02	2.26E+02	2.54E+02	2.93E+02
	Rank	<b>1</b>	5	2	6	4	8	7	3
8	Worst	<b>4.92E+00</b>	6.55E+00	5.65E+00	6.11E+00	6.85E+00	6.66E+00	6.64E+00	6.63E+00
	Average	<b>4.10E+00</b>	5.62E+00	4.96E+00	5.71E+00	5.49E+00	5.88E+00	6.11E+00	5.87E+00
	Best	<b>2.55E+00</b>	4.66E+00	4.39E+00	5.13E+00	2.92E+00	5.35E+00	5.65E+00	4.61E+00
	STD	9.24E-01	7.41E-01	6.37E-01	4.76E-01	1.76E+00	4.64E-01	<b>4.24E-01</b>	8.13E-01
	Rank	<b>1</b>	4	2	5	3	7	8	6
9	Worst	2.47E+00	2.45E+00	2.44E+00	2.86E+00	7.16E+00	4.55E+02	6.42E+00	<b>2.43E+00</b>
	Average	2.46E+00	2.41E+00	2.42E+00	2.62E+00	4.95E+00	2.05E+02	5.43E+00	<b>2.36E+00</b>
	Best	2.45E+00	2.36E+00	2.38E+00	2.43E+00	3.33E+00	4.54E+01	4.06E+00	<b>2.35E+00</b>
	STD	<b>7.39E-07</b>	3.17E-02	3.41E-02	1.76E-01	1.52E+00	1.55E+02	9.59E-01	1.09E-02
	Rank	4	2	3	5	6	8	7	<b>1</b>

**Table 2** (continued)

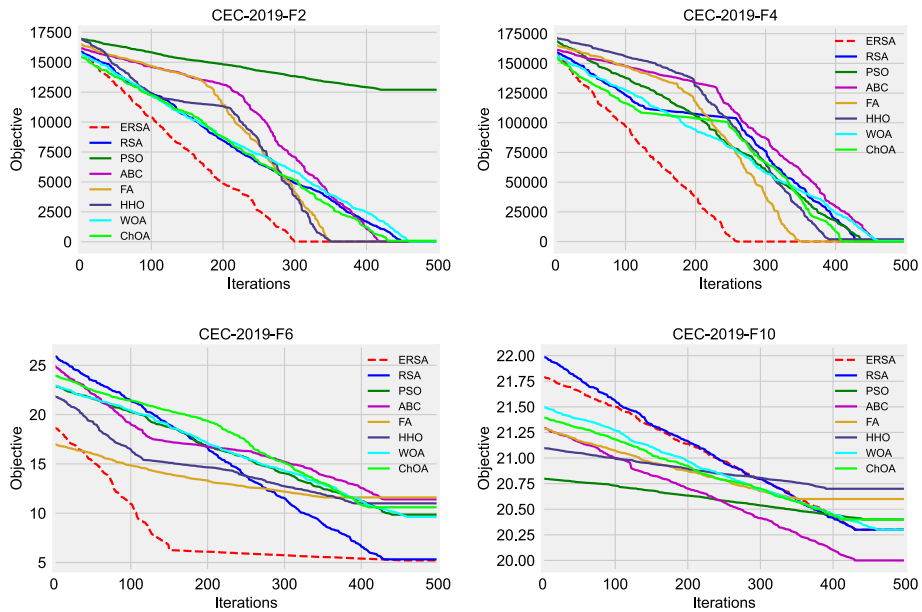
Function	Measure	ERSA	RSA	PSO	ABC	FA	HHO	WOA	ChOA
10	Worst	2.06E+01	2.04E+01	2.07E+01	<b>2.01E+01</b>	2.07E+01	2.08E+01	2.05E+01	2.05E+01
	Average	2.03E+01	2.03E+01	2.04E+01	<b>2.00E+01</b>	2.06E+01	2.07E+01	2.03E+01	2.04E+01
	Best	2.01E+01	2.02E+01	2.01E+01	<b>2.00E+01</b>	2.05E+01	2.06E+01	2.02E+01	2.04E+01
	STD	3.45E-01	1.36E-01	1.85E-01	<b>3.97E-02</b>	4.72E-02	6.64E-02	1.09E-01	5.43E-02
	Rank	2	2	3	1	4	5	2	3
Mean	rank	<b>1.5</b>	2.4	3.4	3.9	4.3	5.7	4.6	3.5
Final	Ranking	<b>1</b>	2	3	5	6	8	7	4

The outcomes of the CEC2019 function testing indicate that on average the proposed ERSA demonstrates the most admirable performance. It is also apparent that this is not the case across all test functions. Nevertheless, this is to be expected, as per the NFL theorem of optimization no single algorithm is equally effective across all applications. Therefore, experimentation is required to determine algorithms best suited to a given problem.

Further analysis and observations help us understand the strengths and weaknesses of the introduced algorithms in comparison to contemporary methods. Interestingly, in certain functions such as F9, while the algorithm did not display the best performance, it nevertheless, attained the best STD score indicating that, while the introduced algorithm isn't the optimal solution by a small margin, it is the most robust and reliable option. This observation is somewhat mirrored for function F4, where despite attaining a good score for both the best and average run, the algorithm does not demonstrate the highest reliability, further emphasizing the importance of extensive experimentation and the NFL theorem.

A more direct improvement can be observed in F2, where despite attaining identical scores for best, average, and worst runs between the RSA and introduced ERSA, the introduced algorithm attained a significant improvement in robustness as demonstrated by a decrease in the STD. In F10, both algorithms performed slightly less favorably than the ABC algorithm, with the original RSA performing slightly better in the works run, while the introduced ERSA attained better performance in the best run, evening out average performance. For test function F3, most metaheuristics performed similarly well. In most other cases the introduced algorithms achieved a performance increase compared to the original as well as also outperformed competing metaheuristics in these cases.

Finally, metaheuristic ranking has been done according to average performance scores, with the best-performing algorithms receiving the lower ranks, while less optimal algorithms received progressively higher rankings. It can be observed that the proposed algorithm attained the best ranking in the majority of cases, closely followed by the original RSA algorithm. Average rankings across all functions are also shown, where the introduced ERSA algorithm attained an average rank of 1.5, followed closely by the original RSA which attained an average rank of 2.4. Based on these rankings it can be determined that the introduced ERSA performed admirably well in comparison with contemporary algorithms, while also improving on the admirable performance maintained by the RSA. It is also worth noting that all evaluations and algorithms have been independently implemented and tested. Furthermore, the attained results are in line with previous works (Abualigah et al. 2022) that have similarly evaluated the original RSA algorithm against several state-of-the-art algorithms.



**Fig. 2** Convergence speed graphs for some arbitrarily CEC2019 functions

Finally, to visualize the performance of evaluated methods, convergence speed graphs for arbitrarily chosen instances depicted for the mean run are shown in Fig. 2. In cases of  $F2$ ,  $F4$ , and  $F8$  benchmarks, the ERSa manages to converge relatively fast, however, for the  $F10$  test, the proposed method exhibits relatively stable, but not the best performance.

## 5 Utilized datasets and basic experimental setup

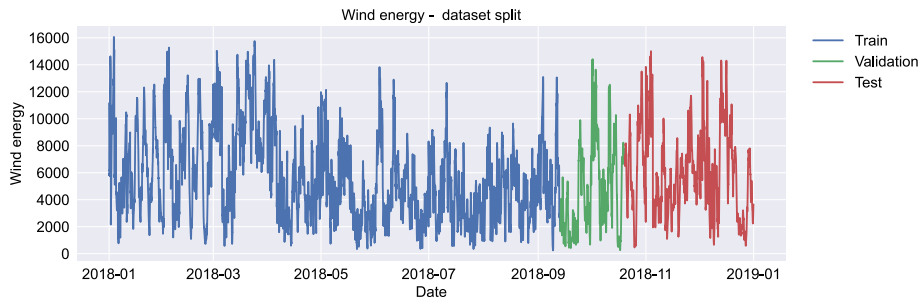
### 5.1 Overview of wind generation datasets

As already noted in Introduction, to evaluate the performance of LSTM models for multi-variate time-series forecasting, two datasets were employed. In this part of the manuscript, a brief description of both datasets is provided. For easier following, the first dataset is labeled as “wind dataset”, while the second is titled as “wind farm dataset”.

#### 5.1.1 Wind dataset

The hourly energy demand generation and weather dataset, available online<sup>1</sup> has been compiled from two primary sources. The first source covered electrical generation and consumption data for Spain. This data has been provided by the ENTSOE public portal for Transmission Service Operator (TSO) data. The provided data covers several sources of

<sup>1</sup> [https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather?select=energy\\_dataset.csv](https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather?select=energy_dataset.csv).



**Fig. 3** Wind dataset split shown on target feature

power such as biomass, geothermal, wind, solar as well as fossil power. The second source includes relevant meteorological data for Valencia Spain. The information for this segment of the dataset has been provided by the Weather API available online.<sup>2</sup> It covers hourly resolution data for temperature, pressure, humidity, wind speed, wind direction as well as rainfall.

The compiled dataset covers a total of 4 years worth of weather, load, and generation data for Spain with an hourly resolution, for the years 2015–2018. The available information makes this compiled data an excellent contender for forecasting power generation based on available meteorological data. For this research, the onshore power generation data has been used as the target variable while the available weather data has been used as inputs.

However, since this is a relatively large dataset, only the period from January 1, 2018 to December 31, 2018, was taken for simulations and it consists of 8759 observations. The dataset training-validation-testing split demonstrated on the target feature can be seen in Fig. 3

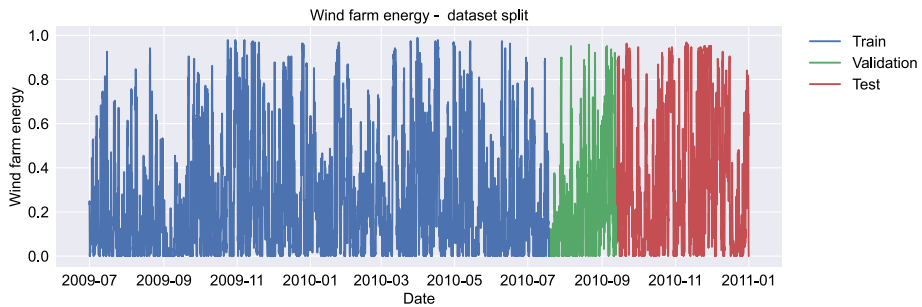
### 5.1.2 Wind farm dataset

Formerly a competition dataset the GEFCom2012 challenge (Hong et al. 2014) for a time was available on Kaggle,<sup>3</sup> the wind dataset has been repurposed for use in wind energy generation forecasting. It was originally introduced with the aim of improving forecasting practices and their utility across industries and serves to bridge the connection between academic research and industry practice. The dataset continues its legacy in this work where it is utilized to assess the proposed models' performance when tackling the complex task of windpower generation forecasting.

The datasets contained encompass weather and generation data for 7 anonymized wind farms in mainland China. The data concerning the power generation of the farms have been normalized to ensure anonymity. Windfarm data is accompanied by 24 h forecasts of relevant wind meteorological data created every 12 h. Wind speed and wind direction are provided alongside zonal and meridional wind components for each wind farm (Fig. 4).

<sup>2</sup> <https://openweathermap.org/api>.

<sup>3</sup> <https://www.kaggle.com/competitions/global-energy-forecasting-competition-2012-load-forecasting/data>.



**Fig. 4** Wind farm dataset split shown on target feature

For experimental purposes the relevant meteorological data has been trimmed into 12 h predictions in every forecast, to create an hourly resolution. This was then combined with the available normalized real-world wind generation data for each respective wind farm on an hourly basis. The original dataset covered four and a half years' worth of hourly resolution data, with the final half years' worth of data reserved for testing. However, the final half year's worth of data has not been made available.

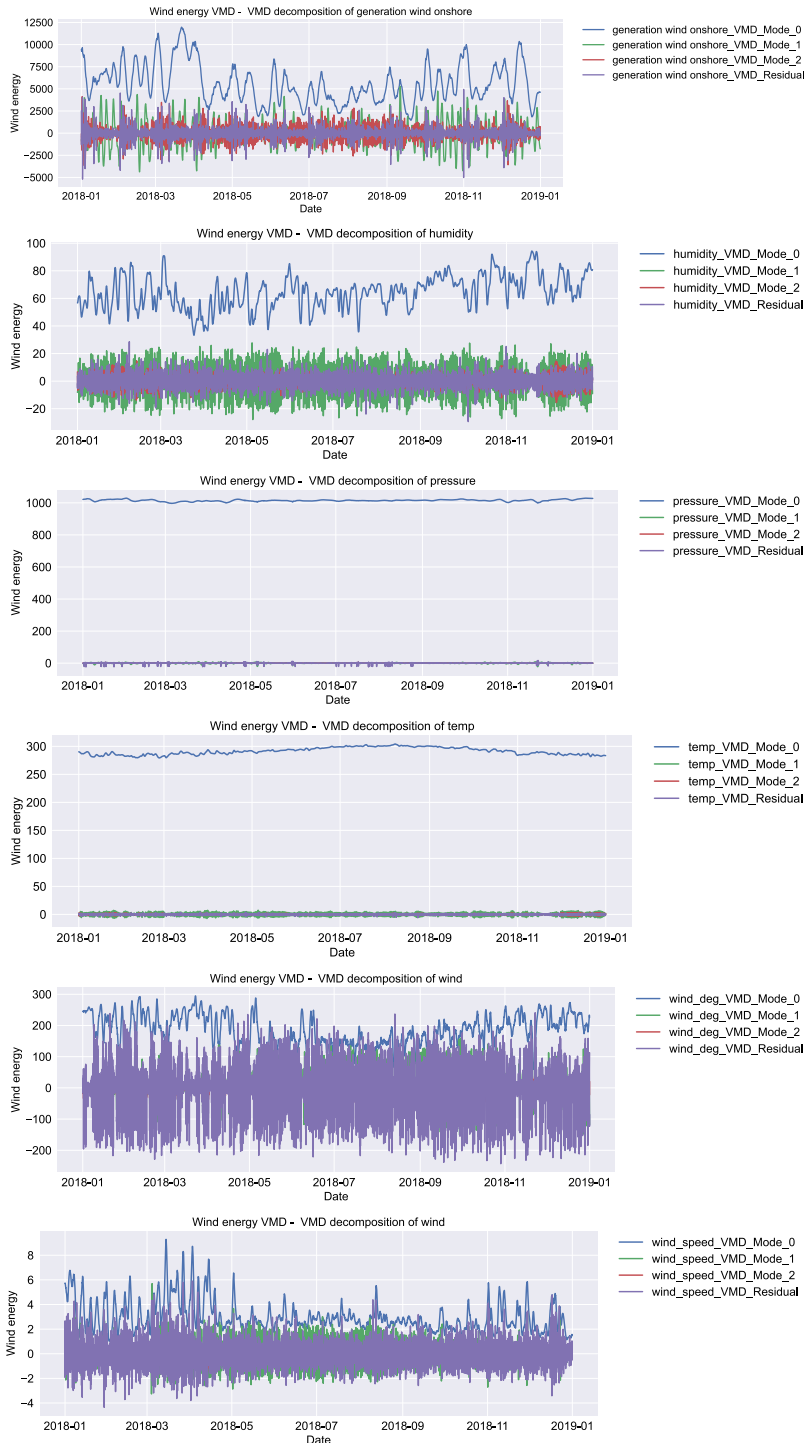
Due to the huge amount of observations available in this dataset, as well as missing values in the later parts of the data, during experimentation a reduced portion of the available data needed to be used. The huge amount of data makes training and evaluating models very resource intensive. Therefore, the experimental dataset used in simulations covers 2 years' worth of data (from January 1, 2009 to December 31, 2010) for a single anonymized wind farm. The final portion of the utilized dataset contains a total of 13176 instances for wind farm 2. The dataset training-validation-testing split demonstrated on the target feature can be seen in Fig. 3.

