

## Article

# A Wavelet PM2.5 Prediction System Using Optimized Kernel Extreme Learning with Boruta-XGBoost Feature Selection

Ali Asghar Heidari <sup>1</sup>, Mehdi Akhoondzadeh <sup>2,\*</sup> and Huiling Chen <sup>3,\*</sup>

<sup>1</sup> School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1439957131, Iran

<sup>2</sup> Photogrammetry and Remote Sensing Department, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, North Amirabad Ave., Tehran 1439957131, Iran

<sup>3</sup> Department of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

\* Correspondence: makhonz@ut.ac.ir (M.A.); chenhuiling\_jsj@wzu.edu.cn (H.C.)

**Abstract:** The fine particulate matter (PM2.5) concentration has been a vital source of info and an essential indicator for measuring and studying the concentration of other air pollutants. It is crucial to realize more accurate predictions of PM2.5 and establish a high-accuracy PM2.5 prediction model due to their social impacts and cross-field applications in geospatial engineering. To further boost the accuracy of PM2.5 prediction results, this paper proposes a new wavelet PM2.5 prediction system (called WD-OSMSSA-KELM model) based on a new, improved variant of the salp swarm algorithm (OSMSSA), kernel extreme learning machine (KELM), wavelet decomposition, and Boruta-XGBoost (B-XGB) feature selection. First, we applied the B-XGB feature selection to realize the best features for predicting hourly PM2.5 concentrations. Then, we applied the wavelet decomposition (WD) algorithm to reach the multi-scale decomposition results and single-branch reconstruction of PM2.5 concentrations to mitigate the prediction error produced by time series data. In the next stage, we optimized the parameters of the KELM model under each reconstructed component. An improved version of the SSA is proposed to reach higher performance for the basic SSA optimizer and avoid local stagnation problems. In this work, we propose new operators based on oppositional-based learning and simplex-based search to mitigate the core problems of the conventional SSA. In addition, we utilized a time-varying parameter instead of the main parameter of the SSA. To further boost the exploration trends of SSA, we propose using the random leaders to guide the swarm towards new regions of the feature space based on a conditional structure. After optimizing the model, the optimized model was utilized to predict the PM2.5 concentrations, and different error metrics were applied to evaluate the model's performance and accuracy. The proposed model was evaluated based on an hourly database, six air pollutants, and six meteorological features collected from the Beijing Municipal Environmental Monitoring Center. The experimental results show that the proposed WD-OLMSSA-KELM model can predict the PM2.5 concentration with superior performance ( $R: 0.995$ ,  $RMSE: 11.906$ ,  $MdAE: 2.424$ ,  $MAPE: 9.768$ ,  $KGE: 0.963$ ,  $R^2: 0.990$ ) compared to the WD-CatBoost, WD-LightGBM, WD-Xgboost, and WD-Ridge methods.



**Citation:** Heidari, A.A.; Akhoondzadeh, M.; Chen, H.A. Wavelet PM2.5 Prediction System Using Optimized Kernel Extreme Learning with Boruta-XGBoost Feature Selection. *Mathematics* **2022**, *10*, 3566. <https://doi.org/10.3390/math10193566>

Academic Editor: Amir Mosavi

Received: 4 September 2022

Accepted: 24 September 2022

Published: 29 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**MSC:** 68T20

## 1. Introduction

With the increased frequency of pollution in recent years and new concerns about mega-cities in developing countries, fine particulate matter (FPM) has received much interest from artificial intelligence scientists [1]. The types of environmental pollutants in China have shifted dramatically as industrialization has progressed, the economy has

grown, and the manufacturing method has shifted [2]. PM2.5 with an equivalent diameter of less than or equal to  $2.5\text{ }\mu\text{m}$  can enter the atmosphere for an extended period [3]. The greater the amount of PM2.5 in the atmosphere, the greater the pollution. Additionally, when opposed to heavier ambient air particulate matter, PM2.5 seems to have a narrow size distribution and higher density and, thus, is easily associated with hazardous and destructive compounds (e.g., toxic elements, germs) [4]. Moreover, PM2.5 has a considerable residence period in the air, which can significantly influence human health and environmental conditions [5].

PM2.5 concentration prediction problems are multi-parameter, complicated, nonlinear procedures, and the efficacy of the linear models is challenging to match with the expectations of the decision-makers when dealing with such nonlinear features [6,7]. Many novel approaches for predicting PM2.5 concentrations have been presented in the past few years [8]. A review of the recent developments in computational models for predicting the PM2.5 series was presented in [9]. Generally, the PM2.5 prediction ideas are classified into four mother types: (1) deterministic algorithms, (2) statistical approaches, (3) artificial intelligence frameworks, and (4) mixed models [10]. The emission, accumulation, dissemination, and transmission of air pollutants are all designed to be simulated by a deterministic model [11]. Its complex meteorological variables and chemical reaction process possess tangible advantages. For example, the authors released a new post-processing technique for outdoor PM2.5 prediction, and their idea was applied to the CMAQ models in [12]. However, our research in this paper primarily utilizes the historical time series info obtained from the prior observations. This technique is highly complicated and expensive and has substantial uncertainty. The statistical model appears simple and effective [13]; however, its system performance is strongly influenced by the ability of linear mapping in the nonlinear procedure [14]. Only information that is linear or nearly linear may be accurately estimated. The actual PM2.5 series, alternatively, appears nonlinear and temperamental [15].

Many researchers have utilized artificial intelligence (AI) paradigms to overcome this disadvantage [16]. For example, in the work by Banga et al. [17], the performance of the extra tree, decision tree, XGBoost, random forest, Light GBM, and AdaBoost regression models was compared for predicting PM2.5 in five cities in China. The artificial intelligence algorithms can handle complicated nonlinear relationships between the involved pollutants and meteorological features and significantly boost the PM2.5 prediction accuracy [18]. Some of the effective models are ELM and KELM, which have been validated in many prediction fields; for instance, they were optimized by the biogeography-based optimizer (BBO) and BBO-KELM utilized by Li and Li [19] to estimate ultra-short-term wind speed in various places. In another research work, cuckoo-search-based ELM was trained for PM10 data of Beijing and Harbin in China, and the results confirmed that the optimization method has a positive impact on the performance of the ELM technique [20].

Since the PM2.5 series has always been a nonlinear dynamic model with nonlinearity, non-stationarity, and complexity [21], a single prediction system cannot reliably estimate the PM2.5 concentration. However, the notion of “decomposition and integration” in the regression method overcomes this drawback by combining the benefits of data decomposition, swarm-based optimization algorithms, machine learning models, and feature selection. It decreases the system’s nonlinear and non-stationarity traits, significantly increases the prediction accuracy, and helps decision-makers obtain more high-quality, optimal solutions [22]. Yang et al. [23] developed a hybrid method based on feature analysis, secondary decomposition, and optimized ELM with the chimp optimizer for a case study on hourly data of Shanghai and Shenyang, China. Furthermore, another research work by Li et al. [10] used crow-search-based KELM hybridized with differential symbolic entropy (DSE) and variational mode decomposition improved by butterfly optimization (BVMD), the results of which were verified on PM2.5 data in Beijing, Shenyang, and Shanghai from 1 January 2016 to 31 March 2021. Liu et al. [24] presented an innovative hybrid system for four towns in China called WPD-PSO-BP-Adaboost, based on wavelet packet decomposition (WPD), the particle swarm algorithm (PSO) algorithm, the back propagation

neural network (BPNN), and the Adaboost model. Sun and Xu [25] proposed a new hybrid hourly PM2.5 prediction framework, called RF-GSA-TVFEMD-SE-MFO-ELM, in which they decomposed the series by time-varying filtering-based empirical mode decomposition (TVFEMD) and then utilized the optimized ELM by moth flame optimization (MFO) for the prediction of hourly PM2.5 series of four cities in the Beijing Tianjin Hebei region. In another paper by Yin et al. [26], the authors proposed two boosting approaches, adapted AdaBoost.RT and gradient boosting (GB), to improve the ELM for ensemble prediction of the PM2.5. A simple salp-swarm-based ELM was proposed by Liu and Ye [27] for the PM2.5 data of Hangzhou from 2016 to 2020. In another research work, a multi-objective Harris hawks optimizer (HHO) was integrated with ELM to predict the PM2.5 of three cities in China [28]. Another HHO-based ELM model was developed for PM2.5 datasets from Beijing, Tianjin, and Shijiazhuang in China [29]. A similar work developed an ensemble pigeon-inspired ELM with the multidimensional scaling and K-means clustering component for air quality prediction [30]. A WAV-VMD-KELM model consisting of wavelet denoising, variational mode decomposition (VMD) of the data, and KELM as the regressor was developed by Xing et al. [31] for predicting hourly PM2.5 series in Xian, China. A group teaching optimized ELM with the wavelet transform (WT) and ICEEMDAN was proposed for PM2.5 data by Jiang et al. [32]. In many hybrid models, in addition to the optimization core, the models have been integrated with decomposition methods, including empirical mode decomposition (EMD), variational mode decomposition (VMD), wavelet decomposition (WD), and secondary decomposition (SD). The data decomposition methodology can decompose air quality signals into a predetermined number of sub-sequences, and its use dramatically enhances hybrid models' prediction capabilities. Each of these methods has its benefits and weaknesses, while the WD is one of the most effective methods in the literature. A comprehensive review of the multi-scale decomposition strategies was presented by Liu et al. [33].

Although China has gained traction in PM2.5 management in context, reducing emissions has proven to be a complex problem. As a motivation of this research, the precise prediction of PM2.5 concentrations with an efficient model is critical for public health protection and developing preventative strategies. Such efficient models can be utilized within integrated information systems to help decision-makers act autonomously. However, there is a significant gap in research about this problem, and although previous prediction models have their advantages, various problems still need to be addressed. Because of their excellent learning capacity and capability to deal with nonlinear data, AI-based approaches have been frequently employed; yet, such systems are prone to being trapped in local optima and generalization error. The integrated optimization methods are swarm-based and basic versions, and the imbalance of exploration and exploitation may result in being trapped in local optima and poor regression accuracy. Moreover, the learning models' performance depends on the optimized set of hyper-parameters, which highly affect the regression accuracy. According to the no free lunch (NFL) theorem, no optimization or machine learning model or hybrid version can outperform all possible models on a specific set of problems [34]. Therefore, there is room for developing more efficient hybrid models for specific PM2.5 datasets. In addition, hybrid models still cannot show the best performance using only one regression method with a basic optimizer since the model with premature regression performance cannot recognize various patterns in the set of features. Hence, there is a need to pre-process the input data more effectively and optimize the model's performance more efficiently to obtain more accurate results.

The new contributions of this research are as follows. This research introduces a new efficient kernel extreme learning machine model (WD-OSMSSA-KELM model) based on B-XGB feature selection and an enhanced multi-strategy variation of the salp swarm algorithm (SSA) and wavelet decomposition to increase the accuracy of PM2.5 prediction findings. To begin, we used B-XGB feature selection to determine the best characteristics for forecasting hourly PM2.5 concentrations and remove redundant features. Then, to reduce the prediction error caused by time series data, we used the wavelet decomposition

(WD) technique to achieve multi-scale decomposition results and single-branch reconstruction of PM2.5 concentrations. In the subsequent steps, we optimized the parameters of the KELM model for each regenerated component using the proposed OSMSSA versus other competitive peers. An enhanced version of the SSA with multiple exploratory and exploitative trends was developed to achieve higher performance for the basic SSA optimization and avoid local stagnation concerns. This paper presents novel procedures based on oppositional-based learning and simplex-based search to address the fundamental flaws of traditional SSA. Furthermore, we used a time-varying parameter instead of the SSA's primary parameter. We suggest utilizing random leaders to drive the swarm towards new parts of the feature space based on a conditional framework to increase the SSA's exploration tendencies. The developed model was assessed using data from the Beijing Municipal Environmental Monitoring Center's hourly database, six key air pollutants, and six significant meteorological components.

Here, we also review the related work on the proposed SSA optimizer in the remaining part.

#### *Literature Review of SSA*

Though there are many applications for the SSA, it suffers from the problems of unbalanced exploitation and exploration operations, local optimum stagnation, and poor exploitation. In order to alleviate these issues and enhance the working properties, many scholars actively study improving the performance of the SSA. Ren et al. [35] presented an adaptive weight Lévy-assisted SSA (WLSSA) and analyzed the optimization ability of the WLSSA. The adaptive weight mechanism extended the global exploration of the basic SSA, and the Lévy flight strategy improved the probability of the whole SSA to escape from local optima. The proposed WLSSA showed excellent performance by integrating the SSA with an adaptive weight mechanism and Lévy flight strategy. Besides, it was applied to three constrained engineering optimization problems in practice. Çelik et al. [36] propounded a modified SSA (mSSA) to solve the optimization problem on a large scale. The most important parameters for balancing exploitation and exploration in the basic SSA are changed chaotically from the first iteration to the last iteration by embedding a sinusoidal map. Besides, the reciprocal relationship between two leader individuals was introduced into the mSSA to improve its search performance. Moreover, a randomized technique was systematically applied to followers to provide the chain with diversity. This method solved several optimization issues in terms of the accuracy of the effective solution and the convergence trend line.

Aljarah et al. [37] proposed an improved multi-objective SSA with two basic components: dynamic time-varying strategy and local optimal solution. These components help the SSA balance local exploitation capacity and global exploration capacity. Salgotra et al. [38] presented a new enhancement to the SSA and proposed seven mutation operators to improve the working properties of the SSA, including Cauchy, Gaussian, Lévy, neighborhood-based mutation, trigonometric mutation, mutation clock, and diversity mutation.

Liu et al. [39] developed a new modified version of the SSA with a chaos-assisted trend and multi-population structure. The chaotic strategy was used to enrich the local exploitation of the SSA, and a multi-population structure with three sub-strategies was arranged to enhance the global exploration of the SSA. In the beginning, divide all the individuals into multiple sub-populations, which only explore the feasible region. Then, it should be noted that with the continuous development of evolution, the algorithm would gradually replace the global exploration with the focus on local exploration. Hence, depending on the iteration of this paper, the number of sub-populations would have different settings. The whole population would be dynamically divided into different numbers of sub-populations during evolution. Following the update of the SSA's individual position, the chaos-assisted exploitation approach was implemented to obtain additional possibilities to examine more interesting search regions.