## 5.2 Decomposition

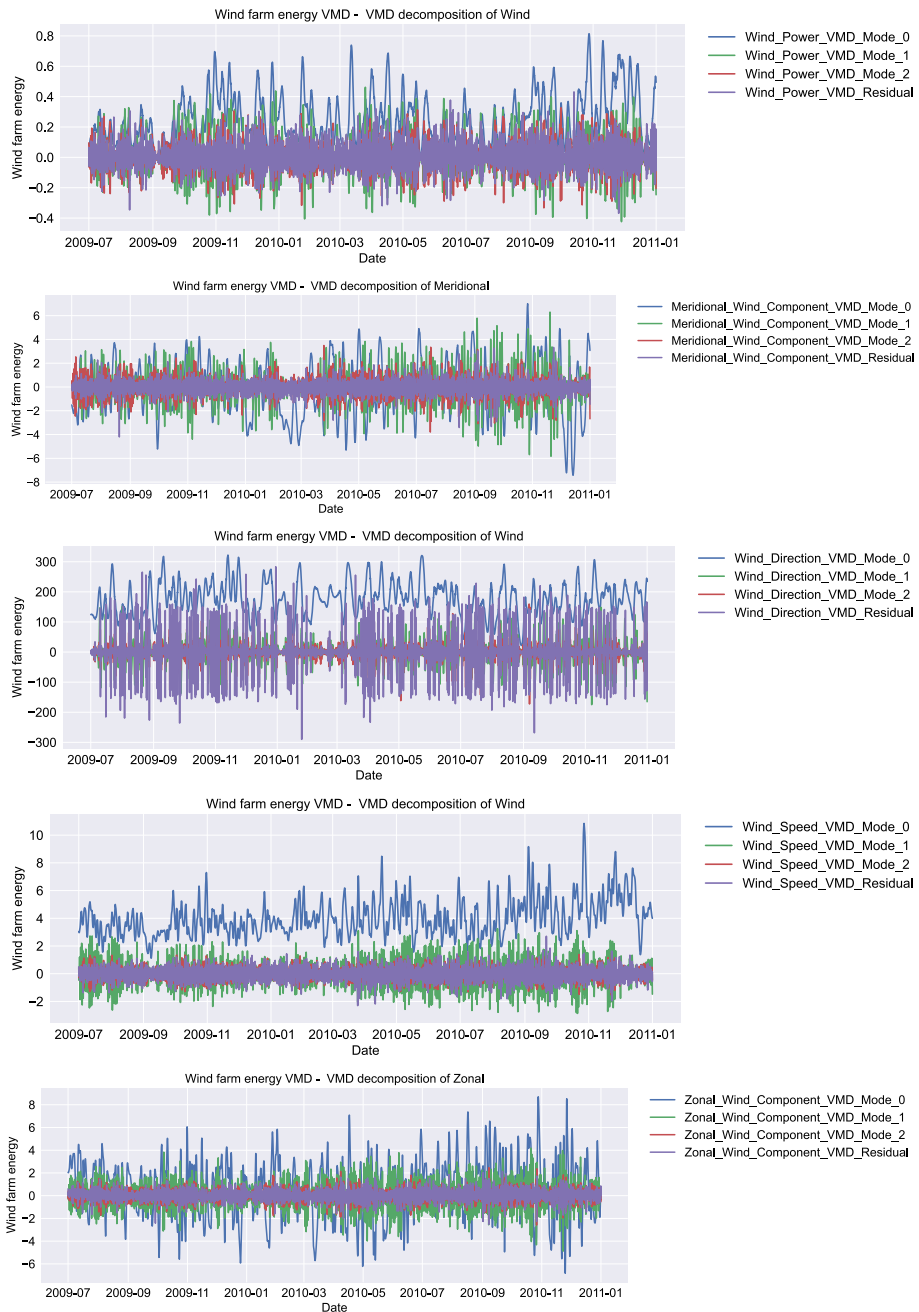
The initial stage of experimentation involves applying decomposition techniques to the available data input features. This is done so as to divide the complex input feature signals into a series of simpler sub-signals, that could later be used for forecasting. This process does, however, increase the number of features a network needs to handle.

The two techniques tested in this work include VMD (Dragomiretskiy and Zosso 2013) and EMD (Huang et al. 1998). The VMD has been carried out with the  $k$  value of three, resulting in a total of four signals. Three signals represent attained modes, while the fourth is the residual values that were not encompassed by a mode. All VMD modes and residuals for the wind and wind farm datasets are shown in Figs. 5 and 6 respectively.

Similarly, the EMD technique number of IMF's was limited to a maximum of four. It is important to note that, once a new IMF cannot be extracted via decomposition EMD terminates the process early, resulting in fewer respective components. In addition to the IMF components, an additional residual component is added that consists of signals that could not be assigned to an IMF. This results in a maximum of 5 sub-signals per input feature. All EMD IMF's and residuals for the wind and wind farm datasets are shown in Figs. 7 and 8 respectively.

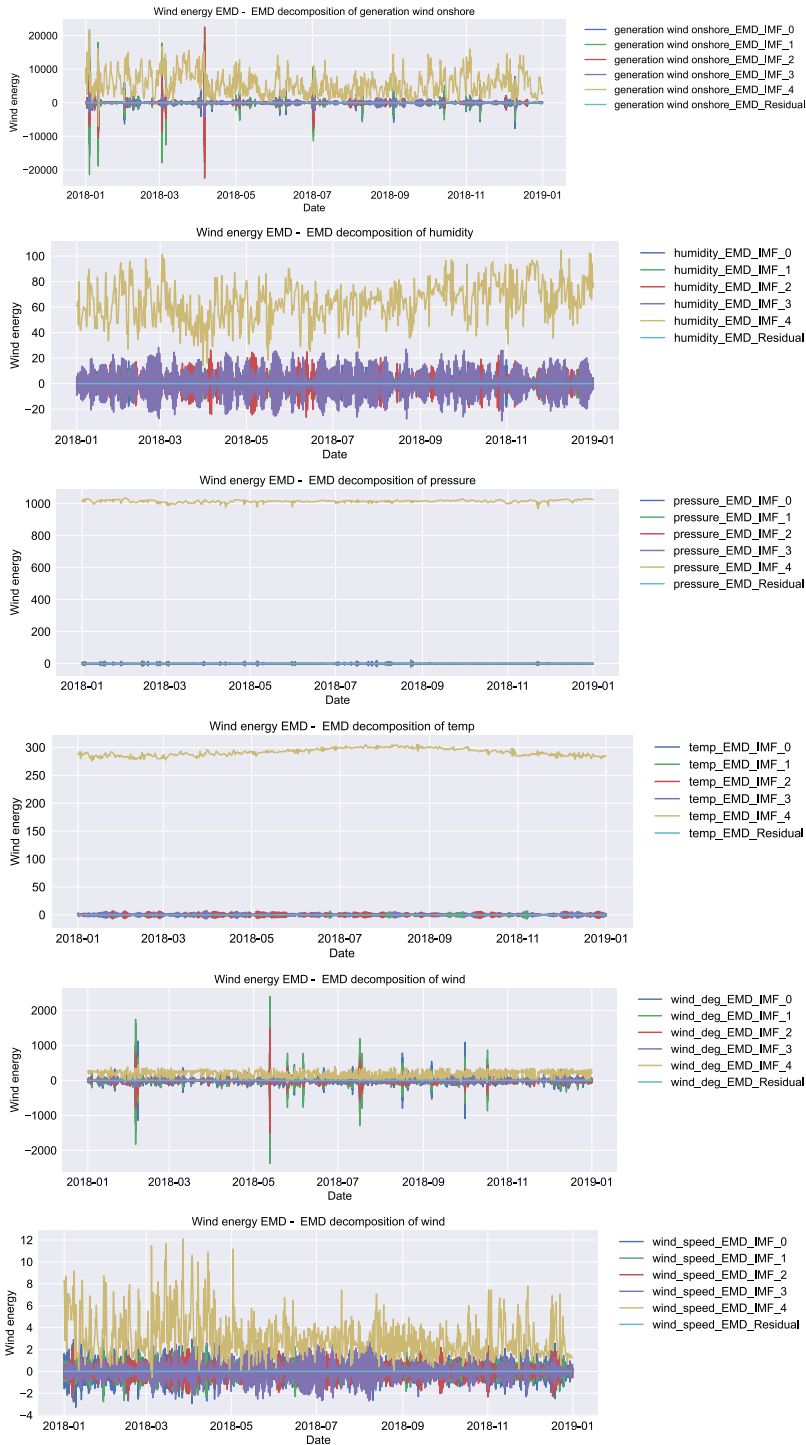


**Fig. 5** Wind dataset input feature VMD modes and residuals

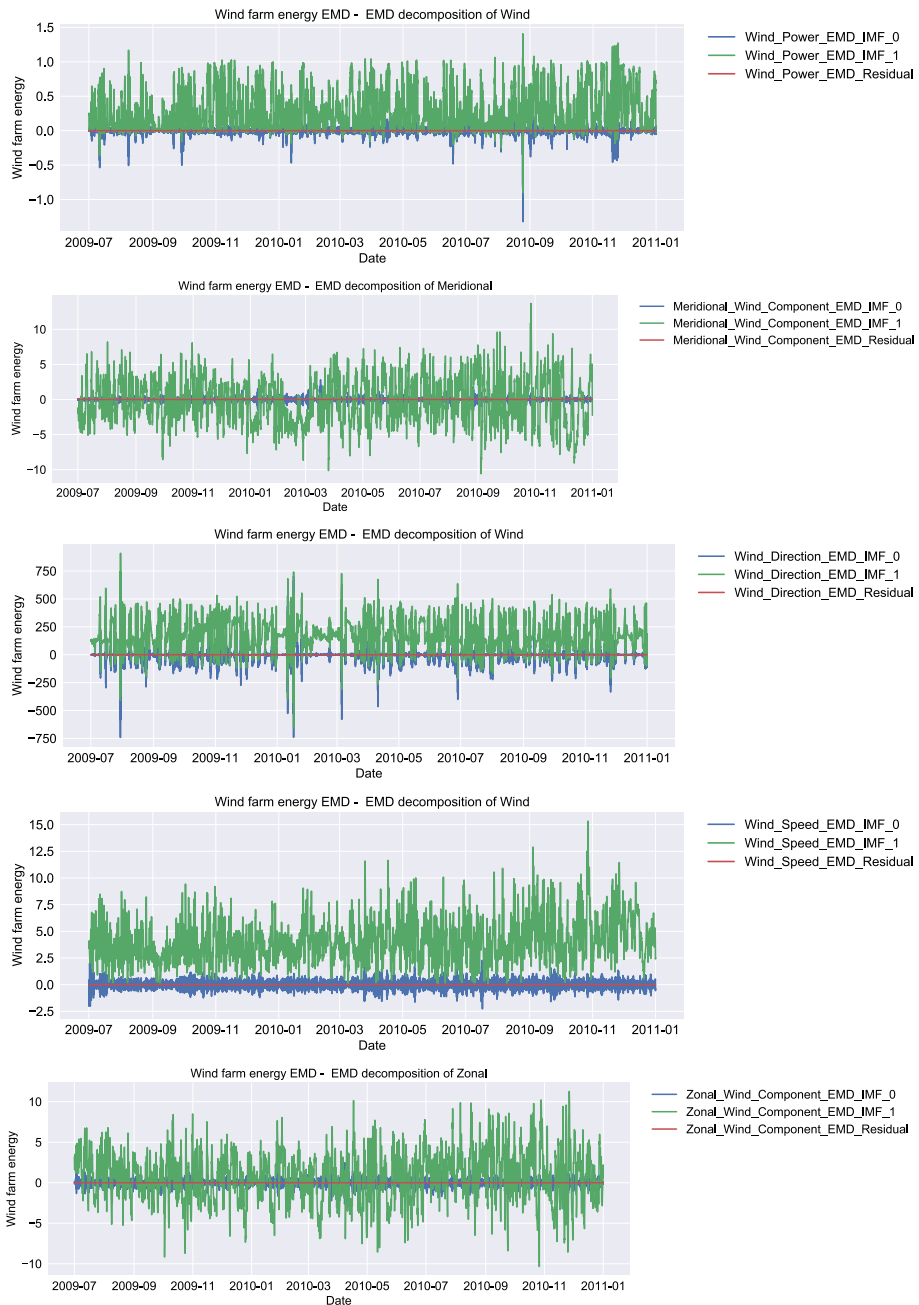


**Fig. 6** Wind farm dataset input feature VMD modes and residuals





**Fig. 7** Wind dataset input feature EMD IMF's and residuals



**Fig. 8** Wind farm dataset input feature EMD IMF's and residuals

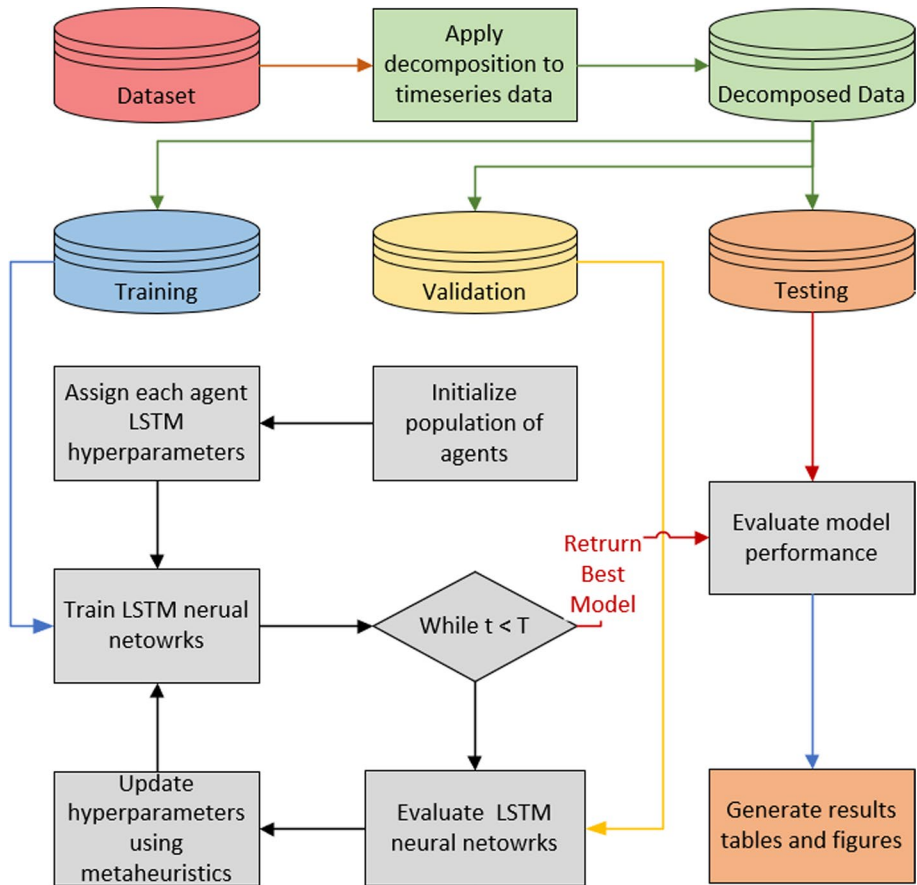


Fig. 9 Flowchart of the forecasting process

### 5.3 Experimental setup

The experimental process involves two stages. Initially, the available data for both datasets were subjected to decomposition. Following this process, the signal components and residual signals were fed into LSTM models tasked with forecasting. Each model was given six points of input data and challenged with making forecasts three steps ahead. A flowchart of the described process is shown in Fig. 9.

Several state-of-the-art metaheuristics were challenged to optimize the parameters of prediction models in order to improve performance. The evaluated models include the introduced ERSA as well as the original RSA. Additionally, several well-known optimization algorithms were included in the comparative analysis including PSO (Kennedy and Eberhart 1995), ABC (Karaboga and Basturk 2008), FA (Yang 2009), HHO (Heidari et al. 2019), WOA (Mirjalili and Lewis 2016), ChOA (Khishe and Mosavi 2020). The evaluated optimization algorithms were assigned a population size of five agents and allowed eight iterations to improve solutions. Additionally, to account for intrinsic

**Table 3** The LSTM hyperparameters included in the optimization and their respective constraints

Parameter name	Lower constraint	Upper constraint
Learning rate	0.0001	0,01
Training epochs	300	600
Dropout rate	0.05	0.2
Network layers	1	2
Neurons in layer 1	100	300
Network in layer 2	100	300

randomness associated with metaheuristic algorithms results were evaluated through 30 independent executions to help attain objective evaluations.

All tested algorithms were tasked with selecting LSTM hyperparameters. The parameter subset selected for optimization was selected due to the high impact on model performance. Optimized parameters and their possible ranges are as follows: the neuron count in the layers determined from range [100, 300], the learning rate range [0.0001, 0, 01], training epochs [300, 600], dropout rate [0.05, 0.2], the total number of network layers between [1, 2] and the number of neurons in the second network layer [100, 300]. The parameters and their respective constraints are highlighted in Table 3 Finally, early stopping has been implemented in order to help prevent model over-fitting. The threshold for early stopping has been empirically determined as  $\frac{epochs}{3}$ . Meaning that if a model should not improve for  $\frac{epochs}{3}$  model training is terminated early. An added benefit of this approach is the reduction in wasted computation resources. The utilized ranges were determined empirically to give the best outcomes considering the computational costs of optimization and performance outcomes.

Experiments were carried out using the Python programming languages. Additionally, standard ML and AI libraries were utilized including Keras, TensorFlow, Scikit-learn. The visuals were generated using Seaborn and Matplotlib libraries. To facilitate experimentation a machine with an Intel i9 11900K CPU, 128GB of ram memory and a RTX4070 GPU was employed.

## 6 Achieved results, comparative analysis, and discussion

In this segment the experimental outcomes on two observed datasets are delivered, namely wind and wind farms. First, the results without decomposition are shown, followed by the results with VMD and EMD employed. Last but not least, this section also provides a SHAP analysis of the best-performing model on each of the two observed datasets. In all tables that contain experimental results, the best outcomes in every regarded category are marked in bold text.

### 6.1 Wind dataset experimental results

The following demonstrated the outcomes on the wind dataset, without decomposition. Table 4 shows the overall metrics in terms of the best, worst, mean, and median values, accompanied by the standard deviance and variance values over 30 separate runs of each regarded algorithm. LSTM-ERSA model accomplished the supreme results in terms of the

**Table 4** Wind dataset overall metrics for best, worst, mean, and median run without decomposition

Method	Best	Worst	Mean	Median	Std	Var
LSTM-ERSA	<b>0.006592</b>	0.006653	<b>0.006616</b>	0.006636	2.30E−05	5.30E−10
LSTM-RSA	0.006611	0.006643	0.006631	0.006635	1.27E−05	1.61E−10
LSTM-PSO	0.006631	0.006672	0.006646	0.006634	1.83E−05	3.36E−10
LSTM-ABC	0.006605	0.006638	0.006627	0.006631	1.24E−05	1.54E−10
LSTM-FA	0.006628	0.006698	0.006649	0.006639	2.53E−05	6.38E−10
LSTM-HHO	0.006632	0.006651	0.006644	0.006648	<b>7.00E−06</b>	<b>4.90E−11</b>
LSTM-WOA	0.006607	0.006655	0.006633	0.006638	1.82E−05	3.30E−10
LSTM-ChOA	0.006609	<b>0.006626</b>	0.006619	<b>0.006625</b>	7.32E−06	5.36E−11

best and median values. The second-best result was scored by the LSTM-ABC method, while LSTM-WOA attained the third-best score. Otherwise, LSTM-ChOA achieved the best results for the worst and median metrics. Finally, LSTM-HHO established the best standard deviation and variance scores, suggesting that it provided the steadiest results across the runs.

Table 5 brings forward the detailed metrics of every prediction step regarding the best run of each algorithm. The prefix L is used to denote that LSMT is used. It can be noted that the suggested LSTM-ERSA attained supreme results for two-step, three-samples forward, and overall results, in terms of the objective—MSE, but also for other important indicators, namely  $R^2$ , MAE (except for two-samples forward) and RMSE. The best scores for one sample forward were achieved by the LSTM-ABC approach. Looking at the overall results, the second-best algorithm was LSTM-ABC, in front of the LSTM-WOA and LSTM-ChOA methods. The observed LSTM-ERSA attained the best overall MSE value of 0.006592, in front of LSTM-ABC with an MSE value of 0.006605.

**Table 5** Wind dataset detailed metrics for each prediction step of the best run without decomposition

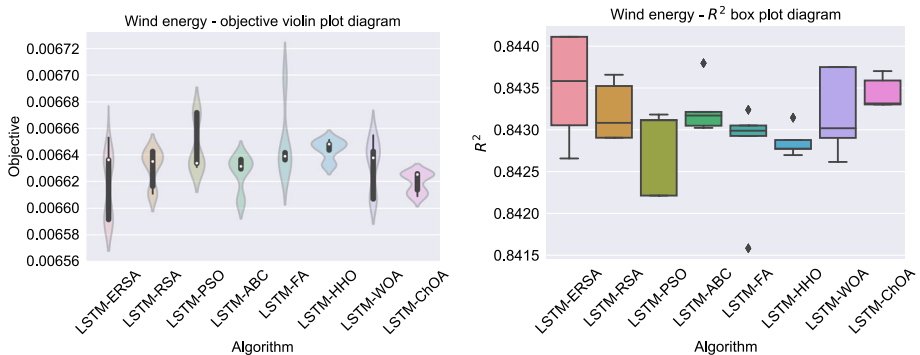
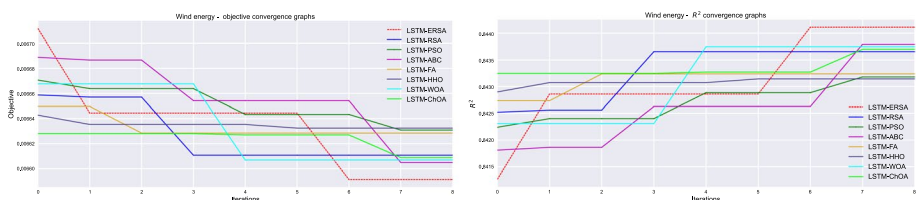
	Error indicator	L-ERSA	L-RSA	L-PSO	L-ABC	L-FA	L-HHO	L-WOA	L-ChOA
One	$R^2$	<b>0.845391</b>	0.845365	0.844410	0.845795	0.843997	0.844825	0.845621	0.845740
Sample	MAE	0.051652	0.051778	0.051980	0.051555	0.051913	0.051905	0.051622	<b>0.051469</b>
Forward	MSE	0.006537	0.006539	0.006579	<b>0.006520</b>	0.006596	0.006561	0.006528	0.006523
Forecasts	RMSE	0.080855	0.080861	0.081110	<b>0.080749</b>	0.081218	0.081002	0.080794	0.080763
Two	$R^2$	<b>0.845174</b>	0.844677	0.844587	0.844842	0.844313	0.844381	0.844986	0.844649
Samples	MAE	0.052106	0.052294	0.052345	0.052263	0.052233	0.052475	<b>0.052025</b>	0.052158
Forward	MSE	<b>0.006547</b>	0.006568	0.006571	0.006561	0.006583	0.006580	0.006555	0.006569
Forecasts	RMSE	<b>0.080911</b>	0.081041	0.081064	0.080998	0.081136	0.081118	0.080960	0.081048
Three	$R^2$	<b>0.841770</b>	0.840932	0.840548	0.840750	0.841407	0.840229	0.840642	0.840716
Samples	MAE	<b>0.053340</b>	0.053716	0.053675	0.053775	0.053446	0.053942	0.053581	0.053602
Forward	MSE	<b>0.006690</b>	0.006725	0.006741	0.006733	0.006705	0.006755	0.006737	0.006734
Forecasts	RMSE	<b>0.081795</b>	0.082011	0.082110	0.082058	0.081888	0.082192	0.082085	0.082067
Overall	$R^2$	<b>0.844111</b>	0.843657	0.843181	0.843795	0.843238	0.843144	0.843749	0.843701
Forecast	MAE	<b>0.052365</b>	0.052596	0.052666	0.052530	0.052530	0.052773	0.052409	0.052409
Outcomes	MSE	<b>0.006591</b>	0.006610	0.006630	0.006604	0.006627	0.006631	0.006606	0.006608
	RMSE	<b>0.081187</b>	0.081305	0.081429	0.081269	0.081414	0.081439	0.081281	0.081294

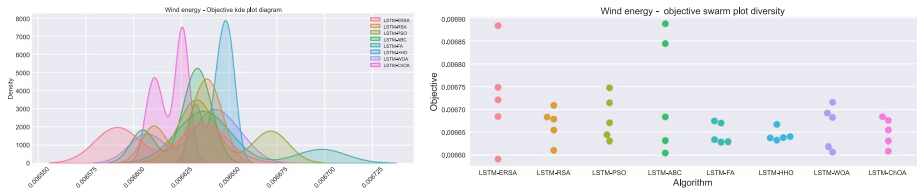
**Table 6** Parameters selected by metaheuristics for best-performing wind prediction models without decomposition

Method	Neurons layer 1	Learning rate	Epochs	Dropout	Layers	Neurons layer 2
LSTM-ERSA	140	0.000894	517	0.108587	2	170
LSTM-RSA	100	0.003232	318	0.091040	2	150
LSTM-PSO	100	0.002880	300	0.050317	2	200
LSTM-ABC	167	0.001091	547	0.081649	2	188
LSTM-FA	100	0.010000	300	0.200000	1	118
LSTM-HHO	184	0.002868	346	0.175802	2	158
LSTM-WOA	131	0.001016	600	0.055888	2	129
LSTM-ChOA	100	0.003440	592	0.183099	1	177

The best set of LSTM parameters produced by the top-performing run of each metaheuristic is shown in Table 6. The proposed LSTM-ERSA established the LSTM structure as follows: 140 neurons in the first layer, a learning rate of 0.000894, 517 epochs, a dropout value of 0.108587, and 170 neurons in the second layer.

Aiming to provide better insight into the results, visualizations are provided in Figs. 10, 11, and 12. Figure 10 shows the violin plots for the objective function (MSE), accompanied by the box plot of the  $R^2$  indicator, for all 30 runs. After that, the convergence diagrams of the objective function and  $R^2$  for the best run of each algorithm are given in Fig. 11. It can

**Fig. 10** Wind dataset objective function and  $R^2$  distribution plots for each metaheuristic without decomposition**Fig. 11** Wind dataset objective function and  $R^2$  convergence plots for each metaheuristic without decomposition



**Fig. 12** Wind dataset objective swarm and KDE plots for each metaheuristic without decomposition

be noted that at the beginning, RSA and WOA metaheuristics are converging faster, but are overrun by the proposed ERSA in the final rounds of execution. Lastly, the Kernel Density Estimation (KDE) diagram and swarm plot are shown in Fig. 12. KDE diagram is used to show the probability density function and can indicate whether or not the results are coming from the normal distribution. The swarm plot shows the diversity of the solutions during the final round of the best-performing run of each algorithm.

### 6.1.1 Wind dataset with VMD

This section presents the experimental outcomes on the wind dataset when VMD has been applied. Table 7 shows the overall metrics in terms of the best, worst, mean, and median values, accompanied by the standard deviation and variance values over 30 separate runs of each regarded algorithm. VMD-LSTM-ERSA model accomplished the supreme results in terms of the best, mean, and median values. The second-best result was scored by the VMD-LSTM-ChOA method, while VMD-LSTM-RSA attained the third-best score. Otherwise, VMD-LSTM-HHO achieved the best result for the worst metric. Finally, VMD-LSTM-HHO also established the best standard deviation and variance scores, suggesting that it provided the steadiest results over 30 independent runs.

Table 8 brings forward the detailed metrics of every prediction step regarding the best run of each algorithm. The prefix VL is used to denote that VMD-LSMT is used. It can be noted that the suggested VMD-LSTM-ERSA attained superior results for one-step and overall results, in terms of the objective—MSE, but also for other important indicators, namely  $R^2$ , MAE, and RMSE. The best scores for the two-samples forward were achieved by the VMD-LSTM-WOA approach, while VMD-LSTM-HHO attained the best outcomes for three-samples forward predictions. Looking at the overall results, the second-best

**Table 7** Wind dataset overall metrics for best, worst, mean, and median run using VMD

Method	Best	Worst	Mean	Median	Std	Var
VMD-LSTM-ERSA	<b>0.001704</b>	0.001841	<b>0.001750</b>	<b>0.001733</b>	5.16E−05	2.66E−09
VMD-LSTM-RSA	0.001729	0.001817	0.001784	0.001800	3.23E−05	1.05E−09
VMD-LSTM-PSO	0.001759	0.001879	0.001813	0.001809	4.39E−05	1.93E−09
VMD-LSTM-ABC	0.001779	0.001866	0.001809	0.001792	3.79E−05	1.44E−09
VMD-LSTM-FA	0.001730	0.001814	0.001756	0.001743	3.21E−05	1.03E−09
VMD-LSTM-HHO	0.001758	<b>0.001811</b>	0.001771	0.001761	<b>2.19E−05</b>	<b>4.80E−10</b>
VMD-LSTM-WOA	0.001762	0.001820	0.001784	0.001765	2.69E−05	7.25E−10
VMD-LSTM-ChOA	0.001726	0.001824	0.001766	0.001743	3.68E−05	1.35E−09

**Table 8** Wind dataset detailed metrics for each prediction step of the best run using VMD

	Error indicator	VL-ERSA	VL-RSA	VL-PSO	VL-ABC	VL-FA	VL-HHO	VL-WOA	VL-ChOA
One	R <sup>2</sup>	<b>0.941916</b>	0.940073	0.937082	0.938810	0.939588	0.936426	0.935198	0.938841
Sample	MAE	<b>0.032073</b>	0.032604	0.033309	0.032766	0.032744	0.033708	0.033620	0.032859
Forward	MSE	<b>0.002456</b>	0.002534	0.002660	0.002587	0.002554	0.002688	0.002740	0.002586
Forecasts	RMSE	<b>0.049558</b>	0.050338	0.051579	0.050866	0.050541	0.051847	0.052346	0.050853
Two	R <sup>2</sup>	0.963863	0.965451	0.965805	0.962887	0.964136	0.964482	<b>0.965907</b>	0.964385
Samples	MAE	0.023949	0.023585	0.023410	0.024447	0.024082	0.023929	<b>0.023259</b>	0.023829
Forward	MSE	0.001528	0.001461	0.001446	0.001569	0.001516	0.001502	<b>0.001442</b>	0.001506
Forecasts	RMSE	0.039090	0.038221	0.038025	0.039614	0.038942	0.038753	<b>0.037968</b>	0.038806
Three	R <sup>2</sup>	0.973313	0.971810	0.972330	0.972100	0.973517	<b>0.974396</b>	0.973880	0.974280
Samples	MAE	0.020805	0.021806	0.021357	0.021592	0.020839	<b>0.020161</b>	0.020653	0.020345
Forward	MSE	0.001127	0.001191	0.001169	0.001179	0.001119	<b>0.001082</b>	0.001103	0.001086
Forecasts	RMSE	0.033590	0.034523	0.034203	0.034346	0.033462	<b>0.032902</b>	0.033232	0.032976
Overall	R <sup>2</sup>	<b>0.959696</b>	0.959111	0.958405	0.957931	0.959080	0.958434	0.958328	0.959168
Forecasts	MAE	<b>0.025608</b>	0.025998	0.026025	0.026268	0.025888	0.025932	0.025843	0.025677
Outcomes	MSE	<b>0.001703</b>	0.001728	0.001758	0.001778	0.001729	0.001757	0.001761	0.001725
	RMSE	<b>0.041280</b>	0.041579	0.041936	0.042175	0.041595	0.041922	0.041975	0.041550



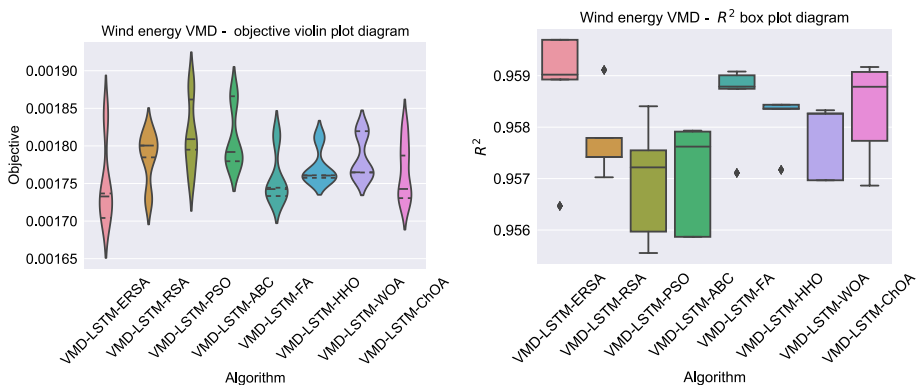
**Table 9** Parameters selected by metaheuristics for best-performing wind prediction models using VMD

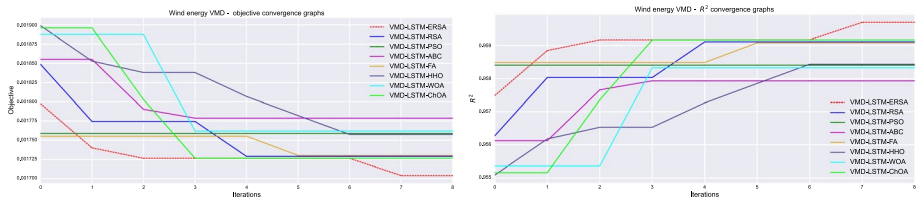
Method	Neurons layer 1	Learning rate	Epochs	Dropout	Layers	Neurons layer 2
VMD-LSTM-ERSA	100	0.010000	563	0.200000	2	100
VMD-LSTM-RSA	100	0.010000	600	0.073598	2	104
VMD-LSTM-PSO	139	0.007558	600	0.050000	2	138
VMD-LSTM-ABC	174	0.006837	509	0.164527	2	139
VMD-LSTM-FA	143	0.010000	600	0.098000	2	130
VMD-LSTM-HHO	171	0.008113	572	0.200000	2	162
VMD-LSTM-WOA	200	0.005682	600	0.055617	2	200
VMD-LSTM-ChOA	100	0.008296	600	0.050000	2	146

algorithm was VMD-LSTM-ChOA, in front of the VMD-LSTM-RSA method. The observed VMD-LSTM-ERSA attained the best overall MSE value of 0.001704, in front of VMD-LSTM-ChOA with an MSE value of 0.001726.