Tubishat et al. [40] presented a new dynamic SSA (DSSA) for feature selection, which used Singer's chaotic map to increase diversification and provide a new local search strategy to boost the exploitative capability. Kansal and Dhillon [41] proposed an emended SSA (ESSA) to settle the multi-objective electric power load dispatch problem. The fuzzy set theory was used to change the multi-objective optimization problem into scalar objectives and through the basic priority to solve the conflict nature of the target. External penalty variable elimination was used to deal with the physical and operational constraints of the unit.

Zhang et al. [42] found a composite mutation strategy and restarted mechanism to improve the basic SSA. The former mutation schemes were inspired by the DE rand local mutation method of Adaptive CoDE, and the latter schemes help the worst individuals jump out of local optima. Elaziz et al. [43] used the DE operator as the local search operator to enhance the capability of the SSA to deal with multi-objective big data optimization. Tu et al. [44] proposed a valuable localization method based on reliable anchor pair selection (RAPS) and the quantum behavior SSA (QSSA) for anisotropic wireless sensor networks. The QSSA was a new SSA variant based on quantum mechanics and trajectory analysis.

Salgotra et al. [45] presented the adaptive SSA (ASSA), adding a logarithmic distributed parameter, which was based on the division of a generation. Equations based on GWO and CS were used for the first half of the generation, while the general SSA equations were used for the second half of the generation. It was explored by GWO-CS and developed by the SSA algorithm. Meanwhile, the logarithmic decreasing function replaced the basic parameter C1 to achieve a new equilibrium of global exploration and local exploitation.

Ren et al. [46] used a random replacement mechanism to speed up convergence and a double-adaptive weighting mechanism to enhance the SSA's exploitation and exploration capabilities. This enhanced method was named RDSSA. In the random replacement mechanism, according to the ratio of the remaining number of runs of the algorithm to the total number of runs compared with the Cauchy random number, the current position had a certain probability to be close to the optimal position, and for the later, the replacement probability was smaller. Inspired by PSO and RDWOA, the double-adaptive weight mechanism introduced two key weights to make the SSA have better global optimization ability in the early stage and better local search ability in the later stage.

Chouhan et al. [47] introduced the concept of inertia weight to the SSA for optimizing the coverage and energy efficiency of wireless sensor networks. Wang et al. [48] proposed a novel orthogonal lens opposition-based learning SSA, named the OOSSA. An adaptive strategy was used to develop the exploration capacity, and the lens opposition-based learning and orthogonal design were used to avoid local optima, while the use of ranking-based dynamic learning strategies also enhanced the local exploitation capacity. Majhi et al. [49] improved the performance of the SSA using a chaotic oscillation generated by the quadratic integration and fire neural model for function optimization.

At present, the combination of two algorithms to improve the performance and solve optimization problems is also a popular research trend. In fact, the advantages of the hybrid algorithm are eliminating each other's weaknesses to a certain extent and achieving a balance between exploitation and exploration to solve optimization problems. Neggaz et al. [50] improved the SSA for feature selection by taking inspiration from the sine cosine algorithm (SCA), which updated the position of followers in the SSA using sine/cosine operators. The combination strengthened the convergence capacity. Ewees et al. [51] modified the SSA by the firefly algorithm (FA) for an unrelated parallel machine scheduling problem. The FA technique was taken as the local search operator to improve the SSA's performance. Saafan and El-Gendy [52] improved the basic WOA by using the exponential relationships of its key parameters instead of linear relationships and introduced the improved WOA to the SSA for optimization problems. Ibrahim et al. [53] presented a hybrid method to improve the efficacy of exploitation and exploration for feature selection, which combined the SSA with PSO. Zhang et al. [54] was inspired by

the SSA and embedded the SSA into the conventional HHO to expand the search ability and increase the diversity of the population. Therefore, the hybridized SSA with other methods maintains a balance between global exploration and local exploitation and has been applied in many fields.

Despite all the advantages of the SSA in dealing with the different optimization cases reviewed above, there is still room for improvement. The satisfactory results of the SSA are diminished in some numerical cases due to its inertia to LOs and immature convergence. The basic SSA can still be improved in terms of diversification and intensification inclinations and their fine balancing state. It may be stuck in LOs. To accelerate the convergence propensities and avoid LOs, as well as control a fine balance among the searching trends of the SSA, we modified the original structure of the SSA. The extensive results show that the proposed mechanisms in the new variant of the SSA can highly mitigate the core problems of the SSA and improve its efficacy in dealing with the studied problems.

The remainder of this work is organized as follows: In Section 2, we review the main concepts. In Section 3, the proposed method is described in detail. Section 6 presents the results. Finally, Section 7 concludes with the main remarks of this work, in addition to presenting the main future directions for this paper.

## 2. Materials and Methods

### 2.1. Study Area and Data Description

The unavailability of research data has been a concern in the literature on air pollution forecasting as there is a need to use standard publicly open datasets. We concentrated on publicly released dataset to allow for independent testing and fair assessments of the model predictions. These air quality data from 1 January 2016 to 28 February 2017 [55], were taken from the mega-city of Beijing in China, which suffers from the side effects of air pollution. Thanks to the Beijing Municipal Environmental Monitoring Center, the dataset is publicly available (<https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data> (accessed on 22 June 2022)), and we have chose Aotizhongxin station; the dataset includes nationally controlled hourly data of the primary air pollutants of PM2.5 ( $\mu\text{g}/\text{m}^3$ ), PM10 ( $\mu\text{g}/\text{m}^3$ ), SO<sub>2</sub> ( $\mu\text{g}/\text{m}^3$ ), NO<sub>2</sub> ( $\mu\text{g}/\text{m}^3$ ), CO ( $\mu\text{g}/\text{m}^3$ ), and O<sub>3</sub> ( $\mu\text{g}/\text{m}^3$ ) and six meteorological features such as the temperature (degrees Celsius) (TEMP), pressure (hPa) (PRES), dew point temperature (degrees Celsius) (DEWP), precipitation (mm) (RAIN), wind direction (WD), and wind speed (m/s) (WSPM). The statistical info of the dataset is reported in Table 1.

**Table 1.** Statistical info of the dataset.

Index	PM10	SO <sub>2</sub>	NO <sub>2</sub>	CO	O <sub>3</sub>	TEMP
<b>count</b>	10,200	10,200	10,200	10,200	10,200	10,200
<b>mean</b>	96.20167	12.23108	50.80431	1256.775	59.98735	12.04015
<b>std</b>	92.06978	16.39837	36.47572	1319.423	57.43134	12.0805
<b>min</b>	3	2	2	100	2	-16.8
<b>25%</b>	31	2	22	500	11	1.1
<b>50%</b>	70	6	41	800	50	11.35
<b>75%</b>	129	15	71	1500	85	23
<b>max</b>	884	341	218	10000	350	37.3
Index	PRES	DEWP	RAIN	wd	WSPM	PM2.5
<b>count</b>	10,200	10,200	10,200	10,200	10,200	10,200
<b>mean</b>	1014.321	0.150667	0.069069	7.698922	1.873206	77.4951
<b>std</b>	10.68535	14.44595	0.910892	4.62194	1.194687	84.0535
<b>min</b>	989.7	-35.3	0	1	0	3
<b>25%</b>	1005.1	-12	0	4	1.1	17
<b>50%</b>	1014.6	-1.2	0	7	1.6	49
<b>75%</b>	1023.1	13.2	0	11	2.4	106
<b>max</b>	1042	27.3	46.4	16	8.9	898

## 2.2. Kernel Extreme Learning Machine

KELM is a new variant of the well-regarded extreme learning machine (ELM) developed for the first time by Huang et al. [56]. The KELM integrates the kernel function with the structure of ELM to guarantee that the resulting network can reveal an acceptable generalization efficacy and enhance the forward learning speed rate [57,58]. Based on the  $L$  hidden nodes in the output layer (OL), the output function of single-hidden-layer feed-forward neural networks (SLFNs) is formulated as follows:

$$f(x) = \sum_{i=1}^L \beta_i g_i(x) = \sum_{i=1}^L \beta_i g(w_i x + b_i) = \sum_{i=1}^L \beta_i G(w_i, b_i, x) \quad (1)$$

The function expressed in Equation (1) can be reformulated as:

$$f(x) = h(x) \cdot \beta \quad (2)$$

where  $\beta = [\beta_1, \beta_2, \beta_3, \dots, \beta_L]^T$  denotes the vector of output weights in between the hidden layer with  $L$  neurons and the output neurons and  $h(x) = [h_1(x), h_2(x), \dots, h_L(x)]$  shows the output vector relative to the hidden layer of input  $x$ , which is utilized to map the info from the input feature space to the ELM-based feature space. In KELM, the integration of ELM with a positive coefficient in order to assist it in the learning system can result in more stability for the network. When it is not singular and during the processing of the output weight  $\beta$ , the coefficient  $C$  can be inserted into the diagonal of  $HH^T$ :

$$\beta = H^T \left( \frac{1}{c} + HH^T \right)^{-1} T \quad (3)$$

In this regard, the output function of the regularized ELM can be calculated as follows:

$$F(x) = h(x)\beta = h(x)H^T \left( \frac{1}{c} + HH^T \right)^{-1} T \quad (4)$$

ELM with a kernel matrix is expressed as:

$$\Omega_{KELM} = HH^T : \Omega_{KELM}^{i,j} = h(x_i) \cdot h(x_j) = K(x_i, x_j) \quad (5)$$

and the output function is:

$$f(x) = h(x)H^T \left( \frac{1}{c} + HH^T \right)^{-1} T = \begin{bmatrix} k(x, x_1) \\ \dots \\ k(x, x_N) \end{bmatrix}^T \left( \frac{1}{c} + \Omega_{KELM} \right)^{-1} T \quad (6)$$

In this situation, no info about the hidden layer feature map  $h(x)$  is required as it is replaced with the matched kernel function  $K(u, v)$ . The most well-regarded kernel function is the Gaussian kernel function, which is calculated based on Equation (7):

$$K(u, v) = \exp(-\gamma \|u - v\|^2) \quad (7)$$

where  $\gamma$  is a parameter used to manage the width of the sample Gaussian distribution. According to all previous works on KELM, it has been verified that the optimal selection of these parameters ( $C, \gamma$ ) has a significant impact on the efficiency of KELM [35,59]. Therefore, it is required to optimize these parameters based on an efficient method.

## 2.3. Wavelet Transform

The wavelet transform (WT) technique has been developed based on the idea of short-time Fourier transform localization. This popular approach has been widely utilized in the literature to mitigate the inadequacies of the window size not varying with the

frequency [60]. The WT idea can offer the core prediction system a “time-frequency” window that varies with the frequency. Consider that  $\varphi$  is to detect the mother wavelet; the continuous WT (CWT) can be expressed by Equation (8) [61]:

$$\omega(b, c) = \int f(t) \times \left( \frac{1}{\sqrt{b}} \right) \varphi\left(t - \frac{c}{b}\right) dt \quad (8)$$

where the  $b$  factor is a scale and shows the stretch or duration of the wavelet. The  $c$  factor is a transfer parameter providing the needed time concentration and expressing the point of the wavelet on the time pivot. In order to deal with discrete equations, the continuous WT needs to be discretized. The wavelet coefficients may be found at any point in the waveform ( $b$ ) and for any scale value ( $c$ ) based on Equation (9):

$$\varphi_{b,c}(t) = \frac{1}{\sqrt{b}} \varphi\left(\frac{|t - c|}{b}\right) \quad (9)$$

These transform and scale factors are disconnected based on Equation (10):

$$b = 2^k, c = 2^l \quad (10)$$

where  $k$  and  $l$  are integers. By changing  $b$  and  $c$  in the above rule, we can attain the relation in Equation (11):

$$\varphi_k, l(t) = 2^{-\frac{k}{2}} \varphi[2^{-k}t - l] \quad (11)$$

Hence, the wavelet function is a discrete wavelet. The DWT can be obtained using the rule in Equation (12):

$$\omega(b, c) = 2^{-\frac{k}{2}} \int f(t) \times \varphi(2^{-k}t - l) dt \quad (12)$$

#### 2.4. Boruta-XGBoost Method

The Boruta method was proposed by Kursa and Rudnicki [62], as a wrapper feature selection technique to determine the most relevant features required for effective prediction and to construct the ML model using the most important features of the dataset. The XGBoost technique can be used as the base algorithm for Boruta, and the suggested method is called the B-XGB algorithm. This method uses a modified variant of the BorutaPy Python module to work with XGBoost. The B-XGB technique has the ability to detect the most-important predictor variables using the Z-scores of each input predictor for the duplicate (shadow) property. We can obtain the Z-score as in the rule in Equation (13).

$$Z\text{-score} = \frac{\text{MDA}}{\text{SD}} \quad (13)$$

where MDA states the mean decrease in accuracy of the input and shadow variables and SD indicates the standard deviation of precision losses.

The main stages of the B-XGB algorithm are explained in Algorithm 1.

---

**Algorithm 1** Steps of the B-XGB algorithm.

---

**start**

1. Construct the characteristics at random. All of the characteristics in the dataset have been randomly scrambled. Furthermore, their numerical order should be changed.
2. Find the relative importance of the initial features and shadow characteristics, based on obtaining the Z-score premium rate using the XGBoost strategy.
3. Select the most-relevant features according to the Z-score. If the Z-score of a base feature is greater than the top Z-score in the set of shadow features, we consider the feature as “important”; otherwise, we can remove it.
4. Repeat Steps 1-3 until all features are processed.

**end**

---

## 2.5. Mathematical Model of SSA Optimization

The SSA is a class of the population-based optimizer proposed in 2017 based on the swarming foraging of salps to deal with complex and numerical optimization problems [63]. From the optimization perspective, salp chains provide more chances for the SSA to avoid premature convergence and the inertia to the local optima (LOs) to some degree. Matter of fact, the basic SSA frequently fails to maintain a steady balance between exploration and exploitation impulses. Consequently, the SSA may fail to achieve an excellent optimum in dealing with some practical cases. In the SSA, two classes of salps (search agents) are employed to perform the diversification and intensification phases: the leader agent and follower agents. The leader agent should guide and direct the other agents. Hence, it is situated at the head of the chain, whereas other search agents should follow their leader.