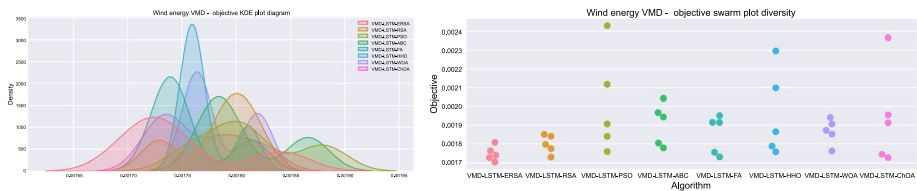
The best set of LSTM parameters produced by the top-performing run of each metaheuristic is shown in Table 9. The proposed VMD-LSTM-ERSA established the LSTM structure as follows: 100 neurons in the first layer, a learning rate of 0.010000, 563 epochs, a dropout value of 0.200000, and 100 neurons in the second layer.

Aiming to provide better insight into the results, visualizations are provided in Figs. 13, 14, and 15. Figure 13 depicts the violin plots for the objective function (MSE), accompanied by the box plot of the  $R^2$  indicator, for all 30 runs. After that, the convergence diagrams of the objective function and  $R^2$  for the best run of each algorithm are given in Fig. 14. It can be noted that in this case, the proposed ERSA exhibits the fastest convergence from the beginning to the end. Lastly, the KDE diagram and swarm plot for the objective function is shown in Fig. 15. The swarm plot shows the diversity of the solutions during the final round of the best-performing run of each algorithm, and it can be noted that all the solutions of the proposed ERSA are in close proximity to the optimum in this case.

**Fig. 13** Wind dataset objective function and  $R^2$  distribution plots for each metaheuristic with VMD



**Fig. 14** Wind dataset objective function and  $R^2$  convergence plots for each metaheuristic with VMD



**Fig. 15** Wind dataset objective swarm and KDE plots for each metaheuristic with VMD

### 6.1.2 Wind dataset with EMD

This section presents the experimental outcomes on the wind dataset when EMD has been applied. Table 10 shows the overall metrics in terms of the best, worst, mean, and median values, accompanied by the standard deviation and variance values over 30 separate runs of each regarded algorithm. The EMD-LSTM-ERSA model accomplished superior results in terms of the best, mean, and median scores. The second-best result was attained by the EMD-LSTM-PSO method, while EMD-LSTM-FA obtained the third-best score. Otherwise, EMD-LSTM-PSO achieved the best result for the worst metric. Finally, EMD-LSTM-RSA established the best standard deviation and variance scores, suggesting that it provided the steadiest results over 30 independent runs in this scenario.

Table 11 brings forward the detailed metrics of every prediction step regarding the best run of each algorithm. The prefix EL is used to denote that EMD-LSMT is used. In this scenario, it can be noted that the suggested EMD-LSTM-ERSA attained supreme outcomes for all metrics: one-step, two-step, three-samples forward and overall results, in

**Table 10** Wind dataset overall metrics for best, worst, mean, and median run using EMD

Method	Best	Worst	Mean	Median	Std	Var
EMD-LSTM-ERSA	<b>0.004831</b>	0.005199	<b>0.005015</b>	<b>0.005015</b>	1.84E−04	3.39E−08
EMD-LSTM-RSA	0.005139	0.005183	0.005156	0.005139	<b>2.15E−05</b>	<b>4.64E−10</b>
EMD-LSTM-PSO	0.004994	<b>0.005081</b>	0.005032	0.004994	4.28E−05	1.83E−09
EMD-LSTM-ABC	0.005179	0.005537	0.005358	0.005358	1.79E−04	3.20E−08
EMD-LSTM-FA	0.005065	0.005198	0.005136	0.005198	6.62E−05	4.39E−09
EMD-LSTM-HHO	0.005123	0.005229	0.005179	0.005229	5.31E−05	2.82E−09
EMD-LSTM-WOA	0.005130	0.005184	0.005155	0.005130	2.66E−05	7.06E−10
EMD-LSTM-ChOA	0.005126	0.005173	0.005149	0.005149	2.35E−05	5.52E−10

**Table 11** Wind dataset detailed metrics for each prediction step of the best run using EMD

	Error indicator	EL-ERSA	EL-RSA	EL-PSO	EL-ABC	EL-FA	EL-HHO	EL-WOA	EL-ChOA
One	$R^2$	<b>0.882618</b>	0.874563	0.876980	0.877066	0.875932	0.870262	0.872794	0.875291
Sample	MAE	<b>0.046090</b>	0.048054	0.047590	0.047374	0.047323	0.048471	0.048793	0.048225
Forward	MSE	<b>0.004963</b>	0.005304	0.005202	0.005198	0.005246	0.005486	0.005379	0.005273
Forecasts	RMSE	<b>0.070451</b>	0.072828	0.072123	0.072098	0.072430	0.074066	0.073340	0.072617
Two	$R^2$	<b>0.886238</b>	0.876369	0.883291	0.871967	0.882816	0.884023	0.882208	0.879532
Samples	MAE	<b>0.045484</b>	0.047946	0.046415	0.048618	0.045821	0.045980	0.046948	0.047558
Forward	MSE	<b>0.004810</b>	0.005228	0.004935	0.005414	0.004955	0.004904	0.004981	0.005094
Forecasts	RMSE	<b>0.069356</b>	0.072302	0.070249	0.073578	0.070392	0.070028	0.070574	0.071371
Three	$R^2$	<b>0.888399</b>	0.884489	0.885392	0.883520	0.881879	0.882265	0.881007	0.881513
Samples	MAE	<b>0.045805</b>	0.046869	0.046693	0.046724	0.046394	0.047158	0.047604	0.047710
Forward	MSE	<b>0.004718</b>	0.004883	0.004845	0.004924	0.004994	0.004977	0.005030	0.005009
Forecasts	RMSE	<b>0.068693</b>	0.069886	0.069612	0.070189	0.070671	0.070556	0.070931	0.070780
Overall	$R^2$	<b>0.885751</b>	0.878473	0.881887	0.877517	0.880208	0.878849	0.878669	0.878778
Forecasts	MAE	<b>0.045792</b>	0.047622	0.046899	0.047571	0.046512	0.047202	0.047781	0.047830
Out-comes	MSE	<b>0.004830</b>	0.005138	0.004993	0.005178	0.005064	0.005122	0.005129	0.005125
	RMSE	<b>0.069503</b>	0.071683	0.070669	0.071964	0.071169	0.071572	0.071625	0.071593

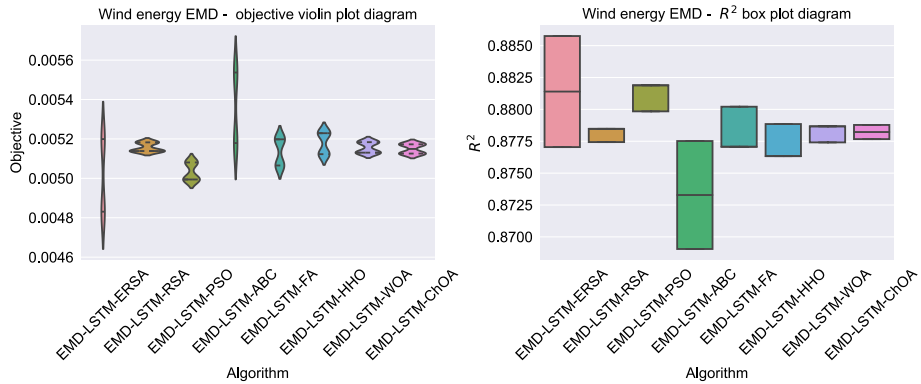
terms of the objective—MSE, and all other important indicators, namely  $R^2$ , MAE and RMSE. Looking at the overall results, the second-best algorithm was EMD-LSTM-PSO, in front of the EMD-LSTM-FA method. The proposed EMD-LSTM-ERSA attained the best overall MSE value of 0.004831, in front of EMD-LSTM-PSO which achieved an MSE value of 0.004994.

The best set of LSTM parameters produced by the top-performing run of each metaheuristic for the scenario with EMD employed is shown in Table 12. The proposed EMD-LSTM-ERSA established the LSTM structure as follows: 108 neurons in the first layer, a learning rate of 0.007837, 600 epochs, a dropout value of 0.050000, and 151 neurons in the second layer.

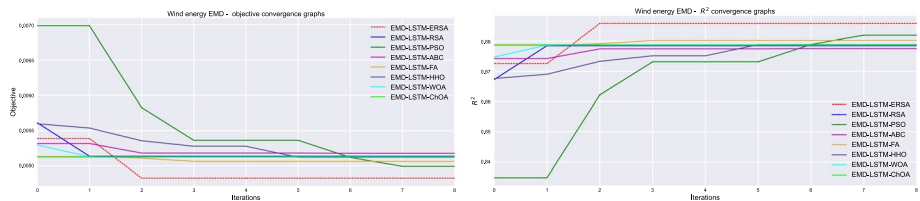
Aiming to provide better insight into the results, visualizations are provided in Figs. 16, 17, and 18. Figure 16 depicts the violin plots for the objective function (MSE), accompanied by the box plot of the  $R^2$  indicator, for all 30 runs. After that, the convergence

**Table 12** Parameters selected by metaheuristics for the respective best-performing wind prediction models using EMD

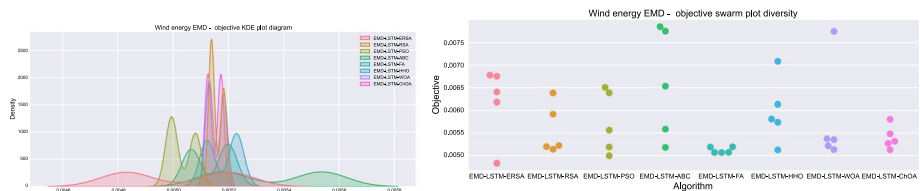
Method	Neurons layer 1	Learning rate	Epochs	Dropout	Layers	Neurons layer 2
EMD-LSTM-ERSA	108	0.007837	600	0.050000	2	151
EMD-LSTM-RSA	200	0.010000	600	0.119344	2	100
EMD-LSTM-PSO	121	0.008541	600	0.173318	2	164
EMD-LSTM-ABC	165	0.007747	600	0.119184	2	133
EMD-LSTM-FA	155	0.010000	600	0.200000	2	101
EMD-LSTM-HHO	106	0.006599	570	0.062141	2	112
EMD-LSTM-WOA	100	0.010000	600	0.184404	2	118
EMD-LSTM-ChOA	136	0.004410	549	0.114572	2	104



**Fig. 16** Wind dataset objective function and  $R^2$  distribution plots for each metaheuristic with EMD



**Fig. 17** Wind dataset objective function and  $R^2$  convergence plots for each metaheuristic with EMD



**Fig. 18** Wind dataset objective swarm and KDE plots for each metaheuristic with EMD

diagrams of the objective function and  $R^2$  for the best run of each algorithm are given in Fig. 17. It can be noted that in this case, the proposed ERSA exhibits the fastest speed of convergence from the beginning to the end. Lastly, the KDE diagram and swarm plot for the objective function is shown in Fig. 18. The swarm plot shows the diversity of the solutions during the final round of the best-performing run of each algorithm.

### 6.1.3 Comparison with other models on the Wind dataset

To demonstrate the comparative performance improvements of the coupling optimizers with decomposition's techniques in the introduced methodology the best outcomes of

**Table 13** Comparison of the best performing methods with other contemporary prediction models applying to the wind dataset

Method	MSE	R <sup>2</sup>
SVM	0.007283	0.814233
Random forest	0.007303	0.772332
XGBoost	0.007126	0.807384
KELM	0.007200	0.801634
AdaBoost	0.007203	0.802330
ANN (3 layers)	0.007190	0.791192
RNN (3 layers)	0.007199	0.805466
LSTM	0.007233	0.824233
LSTM-ERSA	0.006591	0.844111
VMD-LSTM-ERSA	<b>0.001703</b>	<b>0.0959696</b>
EMD-LSTM-ERSA	0.004830	0.885751

each approach have been compared to several contemporary prediction models. The outcomes of the objective (MSE) and indicator R<sup>2</sup> factions is shown in Table 13.

As demonstrated in Table 13, outcome attained by applying VMD followed by LSTM networks optimized via introduced metaheuristics demonstrate notable improvements compared to their approaches applied to the same task.

## 6.2 Wind farm dataset experimental results

This section presents the results of the wind farm dataset, without decomposition. Table 14 shows the overall metrics in terms of the best, worst, mean, and median values, accompanied by the standard deviance and variance values over 30 separate runs of each regarded algorithm. LSTM-ERSA model accomplished the supreme results in terms of the best, worst, and mean values. The second-best result was scored by the LSTM-HHO method, while LSTM-ABC attained the third-best score. Otherwise, LSTM-ABC achieved the best results for the median metric. Last but not least, LSTM-RSA established the best standard deviation and variance scores, suggesting that it provided the steadiest results across the runs.