In the SSA, population  $X$  will include  $N$  salps (particles or search agents) with  $d$  dimensions. Therefore, the population is presented by a  $N \times d$ -dimensional matrix expressed in Equation (14):

$$X_i = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix} \quad (14)$$

In the SSA, all agents try to track and chase the food source at the intended location. Therefore, the state of the leader agent is determined based on Equation (15):

$$x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & c_3 < 0.5 \end{cases} \quad (15)$$

where  $x_j^1$  denotes the state of the leader,  $F_j$  denotes the state of the food source in the  $j$ th dimension,  $ub_j$  and  $lb_j$  are the boundaries of the  $j$ th dimension,  $c_2$  and  $c_3$  are random numbers in  $[0, 1]$ , and  $c_1$  is the only adaptive parameter of the technique, which can be expressed as in Equation (16):

$$c_1 = 2e^{-(\frac{4t}{L})^2} \quad (16)$$

where  $t$  is the iteration and  $L$  denotes the upper limit of the iterations. The parameter  $c_1$  is designed to help the SSA make a better stability between the exploration and exploitation inclinations. Follower agents update their locations according to Equation (17):

$$x_j^i = \frac{x_j^i + x_j^{i-1}}{2} \quad (17)$$

where  $i \geq 2$  and  $x_j^i$  shows the situation of the  $i$ th search agent at the  $j$ th dimension.

The pseudo-code of the original SSA is described in Algorithm 2.

---

### Algorithm 2 Pseudo-code of the SSA.

---

```

Generate random agents  $x_i$  ( $i = 1, 2, \dots, n$ )
while (Looping condition is not met) do
    Check the fitness of the agents
    Find the fittest solution, and record it as food source  $F$ 
    Update  $c_1$  using Equation (16)
    for (all agents ( $x_i$ )) do
        if ( $i \leq n/2$ ) then
            Update the position of the leader agent using Equation (15)
        else ( $i > n/2$  and  $i \leq n + 1$ )
            Update the position of the follower agents using Equation (17)
        Update the swarm inside the range of the variables
        Return back agents that violate the boundaries
    Return  $F$ 

```

---

## 2.6. Opposition-Based Learning

OBL is a machine-learning-based concept firstly presented by Tizhoosh [64], which is related to finding paired potential agents from a collection of seed points. This association is called “opposite”, and it considers paired candidates to map the agents for speeding up the coverage rate of optimizers and enhancing the accuracy of the search. While evaluating an agent  $y$  for a specific situation, calculating the opposing candidate might increase the algorithm’s chances of discovering a better agent that is nearer the desired point. OBL has some formal definitions, which can be presented as follows [64]:

**Definition 1.** *Opposite number:* Let  $y \in [lb, ub]$  be a real number. The opposite number  $\tilde{y}$  is obtained by [64]:

$$\tilde{y} = lb + ub - y \quad (18)$$

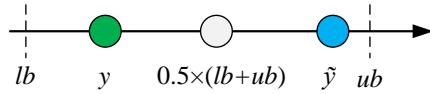
where  $lb$  and  $ub$  indicate the objective space’s lower and upper boundaries, correspondingly.

In the multidimensional version, the definition of  $\tilde{y}$  is expressed as in Definition 2 [64].

**Definition 2.** Suppose  $y = [y_1, y_2, \dots, y_n] \in \mathbb{R}^n$ , where  $y_1, y_2, \dots, y_n \in \mathbb{R}$  and  $y_j \in [lb_j, ub_j]$ . The opposite point  $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_n]$  is realized by [64]:

$$\tilde{y}_j = lb_j + ub_j - y_j, j = 1, 2, \dots, n \quad (19)$$

As can be seen in Figure 1 and by focusing on the description of the opposite location, the OBL-based optimization can be explained as follows:



**Figure 1.** Opposite pair inside  $[lb, ub]$ .

**Definition 3.** *Opposition-based optimization:* In this approach, the opposite agent  $\tilde{y}$  should be replaced with matching agent  $y$  with regard to the excellence of the fitness function  $f(\cdot)$ . If  $f(y)$  is superior to  $f(\tilde{y})$ , then  $y$  will be used; else,  $y = \tilde{y}$ . Therefore, both matched agents are evaluated at once to carry on the search with the superior one.

## 2.7. Simplex Search

Simplex search is a well-known, very powerful local search (local descent) proposed by Nelder and Mead [65], which can be utilized for optimization purposes, which does not need the gradient info of the feature landscape [66]. A simplex can be explained as a geometrical concept (polytope) with  $(n + 1)$  points  $z_1, \dots, z_n$  in an  $n$ -dimensional space. The procedures of this search can re-scale a simplex using the local info of the objective by using four operators: reflection, expansion, contraction, and shrinkage [65,66]. The stages of the search can be summarized as follows (see Figure 2):

- Step 1. Obtain the fittest location  $z_g$ , the second-best fitting point  $z_b$ , and the worst point  $z_s$ . The fitness values are  $f(z_g)$ ,  $f(z_b)$ , and  $f(z_s)$ .
- Step 2. Attain the mid-point of  $z_g$  and  $z_b$ .

$$z_c = (z_g + z_b)/2 \quad (20)$$

- Step 3. Run the reflection operator to obtain reflection point  $z_r$ . The reflection factor  $\alpha$  is often fixed to 1.

$$z_r = z_c + \alpha(z_c - z_s) \quad (21)$$

- Step 4. When  $f(z_r) < f(z_g)$ , the point is expanded, while the expansion factor is often fixed to 2:

$$z_e = z_c + \gamma(z_r - z_c) \quad (22)$$

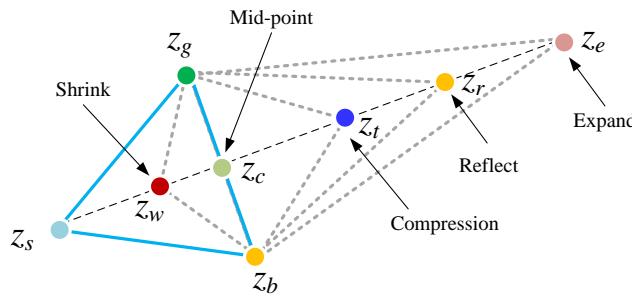
- When  $f(z_e) < f(z_g)$ ,  $z_s$  is replaced with  $z_e$ ; else,  $z_s$  replaces  $z_r$ .
- If  $f(z_r) > f(z_s)$ , compression point  $z_t$  is obtained using the compression operator. The compression factor  $\beta$  is often fixed to 0.5:

$$z_t = z_c + \beta(z_s - z_c) \quad (23)$$

- When  $f(z_g) < f(z_r) < f(z_s)$ , the shrinkage operator is used to obtain the compression point  $z_w$ . The shrinkage factor is often fixed to  $\beta$ :

$$z_w = z_c - \beta(z_s - z_c) \quad (24)$$

When  $f(z_w) < f(z_s)$ ,  $z_s$  replaces  $z_w$ ; else, substitute  $z_s$  with  $z_r$ .



**Figure 2.** Diagram of the simplex scheme.

### 3. The Proposed OSMSSA Algorithm

Despite the merits of the SSA such as its simplicity and efficacy on mathematical problems, it may still fall into local optima (LOs) in dealing with the optimization of the KELM in the wavelet PM2.5 prediction system. Therefore, there is room for further improvements to the crucial exploratory and exploitative inclinations of the SSA to avoid possible stagnation drawbacks. In order to further alleviate the immature convergence and core stagnation behaviors of the SSA, first, the oppositional-based learning (OBL) paradigm was embedded in the basic structure of the SSA.

#### 3.1. OBL-Based Search

It has been proven that OBL can improve the convergence trends of optimizers and boost their exploratory behaviors by expanding the search space. Like other meta-heuristics, the SSA initiates the exploration by generating a set of random initial salps (random agents). A well-completed initialization phase can have a great impact on the convergence of optimizers. According to the overall distances of random agents to the leader of the chain, the SSA has to dedicate more efforts to attract all those agents towards the leader. These distances can delay the convergence and reduce the speed of the search. In this case, OBL can solve this problem by generating new opposite agents. If the agent is far from the leader, the closer agent can be found by searching in the opposite direction. Consequently, the agents become closer to the leader, and quicker convergence can be observed especially on more complex landscapes.

In the proposed OSMSSA, the initialization is instigated by creating a set of random salps  $X$  of size  $N$ , in which the salp  $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ ,  $i = 1, 2, \dots, N$ . Then, OBL is utilized to attain the opposite pair for each salp. Hence, the opposite chain  $\tilde{X}$  is generated. Regarding two chains  $X$  and  $\tilde{X}$ , the fittest  $N$  salps are selected based on the fitness values. Figure 3 shows the opposite pairs on two chains. In the OSMSSA, the OBL is used based on the modified formula described in Equation (25):

$$\tilde{x}_j = lb_j + ub_j - F_j + r_1(F_j - x_j), j = 1, 2, \dots, n \quad (25)$$

where  $\tilde{x}_j$  is the position of the opposite salps. Algorithm 3 shows the pseudo-code of the OBL process in the OSMSSA.

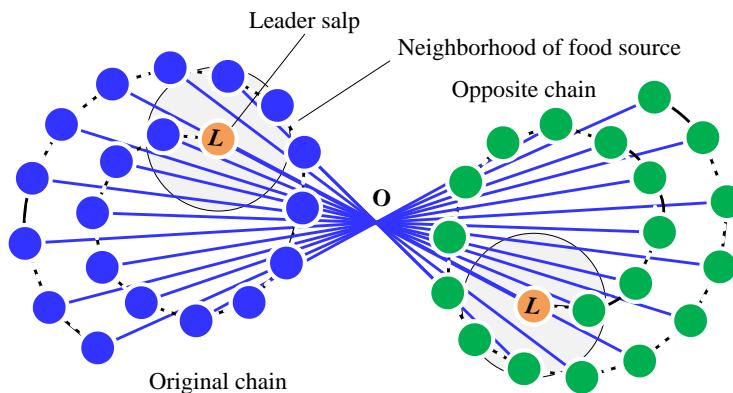
**Algorithm 3** Pseudo-code of the OBL-based process.**start**Generate the randomly distributed population of agents  $x_j (j = 1, 2, \dots, n)$ 

Evaluate the fitness of all agents

Find the fittest solution, and record it as food source  $F$ 

Calculate the opposite chain of agents using Equation (25)

Evaluate the fitness of all agents and their opposite pairs

Find the  $N$  fittest agents from the integrated set, and record them as new population  $X'$ .**end****Figure 3.** A salp chain and its opposite salp chain.

### 3.2. Dynamic Parameter

In the SSA, the  $c_1$  parameter is utilized as a condition to balance the main inclinations of the SSA. However, this condition can be improved by considering another decreasing randomized condition. According to the  $c_1$  parameter in Equation (16), a randomized function is added to the OSMSSA to assist the algorithm in better (smoother) switching from the exploration to exploitation mechanism. This function is defined as in Equation (26):

$$I = 2c'_1 \times q - c'_1 = 4qe^{-(\frac{2t}{L})^2} - 2e^{-(\frac{2t}{L})^2} \quad (26)$$

where  $q$  is a random number in  $(0, 1)$ . When  $|I| > 1$ , the leader is updated; otherwise, when  $|I| < 1$ ,  $i > n/2$ , and  $i < n + 1$ , the OSMSSA will perform the rule in Equation (17). This parameter can further improve the capabilities of the OSMSSA in the fine balancing of the exploratory and exploitative inclinations.

### 3.3. Random Food Source

In the SSA, the state of the leader is updated only with respect to the location of the food source (best solution) [67]. This rule can restrict the exploration potential of the SSA compared to the situation that randomly selected food sources are used to generate the leader. To allow the salps to perform more random jumps and improve the exploratory behaviors of the algorithm, the rule in Equation (15) is modified and the distances of the salps from the random food sources are also considered as in Equations (27) and (28):

$$D = \left| \frac{1}{2}q' \cdot F_{r,j} - x_j \right| \quad (27)$$

$$x_j^1 = \begin{cases} 2(F_{r,j} - A.D) + c_1((ub_j - lb_j)c_2 + lb_j) & q' < 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & q' \geq 0.5 \end{cases} \quad (28)$$

where  $q'$  is a random number inside  $(0, 1)$  and  $F_{r,j}$  is a randomly selected food source in the  $j$ -th dimension. This operator can assist the proposed OSMSSA to further explore

untouched parts of the search space before performing the exploitative phases. It also puts more emphasis on the random nature of the OSMSSA.

### 3.4. Simplex-Based Search

The location of the food source in the SSA has a remarkable impact on the quality of the found agents because it guides the leader salp and then the follower salps. However, if the best search agent is trapped in an LO, the SSA will easily face the stagnation drawback. One effective way to improve the quality of the food source is to utilize the simplex search. As stated earlier, the simplex method can adapt itself to the local topography of the search space and contracts onto the concluding optima. To scan the neighborhood area of the LO and increase the chance of jumping out of the LO, simplex method disturbance is applied to the food source during the iterations. Using this strategy, we can exploit the vicinity of the food source, more efficiently. In addition, the OSMSSA will find more high-quality leaders and salps.

### 3.5. Pseudo-Code of OSMSSA

The pseudo-code of the proposed OSMSSA is described in Algorithm 4.

---

**Algorithm 4** Pseudo-code of the proposed OSMSSA.
 

---

```

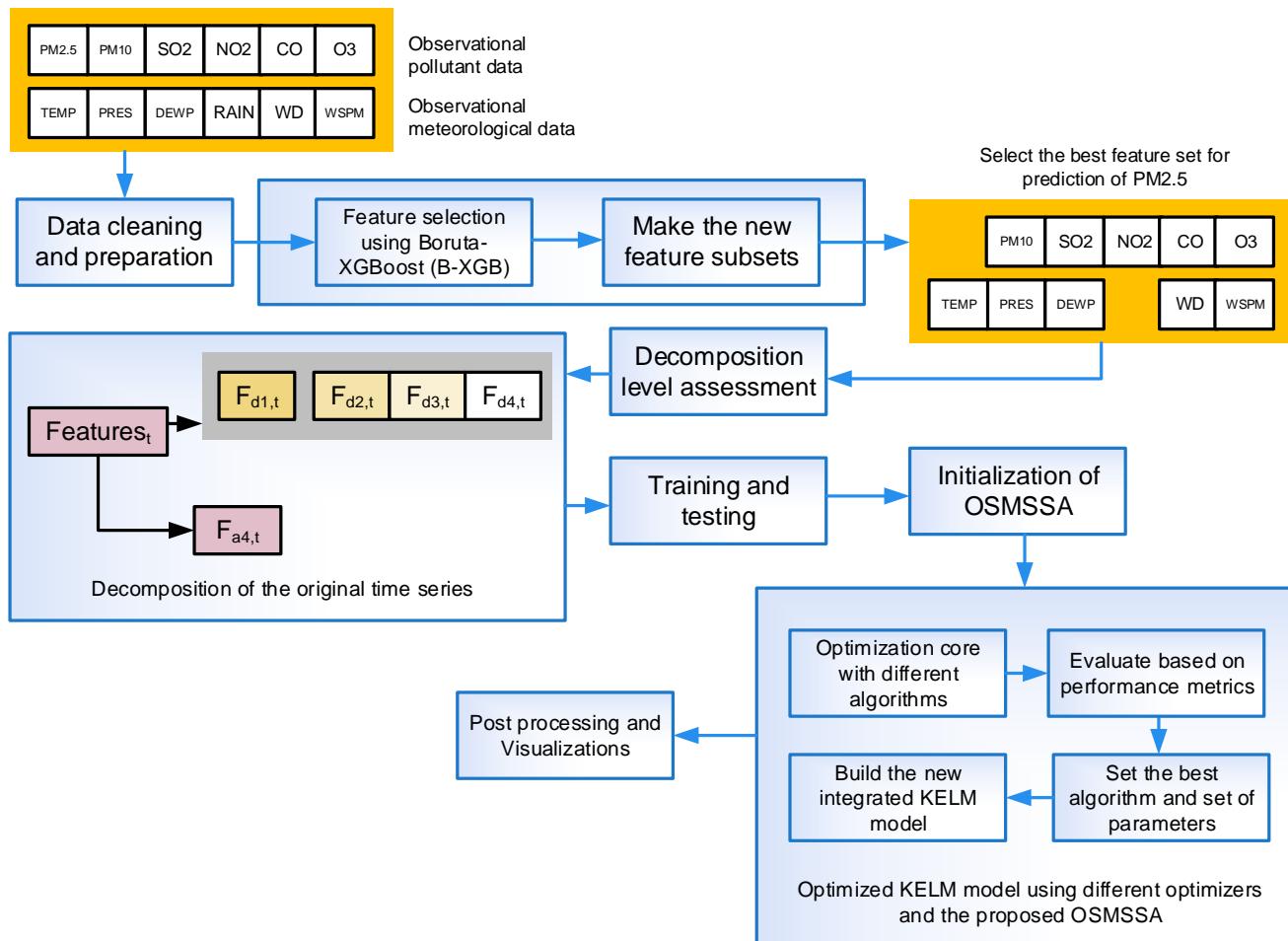
Generate initial swarm using the OBL-based scheme in Algorithm 3
while (Looping condition is not met) do
    Evaluate the fitness of the swarm
    Find the fittest agent, and record it as food source  $F$ 
    Update  $c_1$  using Equation (26)
    Perform the simplex search, and obtain the improved food source  $F$ 
    for (all salps ( $x_i$ )) do
        if ( $|I| > 1$  and  $i \leq n/2$ ) then
            Update the leader agent using Equation (28)
        else ( $|I| < 1$  and  $i > n/2$  and  $i < n + 1$ )
            Update the follower agent using Equation (17)
        Update the population inside the boundaries
        Return back agents that violate the boundaries.
    Return  $F$ 
```

---

## 4. Construction of the Proposed WD-OSMSSA-KELM Model

In this section, we present the stages of the proposed WD-OSMSSA-KELM model. In the first step, the algorithm conducts the data cleaning and filling of the empty cells using the average of the neighborhood cells. The database includes hourly data, six primary air pollutants, and six relevant meteorological features collected from the Beijing Municipal Environmental Monitoring Center. After removing such cells, the dataset is ready to be processed using the Boruta-XGBoost (B-XGB) feature selection algorithm. In this stage, the irrelevant features are removed from the initial data sheets, and then, the new set of features is recorded in a new database. In the third stage, the optimal level of decomposition is obtained and the wavelet decomposition (WD) algorithm is performed to reach the multi-scale decomposition results and single-branch reconstruction of the input features, as well as mitigate the prediction error produced by the initial time series data. After this stage, we made the training and testing set to divide the data for the prediction stage. For the prediction core, first, the algorithm optimizes the structure of the KELM model under each reconstructed component using the improved version of the SSA. When compared to other common machine learning methods such as ELM, the KELM technique offers robust performance, a faster training speed, and better modeling precision. We chose the SSA algorithm for optimizing KELM as it is relatively fast, has few parameters, is simple, is well-known, and is easy to implement. In the OSMSSA, new operators based on oppositional-based learning, random leaders, a time-varying structure, and simplex-

based search are performed to mitigate the local stagnation problems of the conventional SSA when dealing with the optimization of KELM. After this stage, the integrated KELM model was tested on the training data. The optimized KELM model was utilized to predict the PM<sub>2.5</sub> concentrations, and different error metrics were considered in the evaluation stage. In the last stage, the new framework was visualized and the obtained error results compared with other regression methods, while the predicted results were visualized against the initial measured data. The main stages of the proposed WD-OLMSSA-KELM framework are demonstrated in Figure 4.



**Figure 4.** Flowchart of the proposed wavelet PM<sub>2.5</sub> prediction system (WD-OSMSSA-KELM) with Boruta-XGBoost feature selection.

## 5. Evaluation Index

In this section, we present the utilized statistical metrics for measuring the performance of the studied ML models, including correlation coefficient ( $R$ ), mean absolute percentage error ( $MAPE$ ), root-mean-squared error ( $RMSE$ ), median absolute error ( $MdAE$ ), and Kling–Gupta model efficiency ( $KGE$ ). These metrics can be formulated as follows in predicting of the  $Z$  parameter:

$$R = \frac{\sum_{i=1}^N (Z_{M,i} - \bar{Z}_M) \cdot (Z_{P,i} - \bar{Z}_P)}{\sqrt{\sum_{i=1}^N (Z_{M,i} - \bar{Z}_M)^2 \sum_{i=1}^N (Z_{P,i} - \bar{Z}_P)^2}} \quad (29)$$

$$RMSE = \left( \frac{1}{N} \sum_{i=1}^N (Z_{M,i} - Z_{P,i})^2 \right)^{0.5} \quad (30)$$

$$MAPE(\%) = \left( \frac{100}{N} \right) \sum_{i=1}^N \left| \frac{Z_{M,i} - Z_{P,i}}{Z_{M,i}} \right| \quad (31)$$

$$MdAE = \text{median}_{i=1,\dots,N} |Z_{M,i} - Z_{P,i}| \quad (32)$$

$$KGE = 1 - \sqrt{(R - 1)^2 + (\frac{SD_P}{SD_M} - 1)^2 + (\frac{\bar{R}_P}{\bar{R}_M} - 1)^2} \quad (33)$$

where  $Z_{M,i}$  and  $Z_{P,i}$  express the measured and predicted values of  $Z$  at the  $i$ th time step and  $\bar{Z}_M$  and  $\bar{Z}_P$  denote the mean values of the measured and predicted  $Z$ . SD is the standard deviation of the difference between the measured and predicted values of  $Z$ .  $SD_M$  and  $SD_P$  show the standard deviation between the measured and predicted values of  $Z$ .  $R_P$  and  $R_M$  are the average  $R$  of the predicted and measured values.

## 6. Results

### 6.1. Experimental Environment

To provide a fair performance analysis, we used the identical conditions for all experiments in this study, and an operating system with the info presented in Table 2 was utilized to check the performance of all methods.

**Table 2.** The detailed settings of the utilized system.

Name	Setting
<b>Hardware</b>	
CPU	Intel Core(TM) i3 processor
Frequency	3.1 GHz
RAM	8 GB
Hard drive	1000 GB
<b>Software</b>	
Operating system	Windows 7 64-bit
Languages	MATLAB R2018a and Python 3
Packages	Mljar [68], Pandas, Scikit-Learn, NumPy

### 6.2. Pre-Processing and Feature Selection

#### 6.2.1. Missing Data Imputation

Similar to any other datasets obtained from sensors, the Beijing air quality data suffer from some missing values. We propose a hybrid approach to fill in the missing cells to deal with this shortcoming. In this approach, we utilized two approaches depending on the gap size. Small gaps with unavailable cells were replaced by linear interpolation [69], and for significant gaps, the missing cell was solved using the average value; the threshold of the gap size helps to select which of the strategies to be employed. We chose this threshold as 5. We applied the proposed imputation method for all features, and only the wind direction was the exception. We used the last valid observation for the missing cell in the wind direction because this feature can only use discrete values from 1 to 16, standing for north, south–southeast, etc.

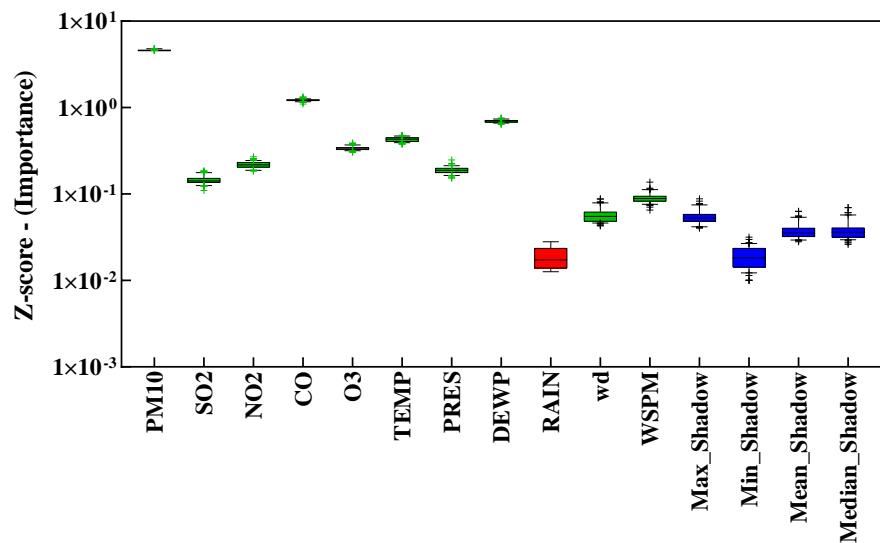
In the first step, the algorithm conducts the data cleaning and filling of the unavailable cells based on the above approach.

#### 6.2.2. Boruta-XGBoost Feature Selection

The database includes hourly records, six primary air pollutants, and six relevant meteorological features obtained from the Beijing Municipal Environmental Monitoring Center. After removing empty cells, the dataset is ready to be processed using the Boruta-

XGBoost (B-XGB) feature selection algorithm for finding the most important features for the prediction task.

For feature selection, we employed the Boruta method, developed by Kursa and Rudnicki [62] in 2010, as a robust and well-established algorithm in the literature [70]. We configured XGBoost with 100 trees, and the maximum tree depth (MTD) was set to 20. The MTD was set to a high value to help us identify higher-order feature interactions that we may see in some datasets [71]. The median Z-scores for the different features are exposed in the form of box plots over 100 generations in Figure 5.



**Figure 5.** Boxplot of median Z-scores attained by the Boruta-XGB algorithm.

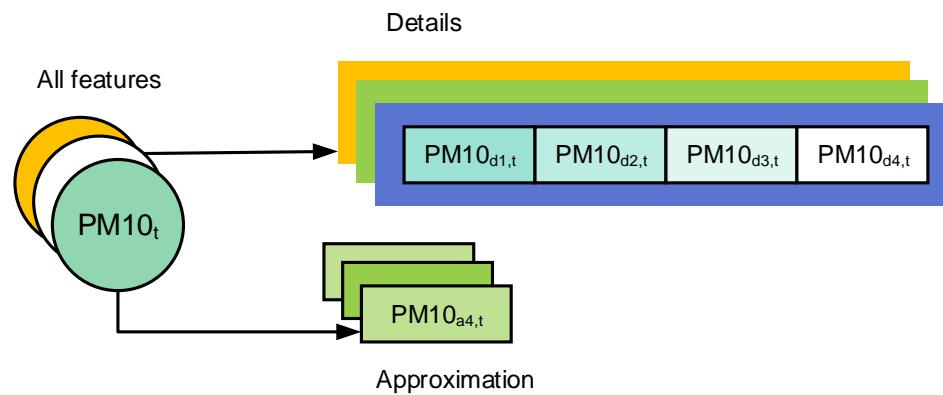
Figure 5 helps us see the relevant values by the green boxes, in which these features are accepted as the relevant input for the solo and complementary ML-based models. As seen in Figure 5, by comparing *ShadowMax* and *Z – scores*, the RAIN feature is red and detected as redundant input, then is discarded from the dataset. Therefore, the other ten input combinations were individually maintained to improve the proposed ML model and predict the hourly PM2.5.

### 6.3. Wavelet Decomposition and Reconstruction Results

For the decomposition of the data, we utilized the mother wavelet *Dmey* and four levels following the formula in Equation (34) [72,73] to detect the optimal decomposition level of PM10, SO<sub>2</sub>, NO<sub>2</sub>, CO, O<sub>3</sub>, TEMP, PRES, DEWP, WD, and WSPM.