Table 15 brings forward the detailed metrics of every prediction step regarding the best run of each algorithm. The prefix L is used to denote that LSMT is used. It can be noted that the suggested LSTM-ERSA attained supreme results for two samples forward and overall results,

**Table 14** Wind farm dataset overall metrics for best, worst, mean, and median run without decomposition

Method	Best	Worst	Mean	Median	Std	Var
LSTM-ERSA	<b>0.020566</b>	<b>0.020809</b>	<b>0.020724</b>	0.020803	9.69E−05	9.38E−09
LSTM-RSA	0.020829	0.020884	0.020864	0.020872	<b>1.88E−05</b>	<b>3.54E−10</b>
LSTM-PSO	0.020754	0.020847	0.020809	0.020821	3.43E−05	1.18E−09
LSTM-ABC	0.020673	0.020857	0.020745	<b>0.020722</b>	6.59E−05	4.34E−09
LSTM-FA	0.020747	0.021040	0.020911	0.020983	1.10E−04	1.22E−08
LSTM-HHO	0.020608	0.020925	0.020807	0.020869	1.23E−04	1.52E−08
LSTM-WOA	0.020730	0.020881	0.020807	0.020826	5.68E−05	3.23E−09
LSTM-ChOA	0.020699	0.020947	0.020775	0.020727	8.98E−05	8.07E−09

**Table 15** Wind farm dataset detailed metrics for each prediction step of the best run without decomposition

	Error indicator	L-ERSA	L-RSA	L-PSO	L-ABC	L-FA	L-HHO	L-WOA	L-ChOA
One	$R^2$	0.771359	0.768250	0.770386	0.769249	0.769645	<b>0.775001</b>	0.769093	0.767537
Sample	MAE	<b>0.102667</b>	0.102964	0.102777	0.102829	0.102680	0.102882	0.102924	0.102791
Forward	MSE	0.022502	0.022808	0.022597	0.022709	0.022670	<b>0.022143</b>	0.022724	0.022878
Forecasts	RMSE	0.150005	0.151022	0.150324	0.150696	0.150566	<b>0.148806</b>	0.150746	0.151254
Two	$R^2$	<b>0.799068</b>	0.796724	0.796229	0.797961	0.796895	0.796674	0.797062	0.796279
Sample	MAE	0.100233	0.099696	0.100523	0.099993	0.100032	0.101472	0.100125	<b>0.099405</b>
Forward	MSE	<b>0.019775</b>	0.020005	0.020054	0.019884	0.019988	0.020010	0.019972	0.020049
Forecasts	RMSE	<b>0.140622</b>	0.141440	0.141612	0.141009	0.141381	0.141457	0.141322	0.141595
Three	$R^2$	0.802662	0.800087	0.800731	0.802615	0.801019	0.800123	0.801910	<b>0.805205</b>
Sample	MAE	0.103533	0.103045	0.103602	0.102485	0.102978	0.104569	0.103026	<b>0.100661</b>
Forward	MSE	0.019420	0.019673	0.019610	0.019424	0.019581	0.019670	0.019494	<b>0.019170</b>
Forecasts	RMSE	0.139357	0.140263	0.140038	0.139374	0.139936	0.140251	0.139623	<b>0.138457</b>
Overall	$R^2$	<b>0.791029</b>	0.788353	0.789115	0.789941	0.789186	0.790598	0.789354	0.789673
Forecast	MAE	0.102144	0.101901	0.102300	0.101768	0.101896	0.102974	0.102024	<b>0.100951</b>
Outcomes	MSE	<b>0.020565</b>	0.020828	0.020753	0.020672	0.020746	0.020607	0.020729	0.020698
	RMSE	<b>0.143406</b>	0.144321	0.144061	0.143779	0.144037	0.143554	0.143980	0.143871

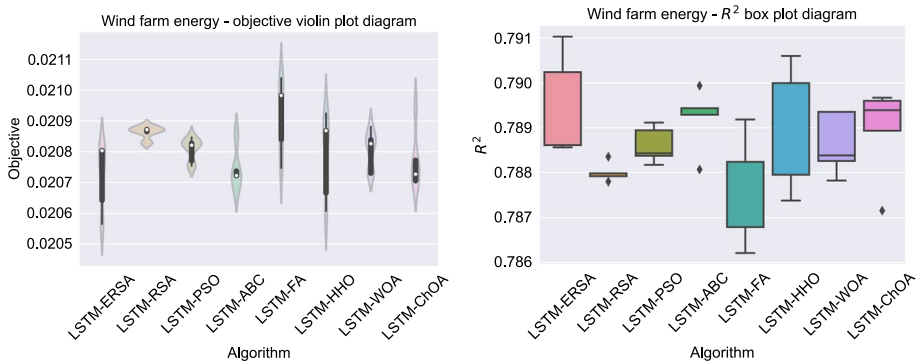
in terms of the objective—MSE, but also for other important indicators, namely  $R^2$  and RMSE (except MAE). The best scores for one-samples forward were achieved by the LSTM-HHO approach, while the best outcomes for three-samples forward were attained by LSTM-ChOA. Looking at the overall results, the second-best algorithm was LSTM-HHO, in front of LSTM-ABC and LSTM-ChOA methods. The observed LSTM-ERSA attained the best overall MSE value of 0.020566, in front of LSTM-HHO with an MSE value of 0.020608.

The best set of LSTM parameters produced by the top-performing run of each metaheuristic for this scenario is shown in Table 16. The proposed LSTM-ERSA established the LSTM structure as follows: 180 neurons in the first layer, a learning rate of 0.005919, 439 epochs, a dropout value of 0.165101, and 200 neurons in the second layer for this particular scenario.

Aiming to provide better insight into the results, visualizations are provided in Figs. 19, 20, and 21. Figure 19 shows the violin plots for the objective function

**Table 16** Parameters selected by metaheuristics for best-performing wind farm prediction models without decomposition

Method	Neurons layer 1	Learning rate	Epochs	Dropout	Layers	Neurons layer 2
LSTM-ERSA	180	0.005919	439	0.165101	2	200
LSTM-RSA	137	0.010000	553	0.200000	2	156
LSTM-PSO	200	0.005864	480	0.196612	2	176
LSTM-ABC	117	0.007639	527	0.099844	2	142
LSTM-FA	186	0.010000	578	0.200000	2	151
LSTM-HHO	176	0.006243	485	0.108088	2	150
LSTM-WOA	197	0.006295	543	0.198430	2	186
LSTM-ChOA	100	0.010000	473	0.050467	2	134

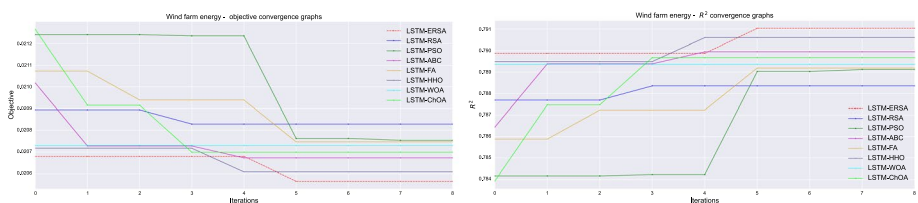


**Fig. 19** Wind farm dataset objective function and  $R^2$  distribution plots for each metaheuristic without decomposition

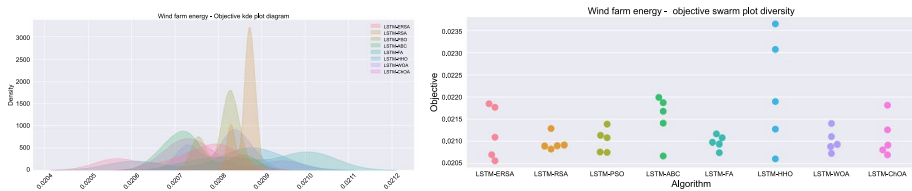
(MSE), accompanied by the box plot of the  $R^2$  indicator, for all 30 runs. After that, the convergence diagrams of the objective function and  $R^2$  for the best run of each algorithm are given in Fig. 20. It can be noted that at the beginning, HHO exhibited a bit faster convergence at one point, however, it was overrun by the proposed ERSA in the final rounds of execution. Finally, the KDE diagram and swarm plot are shown in Fig. 21. KDE diagram is used to show the probability density function and can indicate whether or not the outcomes originate from the normal distribution. The swarm plot shows the diversity of the solutions during the final round of the best-performing run of each algorithm.

### 6.2.1 Wind farm dataset with VMD

This section presents the results on the wind farm dataset, with employed VMD. Table 17 shows the overall metrics in terms of the best, worst, mean, and median values, accompanied by the standard deviation and variance values over 30 separate runs of each regarded algorithm. VMD-LSTM-ERSA model accomplished the supreme results in terms of the best, worst, mean, and median values. The second-best result was scored by the VMD-LSTM-FA method, while VMD-LSTM-PSO attained the third-best score. VMD-LSTM-ERSA also achieved the best results for the standard deviation. Last but not least, LSTM-WOA established the best variance score.



**Fig. 20** Wind farm dataset objective function and  $R^2$  convergence plots for each metaheuristic without decomposition



**Fig. 21** Wind farm dataset objective swarm and KDE plots for each metaheuristic without decomposition

**Table 17** Wind farm dataset overall metrics for best, worst, mean, and median run using VMD

Method	Best	Worst	Mean	Median	Std	Var
VMD-LSTM-ERSA	<b>0.006702</b>	<b>0.006969</b>	<b>0.006802</b>	<b>0.006781</b>	<b>8.53E−05</b>	7.96E−09
VMD-LSTM-RSA	0.006849	0.007406	0.007135	0.007072	2.24E−04	5.00E−08
VMD-LSTM-PSO	0.006845	0.007311	0.007140	0.007176	1.60E−04	2.56E−08
VMD-LSTM-ABC	0.006899	0.007219	0.007021	0.006956	1.33E−04	1.76E−08
VMD-LSTM-FA	0.006747	0.007083	0.006867	0.006806	1.35E−04	1.81E−08
VMD-LSTM-HHO	0.006911	0.007258	0.007046	0.006917	1.61E−04	2.59E−08
VMD-LSTM-WOA	0.006898	0.007104	0.007027	0.007057	8.60E−05	<b>7.40E−09</b>
VMD-LSTM-CHOA	0.006873	0.007132	0.006947	0.006881	9.95E−05	9.90E−09

Table 18 brings forward the detailed metrics of every prediction step regarding the best run of each algorithm. The prefix VL is used to denote that VMD-LSMT is used. It can be noted that the suggested VMD-LSTM-ERSA attained the best scores for overall results, in terms of the objective—MSE, but also for other important indicators, namely  $R^2$  and RMSE (except MAE). The best scores for one sample forward were achieved by the VMD-LSTM-HHO approach, the best results for two-samples forward were obtained by the VMD-LSTM-RSA, while the best outcomes for three-samples forward were attained by VMD-LSTM-FA. Looking at the overall results, the second-best algorithm was VMD-LSTM-FA, in front of VMD-LSTM-PSO. The proposed VMD-LSTM-ERSA attained the best overall MSE value of 0.006702, in front of VMD-LSTM-FA with an MSE value of 0.006747.

The best set of LSTM parameters produced by the top-performing run of each metaheuristic for this scenario is shown in Table 19. The proposed LSTM-ERSA established the LSTM structure as follows: 142 neurons in the first layer, a learning rate of 0.009502, 600 epochs, a dropout value of 0.151008, and 127 neurons in the second layer for this particular scenario.

Aiming to provide better insight into the results, visualizations are provided in Figs. 22, 23, and 24. Figure 22 shows the violin plots for the objective function (MSE), accompanied by the box plot of the  $R^2$  indicator, for all 30 runs. After that, the convergence diagrams of the objective function and  $R^2$  for the best run of each algorithm are given in Fig. 23. It can be noted that at the beginning, FA exhibited slightly faster convergence, however, it was overrun by the proposed ERSA in the final rounds of execution. Finally, the KDE diagram and swarm plot are shown in Fig. 24. KDE diagram is used to show the probability density function and can indicate whether or not the

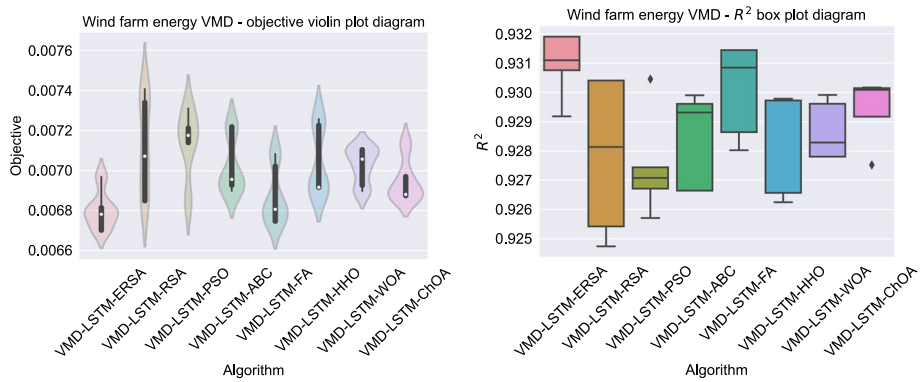
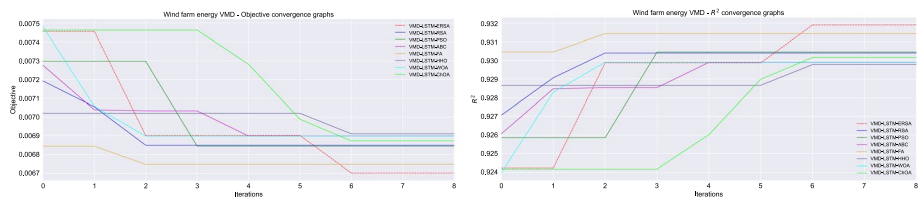
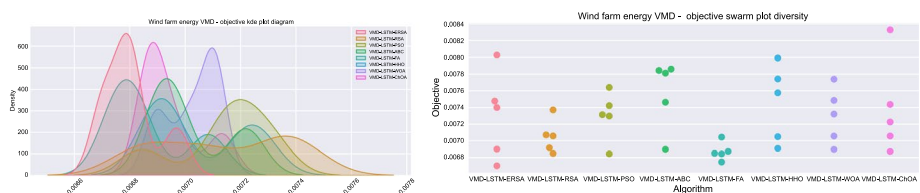


**Table 18** Wind farm dataset detailed metrics for each prediction step of the best run using VMD

	Error indicator	VL-ERSA	VL-RSA	VL-PSO	VL-ABC	VL-FA	VL-HHO	VL-WOA	VL-ChOA
One	R <sup>2</sup>	0.881572	0.877731	0.885070	0.878542	0.878918	<b>0.885834</b>	0.881587	0.883030
Sample	MAE	0.075029	0.076310	0.075125	0.076096	0.075810	<b>0.074311</b>	0.075867	0.074936
Forward	MSE	0.011655	0.012033	0.011311	0.011953	0.011916	<b>0.011236</b>	0.011653	0.011512
Forecasts	RMSE	0.107959	0.109695	0.106352	0.109331	0.109161	<b>0.105998</b>	0.107951	0.107292
Two	R <sup>2</sup>	0.946913	<b>0.946955</b>	0.944295	0.946192	0.946887	0.944130	0.944561	0.944182
Samples	MAE	<b>0.051190</b>	0.051315	0.052515	0.051676	0.051194	0.052663	0.052508	0.052302
Forward	MSE	0.005224	<b>0.005220</b>	0.005482	0.005296	0.005227	0.005498	0.005456	0.005493
Forecasts	RMSE	0.072281	<b>0.072253</b>	0.074041	0.072770	0.072299	0.074151	0.073865	0.074116
Three	R <sup>2</sup>	0.967213	0.966519	0.961981	0.964960	<b>0.968518</b>	0.959374	0.963573	0.963274
Samples	MAE	0.041521	0.042333	0.044512	0.042729	<b>0.040448</b>	0.046377	0.043806	0.043734
Forward	MSE	0.003226	0.003294	0.003741	0.003447	<b>0.003097</b>	0.003997	0.003584	0.003613
Forecasts	RMSE	0.056802	0.057401	0.061167	0.058721	<b>0.055660</b>	0.063230	0.059872	0.060118
Overall	R <sup>2</sup>	<b>0.931900</b>	0.930401	0.930448	0.929897	0.931440	0.929779	0.929906	0.930161
Forecast	MAE	0.055913	0.056652	0.057383	0.056833	<b>0.055817</b>	0.057783	0.057393	0.056990
Outcomes	MSE	<b>0.006701</b>	0.006848	0.006844	0.006898	0.006746	0.006910	0.006897	0.006872
	RMSE	<b>0.081865</b>	0.082760	0.082732	0.083059	0.082140	0.083130	0.083054	0.082903

**Table 19** Parameters selected by metaheuristics for best-performing wind farm prediction models using VMD

Method	Neurons layer 1	Learning rate	Epochs	Dropout	Layers	Neurons layer 2
VMD-LSTM-ERSA	142	0.009502	600	0.151008	2	127
VMD-LSTM-RSA	140	0.006202	520	0.155388	2	155
VMD-LSTM-PSO	130	0.010000	600	0.200000	2	200
VMD-LSTM-ABC	123	0.008691	529	0.085940	2	152
VMD-LSTM-FA	100	0.010000	600	0.200000	2	100
VMD-LSTM-HHO	200	0.009344	476	0.155241	2	100
VMD-LSTM-WOA	100	0.009352	481	0.094570	2	128
VMD-LSTM-ChOA	144	0.007816	391	0.138489	2	126

**Fig. 22** Wind farm dataset objective function and  $R^2$  distribution plots for each metaheuristic with VMD**Fig. 23** Wind farm dataset objective function and  $R^2$  convergence plots for each metaheuristic with VMD**Fig. 24** Wind farm dataset objective swarm and KDE plots for each metaheuristic with VMD

**Table 20** Wind farm dataset overall metrics for best, worst, mean, and median run using EMD

Method	Best	Worst	Mean	Median	Std	Var
EMD-LSTM-ERSA	<b>0.020351</b>	0.020556	<b>0.020460</b>	0.020556	1.02E-04	1.05E-08
EMD-LSTM-RSA	0.020470	0.020556	0.020513	<b>0.020513</b>	4.31E-05	1.86E-09
EMD-LSTM-PSO	0.020491	0.020674	0.020589	0.020674	9.17E-05	8.41E-09
EMD-LSTM-ABC	0.020828	0.021115	0.020971	0.020971	1.44E-04	2.07E-08
EMD-LSTM-FA	0.020512	<b>0.020549</b>	0.020531	0.020531	<b>1.84E-05</b>	<b>3.37E-10</b>
EMD-LSTM-HHO	0.020842	0.020887	0.020860	0.020842	2.17E-05	4.72E-10
EMD-LSTM-WOA	0.020617	0.020666	0.020640	0.020617	2.41E-05	5.78E-10
EMD-LSTM-ChOA	0.020509	0.020935	0.020722	0.020722	2.13E-04	4.53E-08

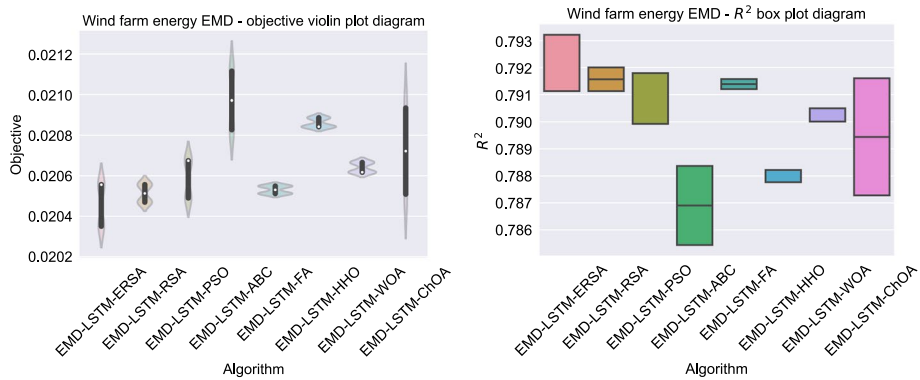
**Table 21** Wind farm dataset detailed metrics for each prediction step of the best run using EMD

	Error indicator	EL-ERSA	EL-RSA	EL-PSO	EL-ABC	EL-FA	EL-HHO	EL-WOA	EL-ChOA
One	R <sup>2</sup>	0.770306	0.770573	0.770635	0.766354	<b>0.770850</b>	0.766029	0.769614	0.767667
Sample	MAE	<b>0.102287</b>	0.102416	0.102454	0.103185	0.102444	0.103351	0.102323	0.102841
Forward	MSE	0.022605	0.022579	0.022573	0.022994	<b>0.022552</b>	0.023026	0.022673	0.022865
Forecasts	RMSE	0.150350	0.150263	0.150242	0.151638	<b>0.150172</b>	0.151743	0.150576	0.151211
Two	R <sup>2</sup>	<b>0.799361</b>	0.798019	0.797192	0.793547	0.796900	0.793721	0.796892	0.797148
Samples	MAE	<b>0.098350</b>	0.098588	0.099168	0.099906	0.098911	0.099839	0.098928	0.099220
Forward	MSE	<b>0.019746</b>	0.019878	0.019959	0.020318	0.019988	0.020301	0.019989	0.019964
Forecasts	RMSE	<b>0.140520</b>	0.140989	0.141277	0.142541	0.141379	0.142481	0.141381	0.141292
Three	R <sup>2</sup>	0.809964	0.807416	0.807551	0.805203	0.806968	0.804903	0.805001	<b>0.809996</b>
Samples	MAE	0.099985	0.099553	0.099394	0.100817	0.099914	0.100610	0.100618	<b>0.099235</b>
Forward	MSE	0.018701	0.018952	0.018939	0.019170	0.018996	0.019199	0.019190	<b>0.018698</b>
Forecasts	RMSE	0.136755	0.137669	0.137620	0.138457	0.137828	0.138564	0.138529	<b>0.136743</b>
Overall	R <sup>2</sup>	<b>0.793210</b>	0.792002	0.791792	0.788367	0.791572	0.788217	0.790502	0.791603
Forecast	MAE	0.100207	<b>0.100185</b>	0.100338	0.101302	0.100422	0.101266	0.100622	0.100431
Outcomes	MSE	<b>0.020350</b>	0.020469	0.020490	0.020827	0.020511	0.020841	0.020616	0.020508
	RMSE	<b>0.142656</b>	0.143072	0.143144	0.144317	0.143220	0.144368	0.143587	0.143209

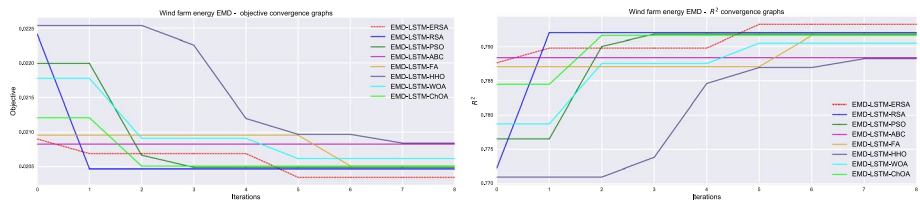
results are coming from the normal distribution. The swarm plot shows the diversity of the solutions during the last iteration of the best-performing run of each algorithm.

## 6.2.2 Wind farm dataset with EMD

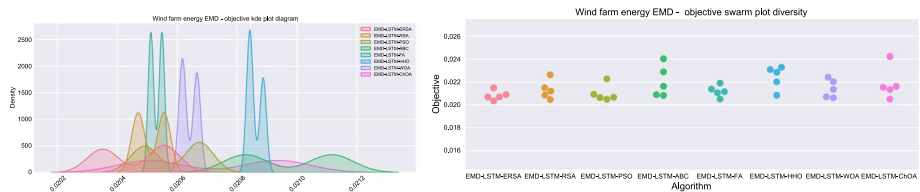
This section presents the results on the wind farm dataset, with employed EMD. Table 20 shows the overall metrics in terms of the best, worst, mean, and median values, accompanied by the standard deviation and variance values over 30 separate runs of each regarded algorithm. EMD-LSTM-ERSA model accomplished the supreme results in terms of the best and mean values. The second-best result was scored by the EMD-LSTM-RSA method, while EMD-LSTM-PSO attained the third-best score. Otherwise, EMD-LSTM-FA obtained the best result for the worst metric, and also for standard deviation and variance. Finally, the best median value was achieved by the EMD-LSTM-RSA method.



**Fig. 25** Wind farm dataset objective function and  $R^2$  distribution plots for each metaheuristic with EMD



**Fig. 26** Wind farm dataset objective function and  $R^2$  convergence plots for each metaheuristic with EMD



**Fig. 27** Wind farm dataset objective swarm and KDE plots for each metaheuristic with EMD

**Table 22** Parameters selected by metaheuristics for best-performing wind farm prediction models using EMD

Method	Neurons layer 1	Learning rate	Epochs	Dropout	Layers	Neurons layer 2
EMD-LSTM-ERSA	100	0.010000	600	0.200000	3	100
EMD-LSTM-RSA	118	0.005365	600	0.121341	2	181
EMD-LSTM-PSO	100	0.004172	600	0.130481	2	100
EMD-LSTM-ABC	115	0.007103	553	0.153556	2	273
EMD-LSTM-FA	100	0.004116	600	0.126646	2	218
EMD-LSTM-HHO	132	0.003038	600	0.107516	2	259
EMD-LSTM-WOA	125	0.003430	600	0.126450	2	112
EMD-LSTM-ChOA	100	0.005069	600	0.161137	3	234

**Table 23** Comparison of the best performing methods with other contemporary prediction models applied to the wind farm dataset

Method	MSE	$R^2$
SVM	0.029070	0.770072
Random forest	0.032550	0.822352
XGBoost	0.023833	0.789320
KELM	0.023340	0.779011
AdaBoost	0.022050	0.775031
ANN (3 layers)	0.027250	0.766072
RNN (3 layers)	0.024250	0.787320
LSTM	0.021339	0.783321
LSTM-ERSA	0.020565	0.791029
VMD-LSTM-ERSA	<b>0.006701</b>	<b>0.931900</b>
EMD-LSTM-ERSA	0.020350	0.793210

Table 21 brings forward the detailed metrics of every prediction step regarding the best run of each algorithm. The prefix EL is used to denote that EMD-LSTM-LSMT is used. It can be noted that the suggested EMD-LSTM-ERSA attained the best scores for overall results, in terms of the objective—MSE, but also for other important indicators, namely  $R^2$  and RMSE (except MAE). Also, EMD-LSTM-ERSA attained the best results for the two-sample forward predictions, for all observed indicators. The best scores for one sample forward were achieved by the EMD-LSTM-FA approach, while the best outcomes for the three samples forward were attained by EMD-LSTM-ChOA. Looking at the overall results, the second-best algorithm was EMD-LSTM-RSA, in front of EMD-LSTM-PSO. The proposed EMD-LSTM-ERSA attained the best overall MSE value of 0.020351, in front of EMD-LSTM-RSA with an MSE value of 0.020556.

The best set of LSTM parameters produced by the top-performing run of each metaheuristic for this scenario is shown in Table 22. The proposed LSTM-ERSA established the LSTM structure as follows: 100 neurons in the initial layer, a learning rate of 0.010000, 600 epochs, a dropout value of 0.200000, three layers, and 100 neurons in the second layer for this particular scenario with employed EMD.

Aiming to provide better insight into the results, visualizations are provided in Figs. 25, 26, and 27. Figure 25 shows the violin plots for the objective function (MSE), accompanied by the box plot of the  $R^2$  indicator, for all 30 runs. After that, the convergence diagrams of the objective function and  $R^2$  for the best run of each algorithm are given in Fig. 26. It can be noted that at the beginning, ChOA, PSO, and RSA exhibited slightly faster convergence, however, all of them were overrun by the proposed ERSa in the final rounds of execution. Finally, the KDE diagram and swarm plot are shown in Fig. 27. KDE diagram is used to show the probability density function and can indicate whether or not the outcomes originate from a normal distribution. The swarm plot shows the diversity of the solutions during the last iteration of the best-performing run of each algorithm. In this scenario, it can be noted that all outcomes of the ERSa were grouped near the best solution at the end of the run.

### 6.2.3 Comparison with other models on the Wind farm dataset

To demonstrate the comparative performance improvements of the coupling optimizers with decomposition's techniques in the introduced methodology the best outcomes of each

**Table 24** Shapiro–Wilk test scores for the single-problem analysis

Problem	ERSA	RSA	PSO	ABC	FA	HHO	WOA	ChOA
Wind	0.024	0.026	0.017	0.021	0.029	0.035	0.031	0.039
Wind VMD	0.029	0.030	0.019	0.011	0.042	0.039	0.027	0.034
Wind EMD	0.014	0.022	0.028	0.018	0.033	0.035	0.039	0.036
Wind farm	0.016	0.024	0.027	0.018	0.023	0.028	0.029	0.033
Wind farm VMD	0.023	0.018	0.014	0.024	0.017	0.031	0.034	0.028
Wind farm EMD	0.021	0.015	0.023	0.019	0.031	0.033	0.031	0.022

**Table 25** Shapiro–Wilk test scores for the multiple problem analysis

Method	ERSA	RSA	PSO	ABC	FA	HHO	WOA	ChOA
<i>p</i> -value	0.0084	0.0093	0.0061	0.0087	0.0074	0.0095	0.0132	0.0147

approach have been compared to several contemporary prediction models (Table 23). The outcomes of the objective (MSE) and indicator  $R^2$  fractions is shown in Table 23.

As it can be observed in Table 23 the VMD demonstrated the best performance when applied alongside metaheuristics optimized LSTM models.

### 6.3 Validation: statistical tests

Modern computer research necessitates scientists to determine if the introduced improvements are statistically significant since experimental outcomes alone are usually inadequate to declare that one algorithm outperforms its competitors. This research manuscript tested eight methods, including the proposed ERSA metaheuristics, for tuning LSTM networks on two wind power generation datasets. The comparison was conducted among eight methods and 6 problem instances utilized for multi-problem analysis as per Eftimov et al. (2016).

Literature recommendations by Eftimov et al. (2016), Derrac et al. (2011) suggest that statistical tests in such scenarios should involve creating a representative collection of outcomes for each method involves creating a sample of outcomes by determining average objective values over several independent executions for each problem. Nevertheless, this methodology may not be ideal when dealing with outliers that originally form non-normal distribution, thus it may result in misleading conclusions. An open question remains regarding whether taking the mean objective function value to use in statistical tests is appropriate for comparison of stochastic methods, as per a literature survey cited by Eftimov et al. (2016). Nonetheless, despite these potential drawbacks, the classification error rate objective function was averaged over 30 independent runs in order to contrast 10 methods across 6 problem instances in this study.

The decision was made after performing the Shapiro–Wilk test (Shapiro and Francia 1972) for single-problem analysis using the described procedure: a data sample was constructed for each algorithm and every problem by gathering the results of each run, and the corresponding *p*-values were computed for all method-problem combinations. The resulting *p*-values are presented in Table 24.

**Table 26** Friedman test scores

Methods	ERSA	RSA	PSO	ABC	FA	HHO	WOA	ChOA
Wind	1	5	7	2	6	8	3	4
Wind VMD	1	3	6	8	4	5	7	2
Wind EMD	1	7	2	8	3	4	6	5
Wind farm	1	8	7	3	6	2	5	4
Wind farm VMD	1	4	3	7	2	8	6	5
Wind farm EMD	1	2	3	7	5	8	6	4
Average ranking	1.00	4.83	4.67	5.83	4.33	5.83	5.50	4.00
Rank	1	5	4	7	3	8	6	2

**Table 27** Friedman aligned test scores

Methods	ERSA	RSA	PSO	ABC	FA	HHO	WOA	ChOA
Wind	14	22	29	18	26	30	20	21
Wind VMD	12	16	28	34	17	27	31	15
Wind EMD	1	43	9	45	19	37	40	38
Wind farm	4	46	39	13	36	7	32	23
Wind farm VMD	3	25	24	42	6	44	41	33
Wind farm EMD	2	5	8	47	11	48	35	10
Average ranking	6.00	26.17	22.83	33.17	19.17	32.17	33.17	23.33
Rank	1	5	3	7	2	6	8	4

Since the  $p$ -values in Table 24 are all below  $\alpha = 0.05$  and therefore not caused by chance, the null hypothesis can be rejected. Consequently, the data samples for all method-problem combinations are not originating from the Gaussian distribution, meaning it is not acceptable to utilize the average objective value in further statistical tests. As a result, this study utilized the best values for further statistical analysis.

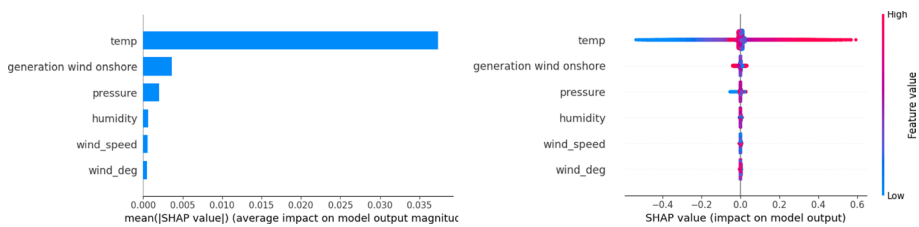
Next, the multi-problems multiple methods statistical analysis has been employed, using data samples made using the best objective function value higher than 30 individual runs for all individual algorithms on every problem instance. To ensure the valid use of parametric tests, we verified the following conditions: independence, normality, and homoscedasticity of the data, as described by LaTorre et al. (2021). Independence was confirmed because each run began with generating a collection of random solutions. To assess normality, the data samples were subjected to the Shapiro–Wilk test, and the subsequent results for each algorithm are presented in Table 25.