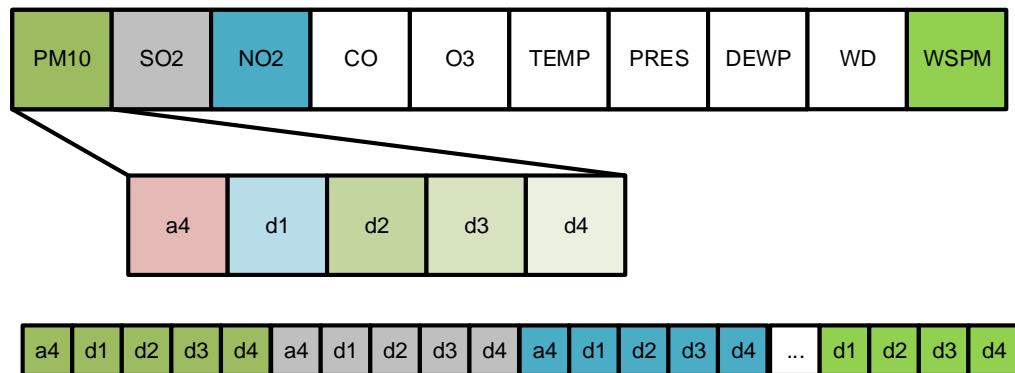
$$nMW = \text{int}[\log(N)] \quad (34)$$

in which *N* denotes the dataset's size, accounting for 10,200. Hence, the optimal level is four for our study. As a result, we decomposed the original signals into four levels of detail and approximations, as shown in Figure 6.



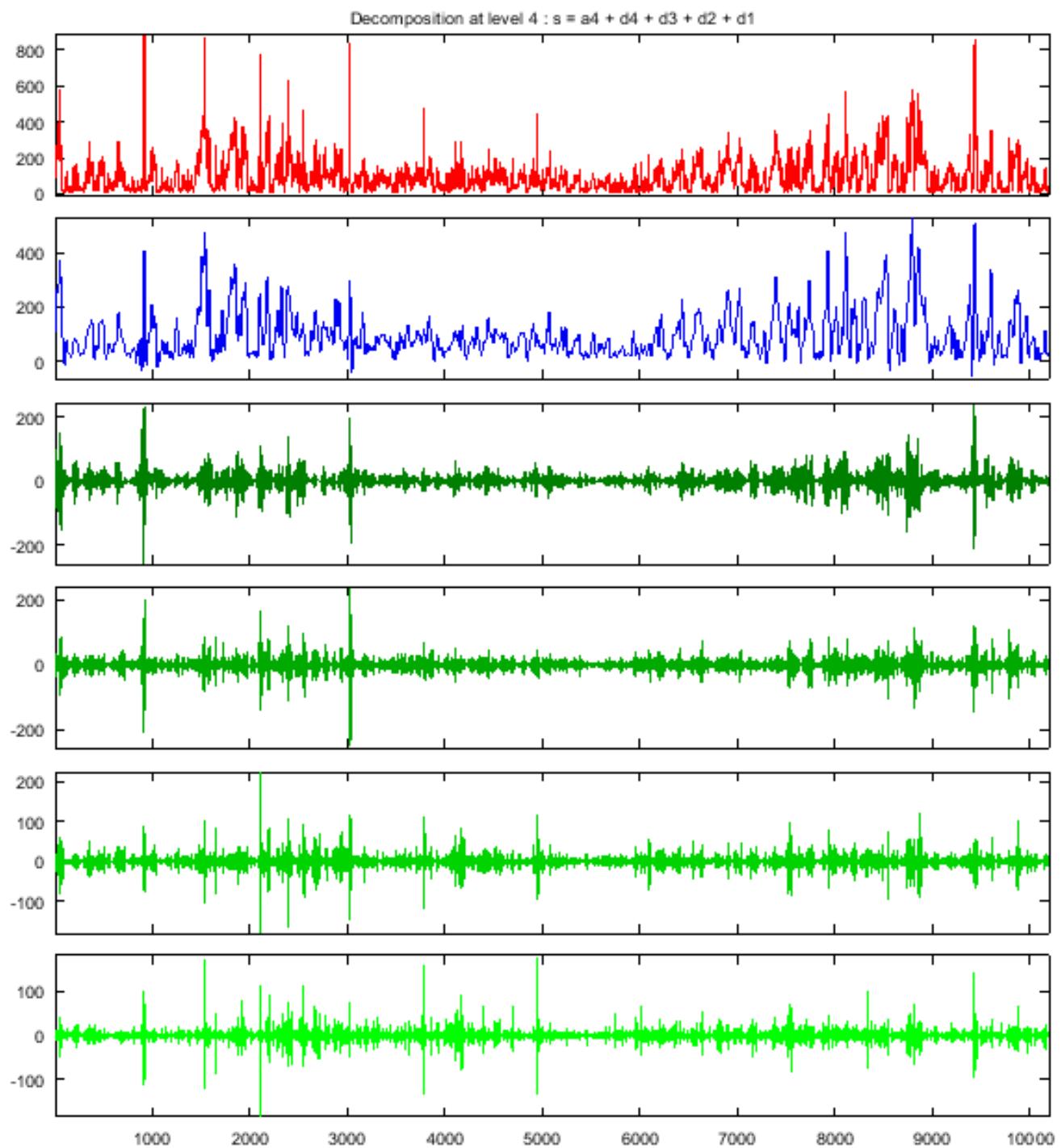
**Figure 6.** Decomposition of datasets using the DWT.

The structure of the input data after decomposition is described in Figure 7. After decomposition of the relevant air pollutants of the original, hourly time series of PM10, SO<sub>2</sub>, NO<sub>2</sub>, CO, and O<sub>3</sub> and the original, hourly time series of the meteorological features of TEMP, PRES, DEWP, WD, and WSPM based on the wavelet decomposition method, we made a new sheet of datasets to be inserted into the prediction module. In the new sheet, each feature column was replaced with its decomposed time series of  $a_4, d_1, d_2, d_3$ , and  $d_4$ , respectively. Hence, if we have a column for PM10, we will have five new columns of  $a_4, d_1, d_2, d_3$ , and  $d_4$  resulting from the decomposition of the original time series of PM10. After the feature selection step, the same process was implemented for the other features.

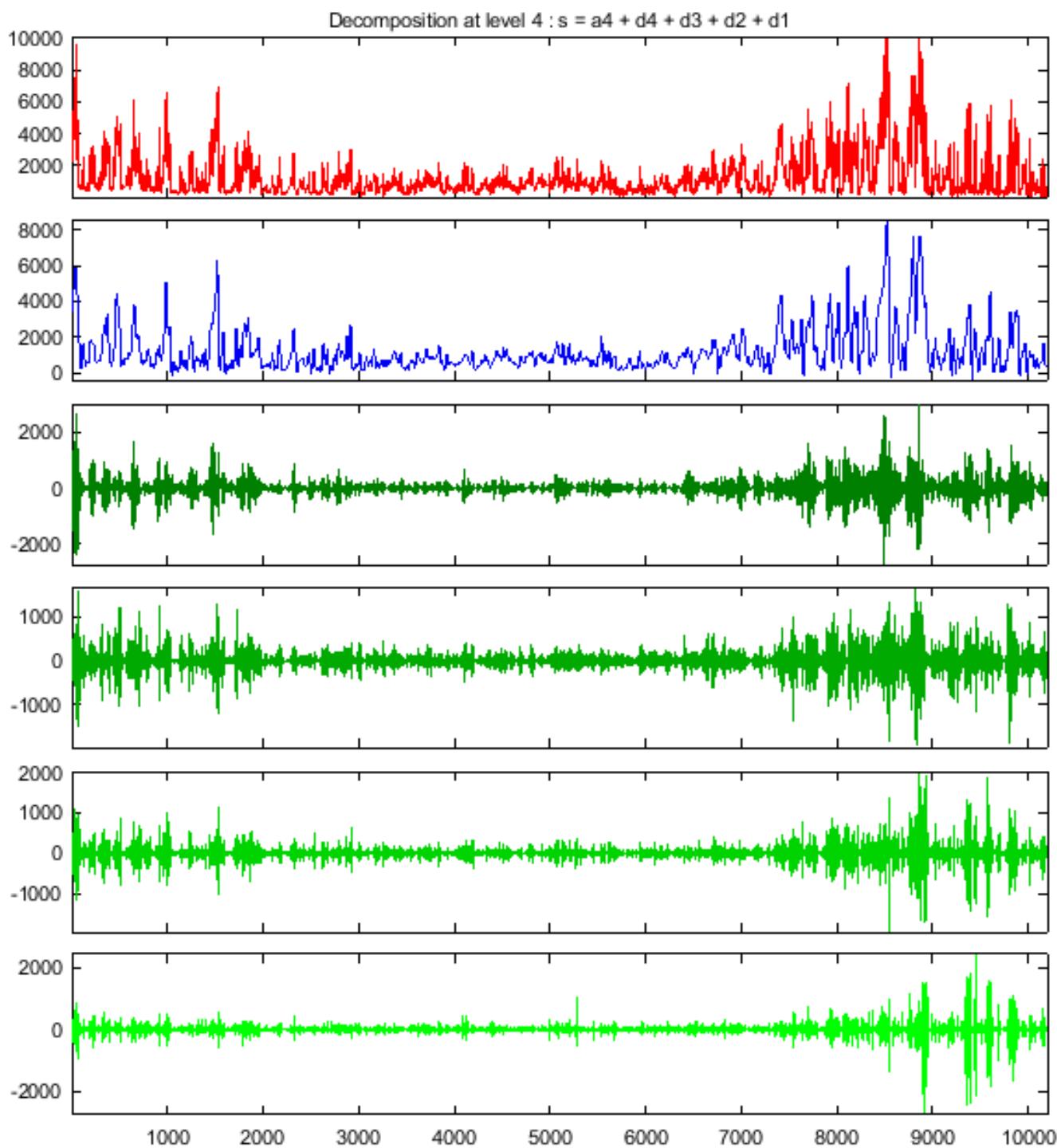


**Figure 7.** Structure of the decomposed input data.

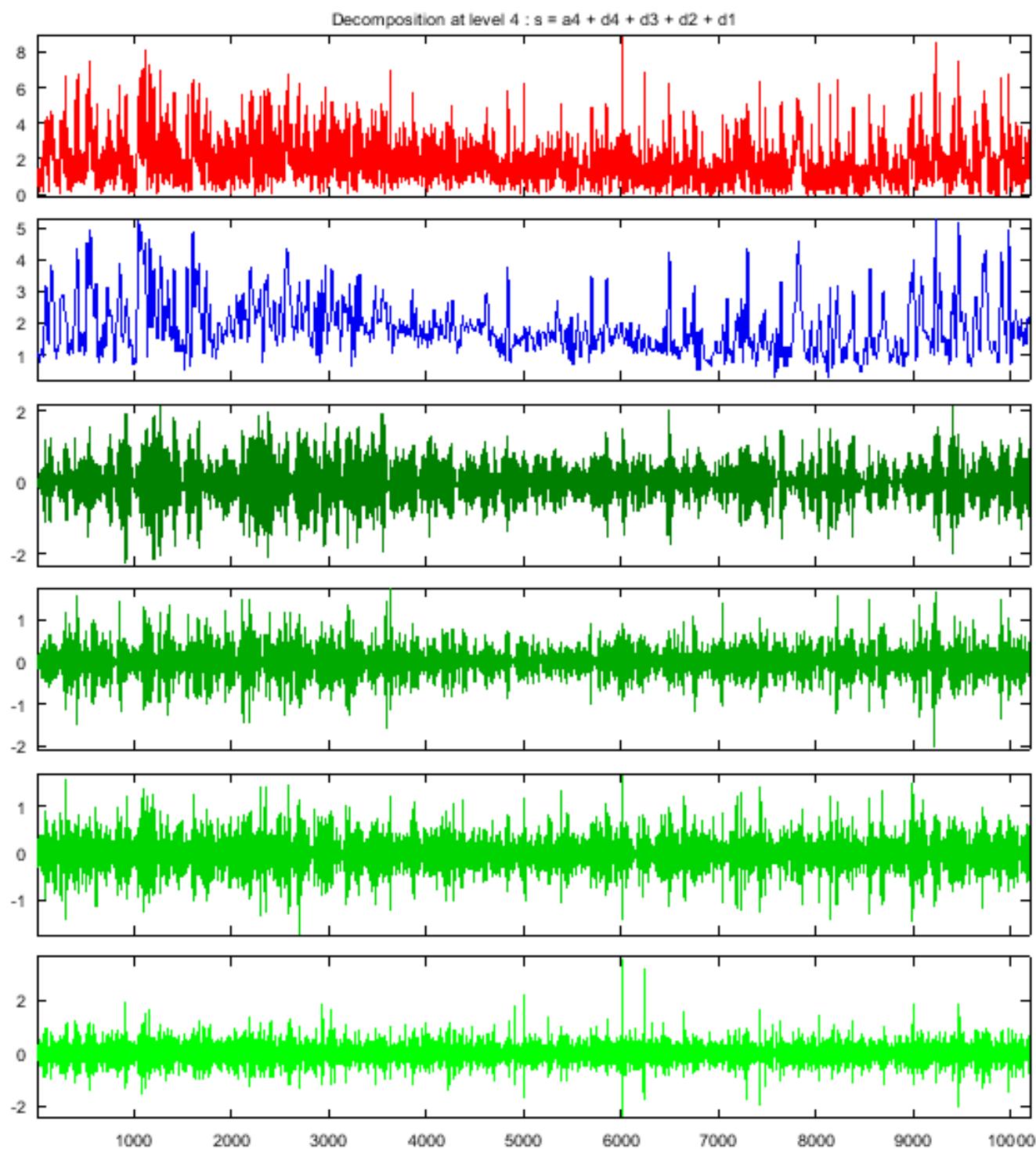
The final decomposition results for the three features are shown in Figures 8–10. All features resulting from the feature selection step were decomposed in the same manner, and we only show some decomposition results here due to limited space. Full results and extra info about the system have been reported in (<https://github.com/aliasgharheidaricom/Wavelet-PM2.5-Prediction-System> (accessed on 28 September 2022))



**Figure 8.** Decomposition of PM10 hourly data. The X-axis is hours, and the first part in each decomposed feature is  $a_4, d_4, d_3, d_2, d_1$ , from top to bottom, respectively



**Figure 9.** Decomposition of CO hourly data. The X-axis is hours, and the first part in each decomposed feature is  $a4, d4, d3, d2, d1$ , from top to bottom, respectively



**Figure 10.** Decomposition of WSPM hourly data. The X-axis is hours, and the first part in each decomposed feature is  $a4, d4, d3, d2, d1$ , from top to bottom, respectively

#### 6.4. Comparison of the WD-OSMSSA-KELM with Other Optimized KELM Models

In this section, we performed experiments on the proposed WD-OSMSSA-KELM framework to evaluate the optimization core of the method versus other optimized WD-KELM models. These tests aimed to detect which algorithm can reveal better efficacy in optimizing the structure of the WD-KELM model. For the ease of reporting in the tables and plots, we used a shorter form of naming for the optimized models, such as SSAKELM instead of WD-SSA-KELM. We selected the approaches for optimization based on the fact

that they are the most well-established and reliable in the literature, and they outperformed other swarm-based methodologies. The optimization algorithms experimented on included the bat algorithm (BAT) [74], differential evolution (DE) [75], whale optimizer (WOA) [76], ant lion optimizer (ALO) [77], and basic SSA algorithms. We performed all tests based on fair comparison rules. To ensure a fair comparison, we fixed the population size and the maximum iterations to 30 and 100 for all the WD-KELM models and the average of 30 runs obtained.

Table 3 reports the training results of the OSMSSAKELM versus other optimized KELM models. As per the results in Table 3, we can see the best method in terms of R was OSMSSA, followed by DE, SSA, ALO, BA, and WOA, respectively. We can observe that the OSMSSA optimizer also provided the best results regarding the RMSE, MdAE, and MAPE, while DE and the SSA obtained the second and third best positions for these metrics. The RMSE of 2.83 for the OSMSSA shows it had the best prediction performance and comprises both precision and accuracy for PM2.5 in training, while the original KELM obtained an RMSE of 35.86, followed by an RMSE of 11.824768 for the SSA. In terms of the KGE metric, we can see that the OSMSSA obtained the best result, while DE, SSA, ALO, BA, and WOA were in the following stages. This shows that the OSMSSAKELM model achieved the best results with superior model efficiency, accuracy, precision, and consistency. These results verify the effectiveness of the OSMSSA integrated with KELM compared to other optimizers, and it shows that the OSMSSAKELM model can reach the best results for the PM2.5 prediction in the training phase compared to methods such as the SSA or DE.

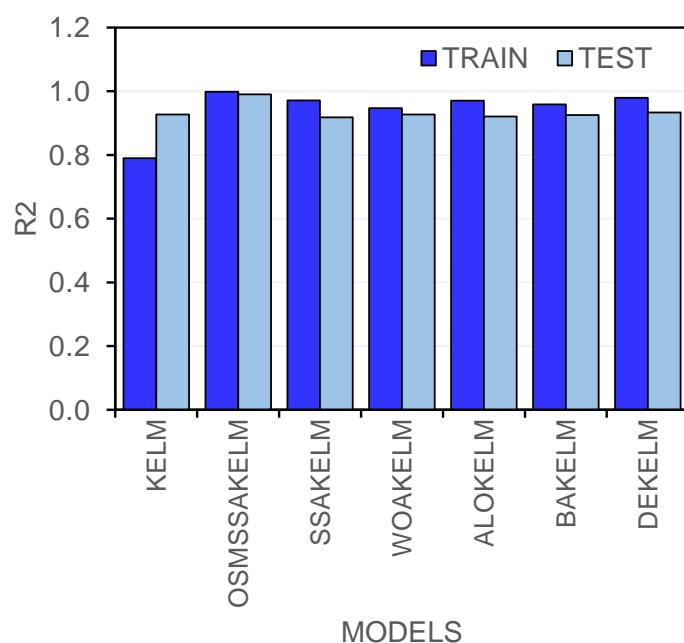
**Table 3.** Comparison of the average training results for different optimized KELM models.

Metrics/Models	KELM	OSMSSAKELM	SSAKELM	WOAKELM	ALOKELM	BAKELM	DEKELM
R	0.8886	<b>0.99915</b>	0.98579	0.97325	0.985204	0.97933	0.98949
RMSE	35.8648	<b>2.833373</b>	11.8248	16.4749	12.14324	14.322	9.97997
MdAE	17.6389	<b>0.559553</b>	2.90888	4.37457	3.203245	4.15657	1.26851
MAPE	77.5981	<b>2.432532</b>	12.6649	16.6247	13.27894	16.16	6.94091
KGE	0.62233	<b>0.997251</b>	0.94784	0.90514	0.941573	0.92818	0.97555
R2	0.78961	<b>0.9983</b>	0.97179	0.94722	0.97063	0.95909	0.97909

Table 4 reports the testing results obtained by the OSMSSAKELM versus other optimized KELM models. The testing results of the RMSE also verify the higher accuracy of the predicted results using the proposed OSMSSAKELM model versus its peers. The OSMSSA enhanced the RMSE of 46.106 for KELM to 11.906. The results also show that the optimized models, especially in the case of the OSMSSA ( $R^2$ : 0.990083, RMSE: 11.90632, MAPE: 9.768373), could boost the standalone KELM models' accuracy. Regarding the MdAE, we can see that the OSMSSAKELM, with an MdAE of 2.42, significantly outperformed its peers. Based on the MAPE of 9.768373 for the proposed method, we see a significant gap with other versions. The same advantage can be seen in the Kling–Gupta model efficiency metric, with a KGE of 0.963327 for the proposed variant. In addition, the histograms of the  $R^2$  index for the testing and training stages are shown in Figure 11. According to this comparison, we can observe the apparent advantage of the OSMSSAKELM hybrid model compared to its peers.

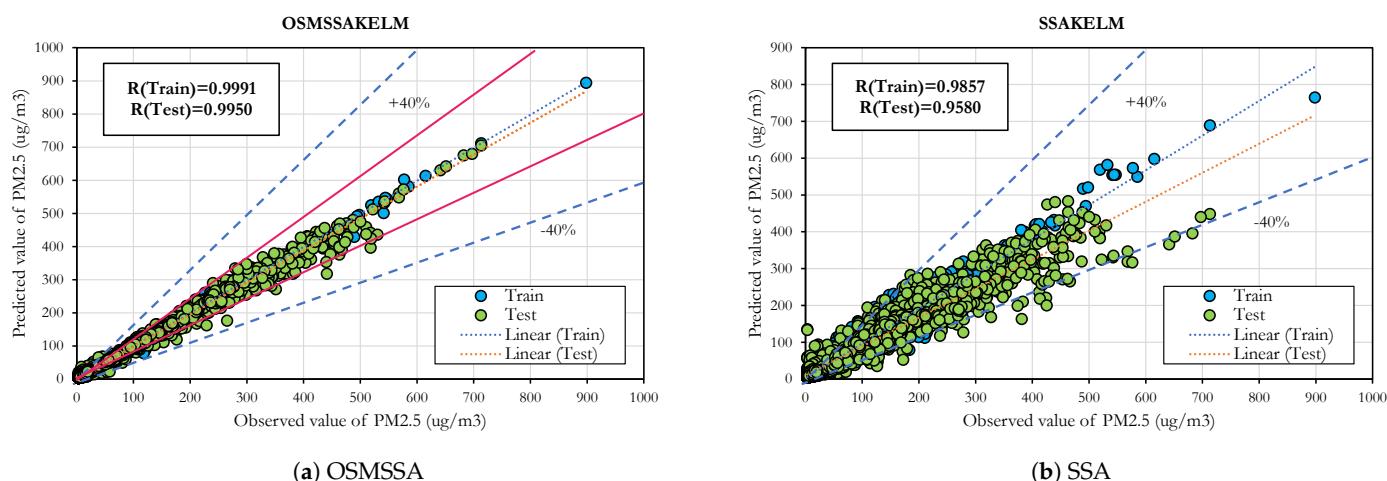
**Table 4.** Comparison of the average testing results for different optimized KELM models.

Metrics/Models	KELM	OSMSSAKELM	SSAKELM	WOAKELM	ALOKELM	BAKELM	DEKELM
R	0.9629	<b>0.995029</b>	0.95807	0.96264	0.959784	0.96197	0.96627
RMSE	46.106	<b>11.90632</b>	38.4236	33.7723	37.54477	36.2508	31.449
MdAE	16.4749	<b>2.424515</b>	10.6253	10.5895	10.58449	10.3691	9.20276
MAPE	69.8938	<b>9.768373</b>	41.0507	32.613	41.31555	40.5519	32.0398
KGE	0.66069	<b>0.963327</b>	0.78098	0.84842	0.787966	0.79972	0.87443
R2	0.927178	<b>0.990083</b>	0.91789	0.926666	0.921185	0.925386	0.933678

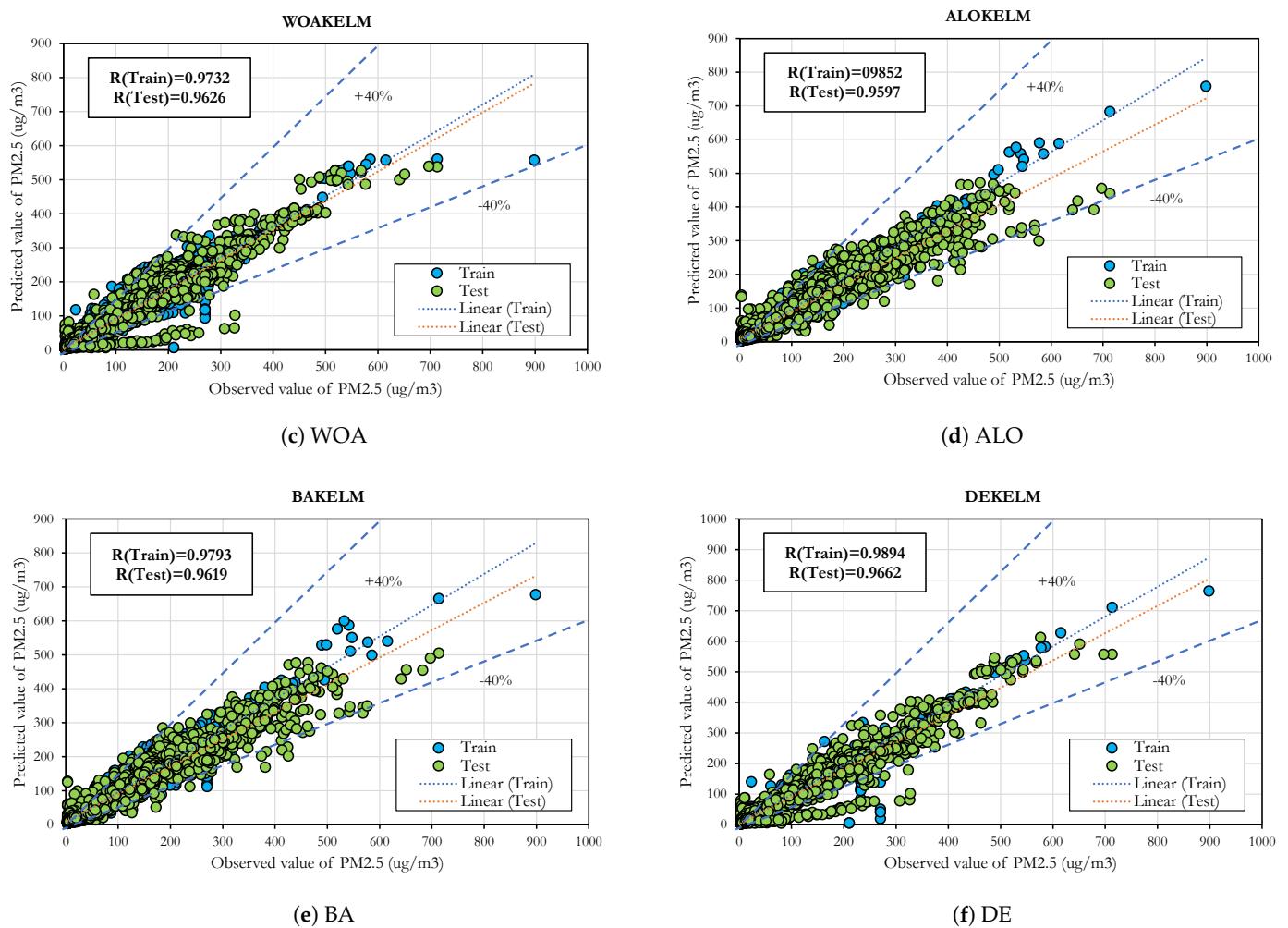


**Figure 11.** The histograms of the  $R^2$  index for the testing and training stages.

Figure 12 shows the scatter data on the observed PM2.5 data against the predicted PM2.5 data for the training and testing stages. These plots give a visual analysis of the predicted PM2.5 scores and prediction errors. The plots verify the higher correlation and strong relationship between the data pairs. As we can observe, the scatter plot for the OSMSSAKELM verifies the better regression performance compared to other combinations of KELM models. We can see that for most methods, the proportion of the error was located among  $\pm 40\%$ , while the proposed method was inside the range of  $\pm 20\%$ . From Figure 12, it is observed that the OSMSSA can enhance the correlation coefficient for training and testing data to  $R = 0.9991$  and  $R = 0.9950$ , respectively. These results also verify that the OSMSSAKELM is the best combination for further experiments.