Despite conducting the Shapiro–Wilk test for all methods, the resulting  $p$ -values were significantly less than  $\alpha = 0.05$ , as indicated in Table 25. This suggests that the assumption of justified use for parametric tests was not met, and non-parametric tests were employed instead. The suggested ERSA method was set as the control algorithm in all conducted non-parametric tests.

As a result, the Friedman test (1937, 1940) was utilized with a two-way variance analysis with ranking, to determine if the performance level of the proposed ERSA was significantly superior to other contenders. The application of this type of test, accompanied by the Holm post-hoc procedure has been proposed by Derrac et al. (2011). The Friedman test

**Table 28** Holm's step-down procedure

Comparison	p-values	Ranking	$\alpha = 0.05$	$\alpha = 0.1$	$H_1$	$H_2$
ERSA-ABC	0.000315783359126	0	0.007143	0.014286	1	1
ERSA-HHO	0.000315783359126	1	0.008333	0.016667	1	1
ERSA-WOA	0.000731358293341	2	0.010000	0.020000	1	1
ERSA-RSA	0.00335832275131	3	0.012500	0.025000	1	1
ERSA-PSO	0.004760945592049	4	0.016667	0.033333	1	1
ERSA-FA	0.00921106272705	5	0.025000	0.050000	1	1
ERSA-ChOA	0.016947426762345	6	0.050000	0.100000	1	1

**Fig. 28** Wind dataset without decomposition best performing LSTM model feature impact determined through SHAP analysis

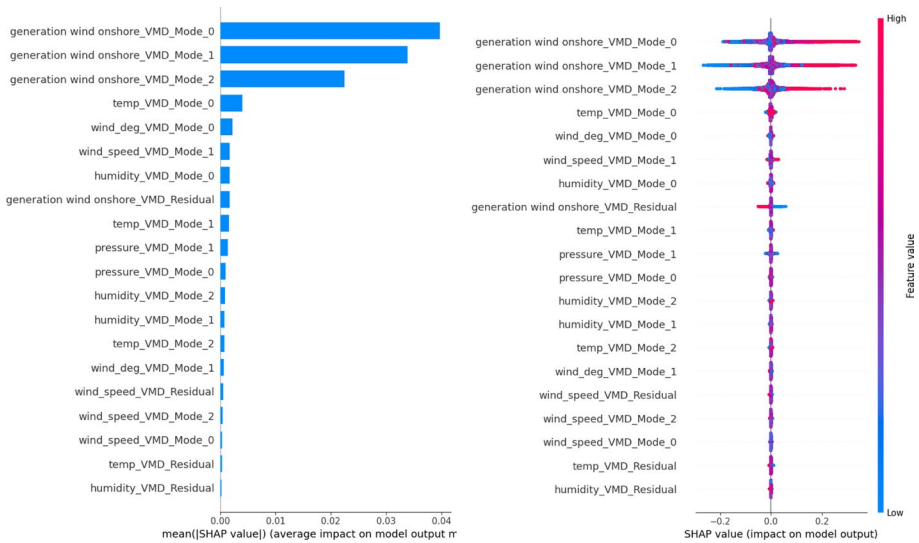
scores are provided in Table 26, accompanied by the scores for the Friedman-aligned test, provided in Table 27.

The results presented in Table 26 indicate that the suggested ERSA method attained a superior level of performance compared to the rest of the methods in the comparative analysis, by scoring the average ranking value of 1.00. In these experiments, the second-best result was obtained by the ChOA algorithm with an average ranking of 4.00, in front of FA in third place with an average ranking of 4.33. The basic implementation of the RSA attained an average ranking of 4.83, showing the obvious superiority of the suggested ERSA over the elementary version of the algorithm. Moreover, the Friedman statistics ( $\chi_r^2 = 17.22$ ) is greater than  $\chi^2$  critical value, with seven degrees of freedom (14.07) with the level of significance  $\alpha = 0.05$ . As the Friedman  $p$ -value is  $1.24 \times 10^{-9}$ , it infers that the results significantly vary with the observed methods. Finally, it allows the rejection of the null hypothesis ( $H_0$ ), and it confirms that the proposed ERSA method attained performance that was significantly statistically different from other contenders. It is possible to draw similar conclusions from the Friedman-aligned test scores, given in Table 27.

Last but not least, as discussed by Sheskin (2020), the Iman and Davenport's test (1980) may provide more detailed results compared to the  $\chi^2$ , and this particular test was executed as well. The Iman and Davenport's obtained score is 3.47, which is greater than the  $F$ -distribution's critical value 2.28, allowing the conclusion that this particular test rejects  $H_0$  as well.

The non-parametric post-hoc Holm's step-down procedure has been employed since both conducted tests reject the null hypothesis, with the findings reported in Table 28. This procedure sorts the regarded methods concerning their  $p$  values evaluated to  $\alpha/(k-i)$ , where  $k$  and  $i$  represent the degree of freedom ( $k = 7$  in this study) and the





**Fig. 29** Wind dataset with VMD best performing LSTM model feature impact determined through SHAP analysis

method's number, after sorting in the ascending order (related to rank). This study utilized  $\alpha$  threshold values of 0.05 and 0.1. The results reported in Table 28 again clearly imply that the introduced ERSA method significantly statistically outscored all contenders for both regarded significance levels 0.05 and 0.1.

## 6.4 Best model interpretation and SHAP analysis

This section brings forward the interpretation of the top-performing model on each of the two regarded datasets. In the previously discussed method, SHAP values were used to estimate feature importance. These values compare the model's predictions with and without each feature, demonstrating the impact of the feature on the observed model's output. Because the order of features can influence predictions, feature importance is calculated in every possible order to ensure fair feature comparisons (García and Aznarte 2020). This study developed separate models to assess wind energy generation, and the importance and impact of the features were evaluated for each model. Specifically, the analysis focused on how each predictor variable affected the predicted probability of observation.

### 6.4.1 Wind dataset SHAP analysis

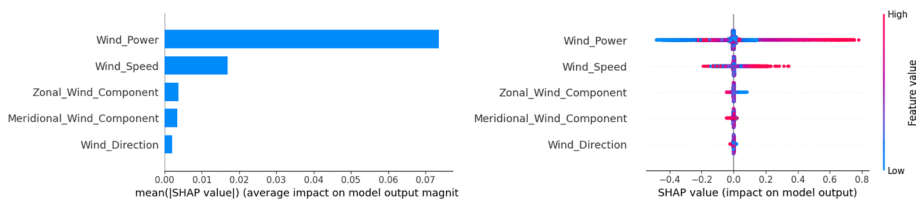
Figure 28 shows the impacts of each feature for the LSTM-ERSA model on the wind dataset, while Fig. 29 presents the impacts of the best-performing model with VMD, namely VMD-LSTM-ERSA. A closer look at the waterfall plots (left part of the figure) indicates that the most important feature in this scenario is temperature, followed by the generation of wind onshore, and pressure.

One interesting observation is that a significant shift in feature influence occurs following the application of VMD. The top-performing model trained without decomposition

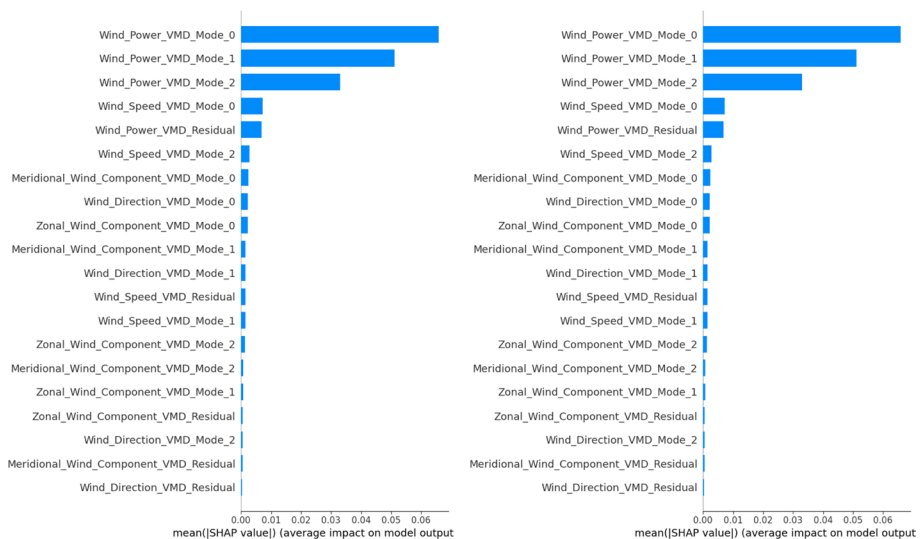
places the highest value on it. However, following the application of VMD, the first three decomposed modes of onshore wind generation become the most significant features. This is likely due to the noise associated with onshore wind generation data. In the raw form, onshore wind generation is quite a complex and noisy data sequence, while the temperature is a smoother more predictable data sequence less prone to sudden shifts. Following decomposition, the onshore wind generation modes are made more reliable, predictable, and less noisy allowing these features to have a more significant contribution to the model, even higher relative to temperature. It is also interesting to note that the residual values for each mode, data components that could not be decomposed to a specific mode mostly containing noise, have the lowest influence on model predictions.

### 6.4.2 Wind farm dataset SHAP analysis

In Fig. 30 the impact of each observed feature for the best-performing LSTM-ERSA model can be seen for the wind farm dataset. Following this Fig. 31 likewise demonstrates feature



**Fig. 30** Wind farm dataset without decomposition best performing LSTM model feature impact determined through SHAP analysis



**Fig. 31** Wind farm dataset with VMD best performing LSTM model feature impact determined through SHAP analysis

impacts following data decomposition using VMD with the VMD-LSTM-ERSA model. Further details can be observed in the accompanying waterfall diagrams on each figure.

The findings of the SHAP analysis indicate a strong influence of the wind power component, followed closely by wind speed. These findings are further reinforced by the analysis of models working with decomposed data. Wind power modes, followed by wind speed modes show a significant influence. It can also be noted that the influence of residual components, consisting mostly of noise is very low for most features. Nevertheless, the wind power residual does show a decent influence on the prediction.

## 7 Conclusion

This study employed and tuned the LSTM ML model aimed at optimizing the prediction of the power production that comes from renewable sources. In the beginning, an improved variant of the swarm intelligence RSA metaheuristics was proposed, that surpass the known deficiencies of the initially introduced algorithm. Later, the introduced algorithm, named ERSA, was used to adjust the hyperparameters of the LSTM network for wind energy production problems.

The introduced methodology was assessed on two wind production datasets, and in three scenarios for each dataset - without decomposition, with VMD, and with EMD employed. The predictions were executed up to three samples forward, and the outcomes were contrasted against those attained by competing metaheuristics applied in identical experimental setups. The obtained results clearly indicate the superiority of the proposed LSTM-ERSA model in all regarded scenarios. Statistical analysis was also employed, concluding that the results attained by the proposed method are statistically significant. Additionally, the influence of proper parameter selection is clearly demonstrated. The performance of optimized networks is significantly improved. Finally, SHAP analysis was performed on the best-performing model on each dataset, aiming to assess the impact of the features on each model.

The conducted research has shown a great deal of potential for using hybrid ML models tuned by metaheuristics algorithms for wind power production prognosis. A crucial task such as the estimation of the expected production by the wind farm is important as the power grid has to be capable of balancing power production and consumption at all times. Future research regarding this important topic will explore developing even more accurate models optimized by various metaheuristics algorithms. Emerging decomposition algorithms and their optimization will be explored. Additionally, the introduced modified metaheuristic will be applied to emerging challenges.

**Acknowledgements** This research was supported by the Science Fund of the Republic of Serbia, Grant No. 7502, Intelligent Multi-Agent Control and Optimization applied to Green Buildings and Environmental Monitoring Drone Swarms - ECOSwarm.

**Author Contributions** MP-K: conceptualization, writing - original draft. LJ: software, validation, data curation, visualization. NB: formal analysis, investigation, data curation, supervision, writing - review & editing. MD: writing - review & editing, project administration, validation, supervision. MZ: investigation, resources, writing - original draft, formal analysis. MT: project administration, validation, supervision. IS: investigation, data curation, visualization, statistical analysis. WP: validation, supervision, project administration.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with animals performed by any of the authors.

**Informed consent** Informed consent was obtained from all individual participants included in the study.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abraham A, Guo H, Liu H (2006) Swarm intelligence: foundations, perspectives and applications. In: *Swarm intelligent systems*. Springer, pp 3–25
- Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng* 376:113609
- Abualigah L, Abd Elaziz M, Sumari P, Geem ZW, Gandomi AH (2022) Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer. *Expert Syst Appl* 191:116158. <https://doi.org/10.1016/j.eswa.2021.116158>
- Akella A, Saini R, Sharma MP (2009) Social, economical and environmental impacts of renewable energy systems. *Renew Energy* 34(2):390–396
- Amalou I, Mouhni N, Abdali A (2022) Multivariate time series prediction by RNN architectures for energy consumption forecasting. *Energy Rep* 8:1084–1091. <https://doi.org/10.1016/j.egy.2022.07.139>
- Awerbuch S, Berger M (2003) Applying portfolio theory to EU electricity planning and policy-making, Sweden. <https://www.osti.gov/etdeweb/biblio/20354690>
- Bacanin N, Bezdan T, Tuba E, Strumberger I, Tuba M, Zivkovic M (2019) Task scheduling in cloud computing environment by grey wolf optimizer. In: 2019 27th telecommunications forum (TEL-FOR). IEEE, pp 1–4
- Bacanin N, Sarac M, Budimirovic N, Zivkovic M, AlZubi AA, Bashir AK (2022a) Smart wireless health care system using graph LSTM pollution prediction and dragonfly node localization. *Sustain Comput Inform Syst* 35:100711
- Bacanin N, Zivkovic M, Stoean C, Antonijevic M, Janicijevic S, Sarac M, Strumberger I (2022b) Application of natural language processing and machine learning boosted with swarm intelligence for spam email filtering. *Mathematics* 10(22):4173
- Bacanin N, Zivkovic M, Al-Turjman F, Venkatachalam K, Trojovský P, Strumberger I, Bezdan T (2022c) Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application. *Sci Rep* 12(1):1–20
- Bacanin N, Stoean C, Zivkovic M, Jovanovic D, Antonijevic M, Mladenovic D (2022d) Multi-swarm algorithm for extreme learning machine optimization. *Sensors* 22(11):4204
- Bacanin N, Venkatachalam K, Bezdan T, Zivkovic M, Abouhawwash M (2023a) A novel firefly algorithm approach for efficient feature selection with covid-19 dataset. *Microprocess Microsyst* 98:104778
- Bacanin N, Stoean C, Zivkovic M, Rakic M, Strulak-Wójcikiewicz R, Stoean R (2023b) On the benefits of using metaheuristics in the hyperparameter tuning of deep learning models for energy load forecasting. *Energies* 16(3):1434
- Belotti J, Siqueira H, Araujo L, Stevan SL Jr, Mattos Neto PS, Marinho MH, Oliveira JFL, Usberti F, Leone Filho MdA, Converti A et al (2020) Neural-based ensembles and unorganized machines to predict streamflow series from hydroelectric plants. *Energies* 13(18):4769

- Beni G (2020) Swarm intelligence. In: Complex social and behavioral systems: game theory and agent-based models. Springer, pp 791–818
- Bezdan T, Zivkovic M, Tuba E, Strumberger I, Bacanin N, Tuba M (2020a) Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. In: International conference on intelligent and fuzzy systems. Springer, pp 718–725
- Bezdan T, Zivkovic M, Antonijevic M, Zivkovic T, Bacanin N (2020b) Enhanced flower pollination algorithm for task scheduling in cloud computing environment. In: Machine learning for predictive analysis. Springer, pp 163–171
- Bezdan T, Zivkovic M, Tuba E, Strumberger I, Bacanin N, Tuba M (2020c) Glioma brain tumor grade classification from MRI using convolutional neural networks designed by modified FA. In: International conference on intelligent and fuzzy systems. Springer, pp 955–963
- Bezdan T, Cvetnic D, Gajic L, Zivkovic M, Strumberger I, Bacanin N (2021) Feature selection by firefly algorithm with improved initialization strategy. In: 7th conference on the engineering of computer based systems. pp 1–8
- Bukumira M, Antonijevic M, Jovanovic D, Zivkovic M, Mladenovic D, Kunjadic G (2022) Carrot grading system using computer vision feature parameters and a cascaded graph convolutional neural network. *J Electron Imaging* 31(6):061815
- Cheng S, Shi Y (2011) Diversity control in particle swarm optimization. In: 2011 IEEE symposium on swarm intelligence. IEEE, pp 1–9
- Coppitters D, Contino F (2023) Optimizing upside variability and antifragility in renewable energy system design. *Sci Rep* 13(1):9138
- Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
- Devi AS, Maragatham G, Boopathi K, Rangaraj A (2020) Hourly day-ahead wind power forecasting with the EEMD-CSO-LSTM-EFG deep learning technique. *Soft Comput* 24(16):12391–12411. <https://doi.org/10.1007/s00500-020-04680-7>
- Dinçer H, Yüksel S, Eti S (2023) Identifying the right policies for increasing the efficiency of the renewable energy transition with a novel fuzzy decision-making model. *J Soft Comput Decis Anal* 1(1):50–62. <https://doi.org/10.31181/jscda1120234>
- Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1(4):28–39
- Dragomiretskiy K, Zosso D (2013) Variational mode decomposition. *IEEE Trans Signal Process* 62(3):531–544
- Dwivedi R, Dave D, Naik H, Singhal S, Omer R, Patel P, Qian B, Wen Z, Shah T, Morgan G et al (2023) Explainable AI (XAI): core ideas, techniques, and solutions. *ACM Comput Surv* 55(9):1–33
- Eftimov T, Korošec P, Seljak BK (2016) Disadvantages of statistical comparison of stochastic optimization algorithms. In: Proceedings of the bioinspired optimization methods and their applications, BIOMA. pp 105–118
- Emmerich M, Shir OM, Wang H (2018) Evolution strategies. In: Handbook of heuristics. Springer, pp 89–119
- Fan H, Jiang M, Xu L, Zhu H, Cheng J, Jiang J (2020) Comparison of long short term memory networks and the hydrological model in runoff simulation. *Water* 12(1):175. <https://doi.org/10.3390/w12010175>
- Fausto F, Reyna-Orta A, Cuevas E, Andrade ÁG, Perez-Cisneros M (2020) From ants to whales: metaheuristics for all tastes. *Artif Intell Rev* 53(1):753–810
- Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J Am Stat Assoc* 32(200):675–701
- Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. *Ann Math Stat* 11(1):86–92
- Fu J, Chu J, Guo P, Chen Z (2019) Condition monitoring of wind turbine gearbox bearing based on deep learning model. *IEEE Access* 7:57078–57087. <https://doi.org/10.1109/ACCESS.2019.2912621>
- Gajic L, Cvetnic D, Zivkovic M, Bezdan T, Bacanin N, Milosevic S (2021) Multi-layer perceptron training using hybridized bat algorithm. In: Computational vision and bio-inspired computing. Springer, pp 689–705
- García MV, Aznarte JL (2020) Shapley additive explanations for NO<sub>2</sub> forecasting. *Eco Inform* 56:101039
- Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 97:849–872
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Hong T, Pinson P, Fan S (2014) Global energy forecasting competition 2012. *Int J Forecast* 30(2):357–363. <https://doi.org/10.1016/j.ijforecast.2013.07.001>

- Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, Yen N-C, Tung CC, Liu HH (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc R Soc Lond Ser A Math Phys Eng Sci* 454(1971):903–995
- Iman RL, Davenport JM (1980) Approximations of the critical region of the fbietkan statistic. *Commun Stat Theory Methods* 9(6):571–595
- Jovanovic L, Jovanovic D, Bacanin N, Jovancai Stakic A, Antonijevic M, Magd H, Thirumalaisamy R, Zivkovic M (2022a) Multi-step crude oil price prediction based on LSTM approach tuned by salp swarm algorithm with disputation operator. *Sustainability* 14(21):14616
- Jovanovic D, Antonijevic M, Stankovic M, Zivkovic M, Tanaskovic M, Bacanin N (2022b) Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. *Mathematics* 10(13):2272
- Jovanovic L, Jovanovic G, Perisic M, Alimpic F, Stanisic S, Bacanin N, Zivkovic M, Stojic A (2023a) The explainable potential of coupling metaheuristics-optimized-XGBoost and SHAP in revealing VOCS' environmental fate. *Atmosphere* 14(1):109
- Jovanovic L, Djuric M, Zivkovic M, Jovanovic D, Strumberger I, Antonijevic M, Budimirovic N, Bacanin N (2023b) Tuning XGBoost by planet optimization algorithm: an application for diabetes classification. In: *Proceedings of fourth international conference on communication, computing and electronics systems: ICCCES 2022*. Springer, pp 787–803
- Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. *Appl Soft Comput* 8(1):687–697
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95-international conference on neural networks*, vol 4. IEEE, pp 1942–1948
- Khishe M, Mosavi MR (2020) Chimp optimization algorithm. *Expert Syst Appl* 149:113338
- LaTorre A, Molina D, Osaba E, Poyatos J, Del Ser J, Herrera F (2021) A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. *Swarm Evol Comput* 67:100973
- Li M, Yao J, Shen Y, Yuan B, Simmonds I, Liu Y (2023) Impact of synoptic circulation patterns on renewable energy-related variables over China. *Renew Energy*. <https://doi.org/10.1016/j.renene.2023.05.133>
- Liu Y, Guan L, Hou C, Han H, Liu Z, Sun Y, Zheng M (2019) Wind power short-term prediction based on LSTM and discrete wavelet transform. *Appl Sci* 9(6):1108
- Liu B, Zhao S, Yu X, Zhang L, Wang Q (2020) A novel deep learning approach for wind power forecasting based on WD-LSTM model. *Energies* 13(18):4964. <https://doi.org/10.3390/en13184964>
- Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) *Advances in neural information processing systems*, vol 30. Curran Associates Inc., New York, pp 4765–4774
- Mattos Neto PS, Oliveira JF, de O. Santos Júnior DS, Siqueira HV, Marinho MH, Madeiro F (2021) An adaptive hybrid system using deep learning for wind speed forecasting. *Inf Sci* 581:495–514
- Milosevic S, Bezdan T, Zivkovic M, Bacanin N, Strumberger I, Tuba M (2021) Feed-forward neural network training by hybrid bat algorithm. In: *Modelling and development of intelligent systems: 7th international conference, MDIS 2020, Sibiu, Romania, October 22–24, 2020, revised selected papers 7*. Springer International Publishing, pp 52–66
- Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Petrovic A, Bacanin N, Zivkovic M, Marjanovic M, Antonijevic M, Strumberger I (2022) The AdaBoost approach tuned by firefly metaheuristics for fraud detection. In: *2022 IEEE world conference on applied intelligence and computing (AIC)*. IEEE, pp 834–839
- Preuss M, Stoean C, Stoean R (2011) Niching foundations: basin identification on fixed-property generated landscapes. In: *Proceedings of the 13th annual conference on genetic and evolutionary computation. GECCO '11*. Association for Computing Machinery, New York, pp 837–844. <https://doi.org/10.1145/2001576.2001691>
- Probst P, Boulesteix A-L, Bischl B (2019) Tunability: importance of hyperparameters of machine learning algorithms. *J Mach Learn Res* 20(1):1934–1965
- Rahnamayan S, Tizhoosh HR, Salama MMA (2007) Quasi-oppositional differential evolution. In: *2007 IEEE congress on evolutionary computation*. pp 2229–2236. <https://doi.org/10.1109/CEC.2007.4424748>
- Razmjoo A, Kaigutha LG, Rad MV, Marzband M, Davarpanah A, Denai M (2021) A technical analysis investigating energy sustainability utilizing reliable renewable energy sources to reduce CO<sub>2</sub> emissions in a high potential area. *Renew Energy* 164:46–57
- Rehman N, Aftab H (2019) Multivariate variational mode decomposition. *IEEE Trans Signal Process* 67(23):6039–6052

- Rehman N, Mandic DP (2010) Multivariate empirical mode decomposition. *Proc R Soc A Math Phys Eng Sci* 466(2117):1291–1302
- Salb M, Jovanovic L, Zivkovic M, Tuba E, Elsadai A, Bacanin N (2022) Training logistic regression model by enhanced moth flame optimizer for spam email classification. In: *Computer networks and inventive communication technologies: proceedings of fifth ICCNCT 2022*. Springer, pp 753–768
- Shahid F, Zameer A, Muneeb M (2021) A novel genetic LSTM model for wind power forecast. *Energy* 223:120069
- Shao B, Song D, Bian G, Zhao Y (2021) Wind speed forecast based on the LSTM neural network optimized by the firework algorithm. *Adv Mater Sci Eng* 2021:1–13
- Shapiro SS, Francia R (1972) An approximate analysis of variance test for normality. *J Am Stat Assoc* 67(337):215–216
- Sheskin DJ (2020) *Handbook of parametric and nonparametric statistical procedures*. CRC Press, Boca Raton
- Shi J, Guo J, Zheng S (2012) Evaluation of hybrid forecasting approaches for wind speed and power generation time series. *Renew Sustain Energy Rev* 16(5):3471–3480
- Stankovic M, Antonijevic M, Bacanin N, Zivkovic M, Tanaskovic M, Jovanovic D (2022) Feature selection by hybrid artificial bee colony algorithm for intrusion detection. In: *2022 international conference on edge computing and applications (ICECAA)*. IEEE, pp 500–505
- Stegherr H, Heider M, Hähner J (2020) Classifying metaheuristics: towards a unified multi-level classification system. *Nat Comput*. <https://doi.org/10.1007/s11047-020-09824-0>
- Stoean C, Zivkovic M, Bozovic A, Bacanin N, Strulak-Wójcikiewicz R, Antonijevic M, Stoean R (2023) Metaheuristic-based hyperparameter tuning for recurrent deep learning: application to the prediction of solar energy generation. *Axioms* 12(3):266
- Strumberger I, Tuba E, Zivkovic M, Bacanin N, Beko M, Tuba M (2019) Dynamic search tree growth algorithm for global optimization. In: *Doctoral conference on computing, electrical and industrial systems*. Springer, pp 143–153
- Tayebi M, El Kafhali S (2022) Performance analysis of metaheuristics based hyperparameters optimization for fraud transactions detection. *Evol Intell*. <https://doi.org/10.1007/s12065-022-00764-5>
- Wang N, Li Z (2023) Short term power load forecasting based on BES-VMD and CNN-Bi-LSTM method with error correction. *Front Energy Res*. <https://doi.org/10.3389/fenrg.2022.1076529>
- Wang L, Liu H, Pan Z, Fan D, Zhou C, Wang Z (2022a) Long short-term memory neural network with transfer learning and ensemble learning for remaining useful life prediction. *Sensors* 22(15):5744. <https://doi.org/10.3390/s22155744>
- Wang D, Cui X, Niu D (2022b) Wind power forecasting based on LSTM improved by EMD-PCA-RF. *Sustainability* 14(12):7307. <https://doi.org/10.3390/su14127307>
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
- Yang X-S (2009) Firefly algorithms for multimodal optimization. In: *International symposium on stochastic algorithms*. Springer, pp 169–178
- Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer, pp 65–74
- Yang X-S, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. *Eng Comput* 29(5):464–83
- Yüksel S, Eti S, Dinçer H, Gökalp Y (2024) Comprehensive risk analysis and decision-making model for hydroelectricity energy investments. *J Soft Comput Decis Anal* 2(1):28–38. <https://doi.org/10.31181/jsdda.2120242>
- Zhang Y, Liu K, Qin L, An X (2016) Deterministic and probabilistic interval prediction for short-term wind power generation based on variational mode decomposition and machine learning methods. *Energy Convers Manag* 112:208–219
- Zhang T, Tang Z, Wu J, Du X, Chen K (2021) Multi-step-ahead crude oil price forecasting based on two-layer decomposition technique and extreme learning machine optimized by the particle swarm optimization algorithm. *Energy* 229:120797. <https://doi.org/10.1016/j.energy.2021.120797>
- Zivkovic M, Bacanin N, Tuba E, Strumberger I, Bezdan T, Tuba M (2020) Wireless sensor networks life time optimization based on the improved firefly algorithm. In: *2020 international wireless communications and mobile computing (IWCMC)*. IEEE, pp 1176–1181
- Zivkovic M, Bacanin N, Venkatachalam K, Nayyar A, Djordjevic A, Strumberger I, Al-Turjman F (2021a) Covid-19 cases prediction by using hybrid machine learning and beetle antennae search approach. *Sustain Cities Soc* 66:102669
- Zivkovic M, Venkatachalam K, Bacanin N, Djordjevic A, Antonijevic M, Strumberger I, Rashid TA (2021b) Hybrid genetic algorithm and machine learning method for covid-19 cases prediction. In: *Proceedings of international conference on sustainable expert systems: ICSES 2020, vol 176*. Springer Nature, p 169

- Zivkovic M, Bezdan T, Strumberger I, Bacanin N, Venkatachalam K (2021c) Improved Harris Hawks optimization algorithm for workflow scheduling challenge in cloud-edge environment. In: *Computer networks, Big Data and IoT*. Springer, pp 87–102
- Zivkovic M, Zivkovic T, Venkatachalam K, Bacanin N (2021d) Enhanced dragonfly algorithm adapted for wireless sensor network lifetime optimization. In: *Data intelligence and cognitive informatics*. Springer, pp 803–817
- Zivkovic M, Bacanin N, Antonijevic M, Nikolic B, Kvascev G, Marjanovic M, Savanovic N (2022) Hybrid CNN and XGBoost model tuned by modified arithmetic optimization algorithm for COVID-19 early diagnostics from X-ray images. *Electronics* 11(22):3798
- Züttel A, Gallandat N, Dyson PJ, Schlapbach L, Gilgen PW, Orimo S-I (2022) Future Swiss energy economy: the challenge of storing renewable energy. *Front Energy Res* 9:785908

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

**Marijana Pavlov-Kagadejev<sup>1</sup> · Luka Jovanovic<sup>2</sup> · Nebojsa Bacanin<sup>3</sup> · Muhammet Deveci<sup>4,5,6</sup> · Miodrag Zivkovic<sup>3</sup> · Milan Tuba<sup>3</sup> · Ivana Strumberger<sup>3</sup> · Witold Pedrycz<sup>7,8,9</sup>**

✉ Nebojsa Bacanin  
nbacanin@singidunum.ac.rs

Marijana Pavlov-Kagadejev  
marijana.pavlov@irmbor.co.rs

Luka Jovanovic  
luka.jovanovic.191@singimail.rs

Muhammet Deveci  
muhammetdeveci@gmail.com

Miodrag Zivkovic  
mzivkovic@singidunum.ac.rs

Milan Tuba  
tuba@ieee.org

Ivana Strumberger  
istrumberger@singidunum.ac.rs

Witold Pedrycz  
wpedrycz@ualberta.ca

<sup>1</sup> Mining and Metallurgy Institute Bor, Zeleni bulevar 35, Bor 19210, Serbia

<sup>2</sup> Technical Faculty, Singidunum University, Danijelova 32, Belgrade 11000, Serbia

<sup>3</sup> Faculty of Informatics and Computing, Singidunum University, Danijelova 32, Belgrade 11000, Serbia

<sup>4</sup> Department of Industrial Engineering, Turkish Naval Academy, National Defence University, 34942, Tuzla, Istanbul, Turkey

<sup>5</sup> The Bartlett School of Sustainable Construction, University College London, London WC1E 6BT, UK

<sup>6</sup> Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon



- <sup>7</sup> Department of Electrical and Computer Engineering, Faculty of Engineering, University of Alberta, 9211 116, Street NW, Edmonton, AB T6G 1H9, Canada
- <sup>8</sup> The Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
- <sup>9</sup> The Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Istinye University, Sariyer/Istanbul, Turkey