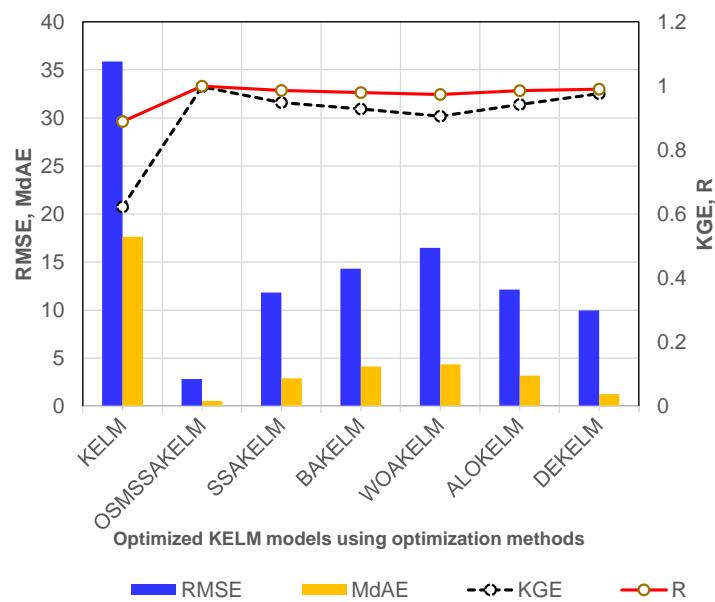
**Figure 12.** *Cont.*



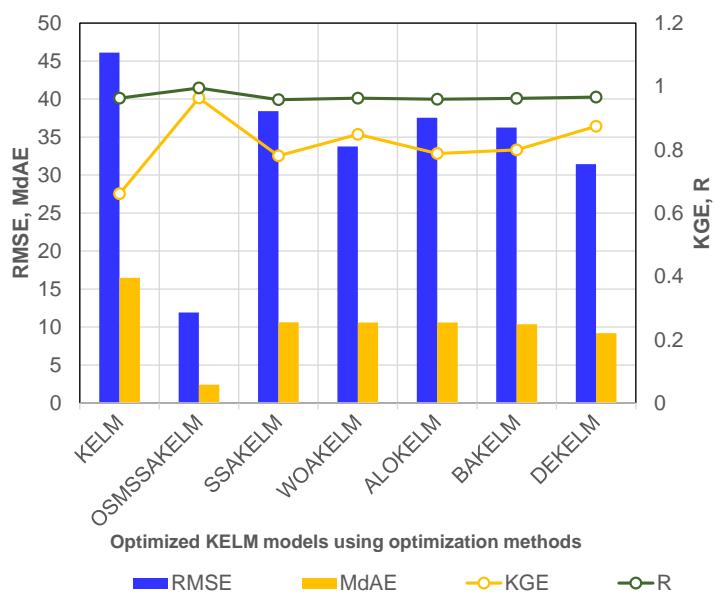
**Figure 12.** Scatter plots of different optimized KELM models.

Figure 13 gives a comparison of different optimized KELM methods for the training stage using the RMSE, MdAE, KGE, and R metrics. Furthermore, Figure 14 compares the results of different optimized KELM methods within the testing phase based on the RMSE, MdAE, KGE, and R metrics. As per the results in Figures 13 and 14, the proposed OSMSSAKELM system had the best prediction performance, both for the training and testing stages. Compared to the other variants of the KELM methods, the error was minimal in both stages, and the fit was the best. Based on the histograms in Figures 13 and 14, we can also observe that the conventional KELM model cannot perform well in dealing with the decomposed original sequence, while the combined model with the OSMSSA could reach the best regression results.

Taylor diagrams offer a concise statistical description of how much the predicted and observed patterns fit each other in terms of the correlation and variance ratio [78]. It helps us to show which of the numerous approximations (or models) of predicted PM2.5 data is the most rewarding. Taylor diagrams of the training and testing stages for all optimized KELM models are demonstrated in Figures 15 and 16.

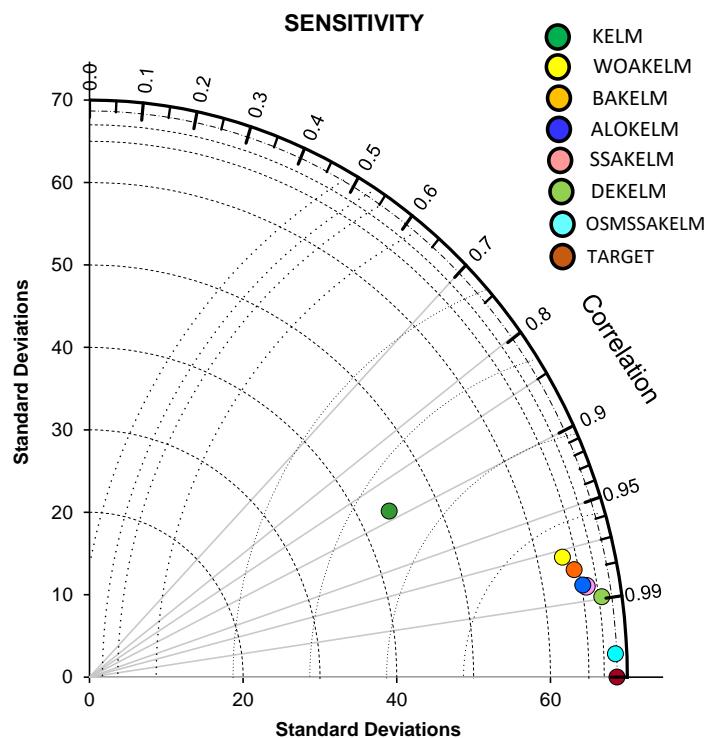


**Figure 13.** Comparison of optimized KELM models for the training phase based on the RMSE, MdAE, KGE, and R metrics.

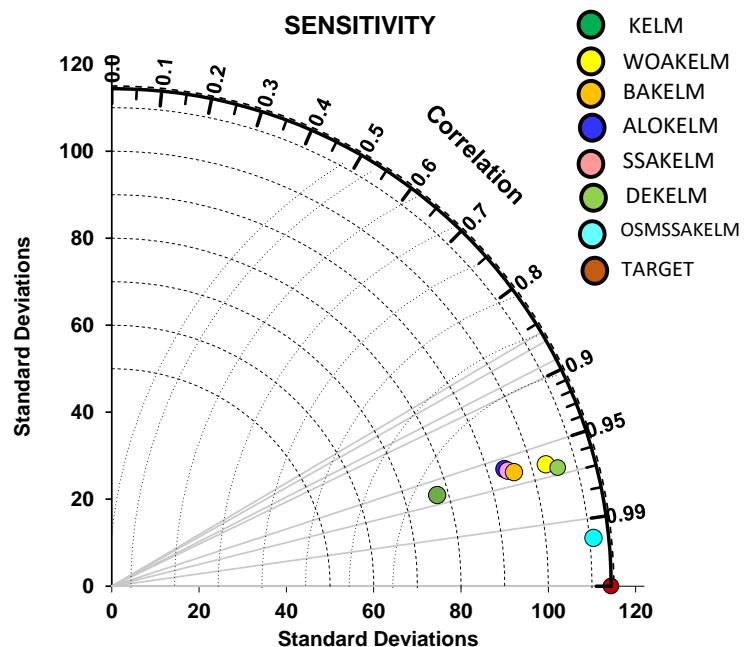


**Figure 14.** Comparison of optimized KELM models for the testing phase based on the RMSE, MdAE, KGE, and R metrics.

The correlation coefficient (R) was projected using the Taylor diagram to evaluate the overall performance of the models, with its details for the models' efficiency [72]. The plot exhibited a more perceptible and credible relationship between predicted and observed PM<sub>2.5</sub> data as per the R and standard deviation. As a consequence, compared to the other versions, the OSMSSAKELM had the best performance for PM<sub>2.5</sub> prediction and was the nearest to the target point. Based on the preceding optimization model study, we can conclude that the proposed OSMSSAKELM had the best prediction effect on PM<sub>2.5</sub> data. In both the testing and training stages, the hybrid model suggested in this study demonstrated high model efficiency, accuracy, and practicality. In this part, the comparative findings confirmed that the error range was minimum and optimal, and the volatility was low, resulting in an ultimate prediction effect.



**Figure 15.** Taylor plot of optimized KELM models for the training phase.



**Figure 16.** Taylor plot of optimized KELM models for the testing phase.

### 6.5. Comparison with Other WD-Based Machine Learning Models

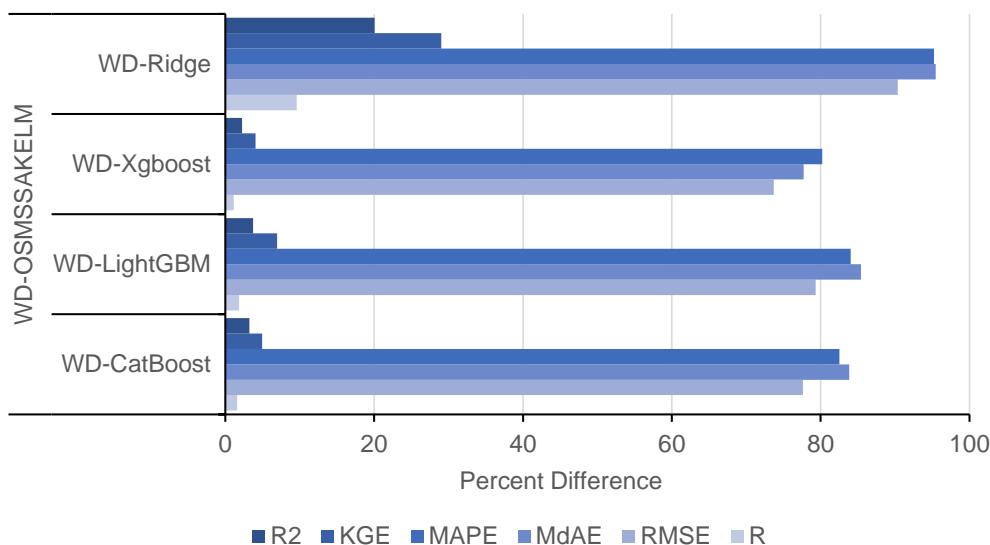
In this section, we compared the prediction performance of the proposed system versus other representative WD-based ML models. For this aim, we compared the WD-OSMSSAKELM model with the most well-known, strong, and competitive models in the literature, including the CatBoost regressor (CatBoost) [79], light gradient boosting machine (LightGBM) [80], extreme gradient boosting (Xgboost) [81], and ridge regression [82] models. We utilized the MLJAR package for the python implementation of these methods provided by Płońska and Płoński [68].

Table 5 compares the results of the proposed WD-OSMSSAKELM model for the training phase versus other studied WD-based models based on different metrics. Table 6 also reports the predictive analysis for the proposed WD-OSMSSAKELM model for the testing phase versus other studied WD-based methods using different metrics.

The training RMSE results in Table 5 show that the WD-OSMSSAKELM with an RMSE of 2.833 can outperform other regression models, followed by WD-Xgboost, WD-CatBoost, WD-LightGBM, and WD-Ridge, respectively. Based on an MdAE of 0.559, we can see the proposed model is on the frontier, while the WD-Xgboost and WD-CatBoost obtained results of 2.5103 and 3.45918, respectively. The results in Table 5 vividly show that the proposed WD-OSMSSAKELM model ( $R: 0.999$ , RMSE: 2.833, MdAE: 0.559, MAPE: 2.432, KGE: 0.997,  $R^2: 0.998$ ) can obtain more high-quality results than its peers. The percent of improvement of the metric results of the proposed WD-OSMSSAKELM model against the other models is shown in Figure 17. As per the results in this plot, we can see significant improvements in the results for the new model. For example, in terms of the MAPE, the results of the WD-OSMSSAKELM were enhanced 82.51%, 84.03%, 80.21%, and 95.22% compared to those for WD-CatBoost, WD-LightGBM, WD-Xgboost, and WD-Ridge, respectively.

**Table 5.** Comparison results of the proposed model for training phase versus other studied models based on different metrics.

Metrics/Models	WD-CatBoost	WD-LightGBM	WD-Xgboost	WD-Ridge	WD-OSMSSAKELM
R	0.983429	0.981051	0.988064	0.911808	<b>0.99915</b>
RMSE	12.6632	13.71367	10.77369	29.39649	<b>2.833373</b>
MdAE	3.459178	3.833557	2.510303	12.31832	<b>0.559553</b>
MAPE	13.91329	15.23907	12.29618	50.93051	<b>2.432532</b>
KGE	0.950138	0.93236	0.958283	0.773029	<b>0.997251</b>
$R^2$	0.967133	0.962461	0.97627	0.831393	<b>0.998302</b>



**Figure 17.** Percent (%) of improvement of the WD-OSMSSAKELM versus the values of other methods for the metrics during the training phase.

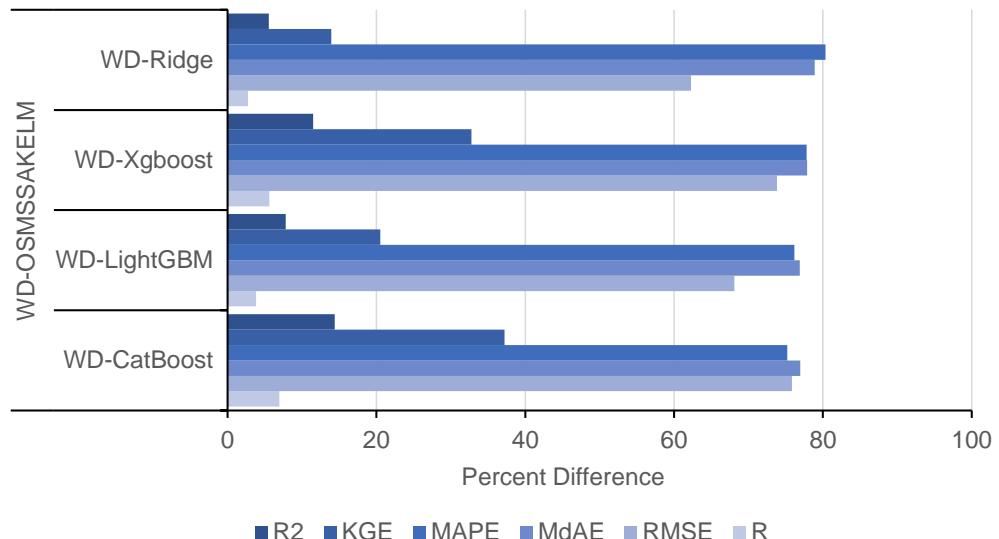
Table 6 shows the testing results obtained by the WD-OSMSSAKELM versus the other studied models. The testing results of the RMSE also verify the higher accuracy of the predicted results using the proposed WD-OSMSSAKELM model versus its peers. The results show that the WD-OSMSSAKELM with an RMSE of 11.906 can outperform other regression models, followed by WD-Ridge, WD-LightGBM, WD-Xgboost, and WD-CatBoost, respectively. Regarding the MdAE, we understand that the WD-OSMSSAKELM significantly outperformed its peers with an MdAE of 2.424. Based on an MAPE of 9.768 for

the proposed technique, we realized a significant gap with other models, while the second- and third-best methods obtained 39.421 and 40.972, respectively. The same advantage can be seen in the Kling–Gupta model efficiency metric, while WD-Ridge and WD-LightGBM obtained the next ranks.

**Table 6.** Comparison results of the proposed model for the testing phase versus other studied models based on different metrics.

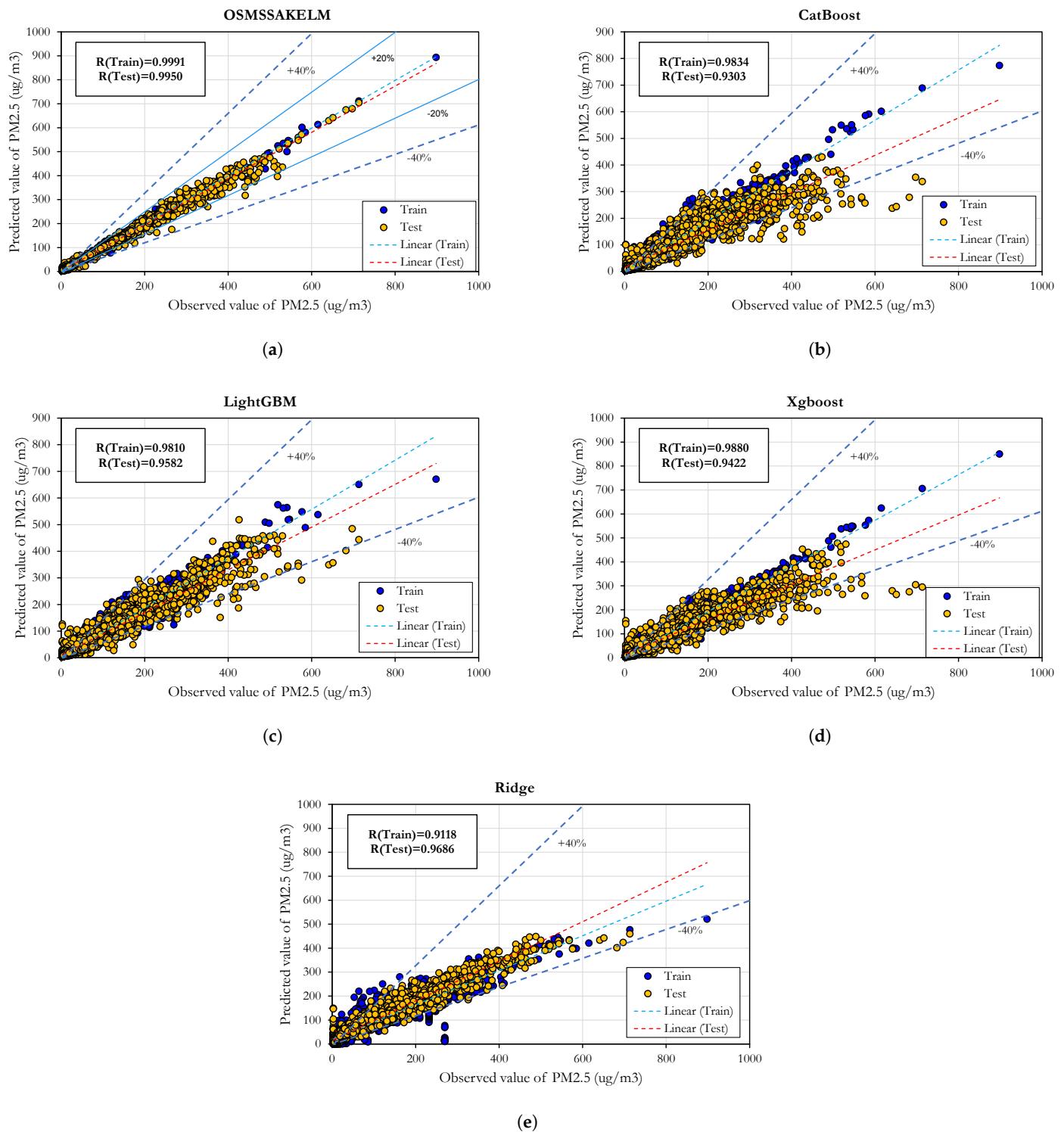
Metrics/Models	WD-CatBoost	WD-LightGBM	WD-Xgboost	WD-Ridge	WD-OSMSSAKELM
R	0.930334	0.95828	0.942253	0.968619	<b>0.995029</b>
RMSE	49.25268	37.33286	45.47209	31.56917	<b>11.90632</b>
MdAE	10.5139	10.48497	10.95629	11.48737	<b>2.424515</b>
MAPE	39.42185	40.97232	44.05232	49.74391	<b>9.768373</b>
KGE	0.701979	0.79923	0.725486	0.84559	<b>0.963327</b>
$R^2$	0.865521	0.9183	0.887841	0.938223	<b>0.990083</b>

The percent of improvement of the metric results of the proposed WD-OSMSSAKELM model against the other models for the testing stage is shown in Figure 18. Based on the results in this analysis, we can detect significant enhancements in the quality of the results for the new method. For example, in terms of the RMSE, the results of the WD-OSMSSAKELM improved up to 75.826%, 68.107%, 73.816%, and 62.284% compared to those for WD-CatBoost, WD-LightGBM, WD-Xgboost, and WD-Ridge, respectively.



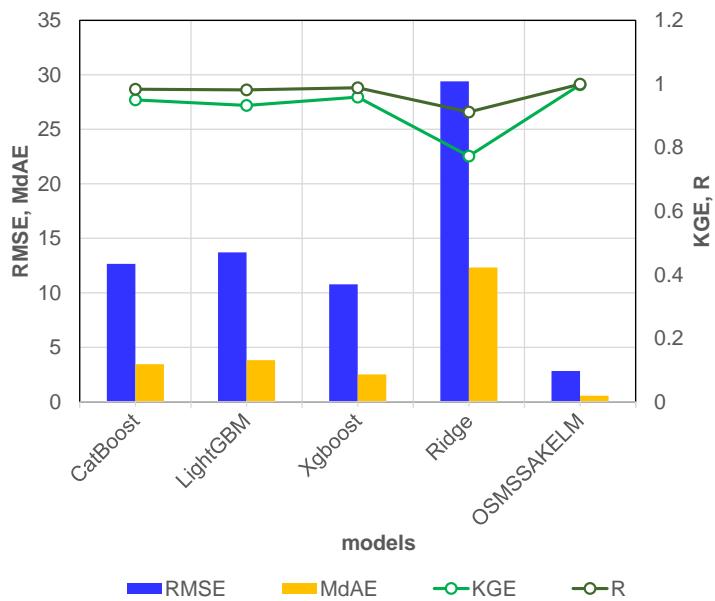
**Figure 18.** Percent (%) of improvement of the WD-OSMSSAKELM versus the values of other methods for the metrics during the testing phase.

Figure 19 exposes the scatter plot on the observed PM2.5 data against the predicted PM2.5 data for the training and testing stages of the WD-OSMSSAKELM and other regression methods. These plots visually show the relationship between the predicted PM2.5 scores and the prediction errors. The plots substantiate the higher correlation of the results for the WD-OSMSSAKELM, which is a strong relationship. As we can observe, the scatter plot for the WD-OSMSSAKELM verifies the better regression performance compared to the other prediction methods. We can see that for WD-Ridge, WD-LightGBM, WD-Xgboost, and WD-CatBoost, the proportion of error was located among  $\pm 40\%$ , while the proposed WD-OSMSSAKELM model was inside the range of  $\pm 20\%$ .

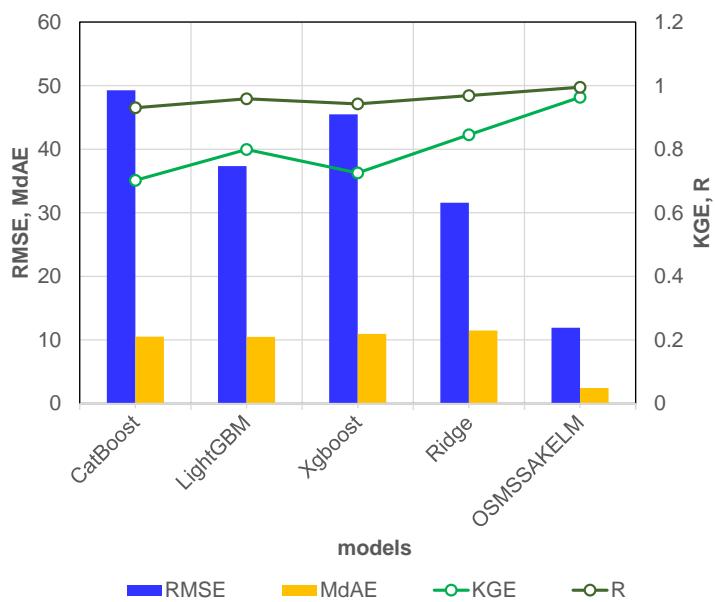


**Figure 19.** Scatter plots of different ML models.

Figures 20 and 21 compare the proposed model for the training and testing phases based on the RMSE, MdAE, KGE, and R metrics versus the other studied models. In both the testing and training stages, these plots demonstrate that the WD-OSMSSAKELM model delivered high efficiency, accuracy, and practicality. They visually confirm that the error range was minimum and optimal, and the volatility was low, with better error results, which provide a powerful prediction effect.

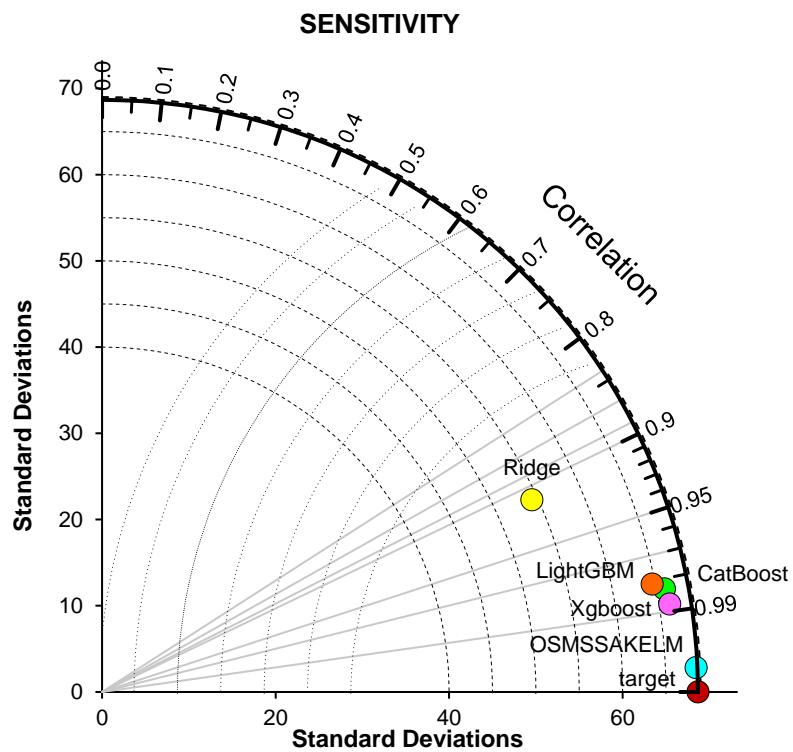


**Figure 20.** Comparison of the proposed model for the training phase versus other studied models based on the RMSE, MdAE, KGE, and R metrics

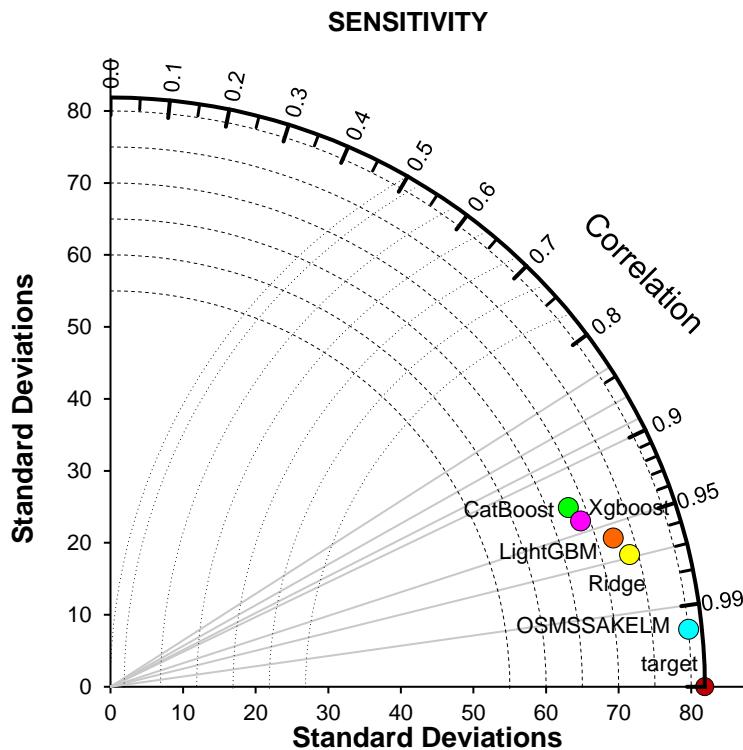


**Figure 21.** Comparison of the proposed model for the testing phase versus other studied models based on the RMSE, MdAE, KGE, and R metrics

Figures 22 and 23 demonstrate the Taylor diagrams of the training and testing stages for all optimized KELM models to investigate the results further. The correlation coefficient (R) was projected using the Taylor diagram to assess the inclusive performance of the models, with it specifying the models' efficiency. The plot exhibits a more perceptible and credible relationship between predicted and observed PM2.5 data as per the R and standard deviation. Consequently, compared to the other studied methods, the proposed WD-OSMSSAKELM had the fittest performance with reduced error results for PM2.5 prediction and was adjacent to the target point.

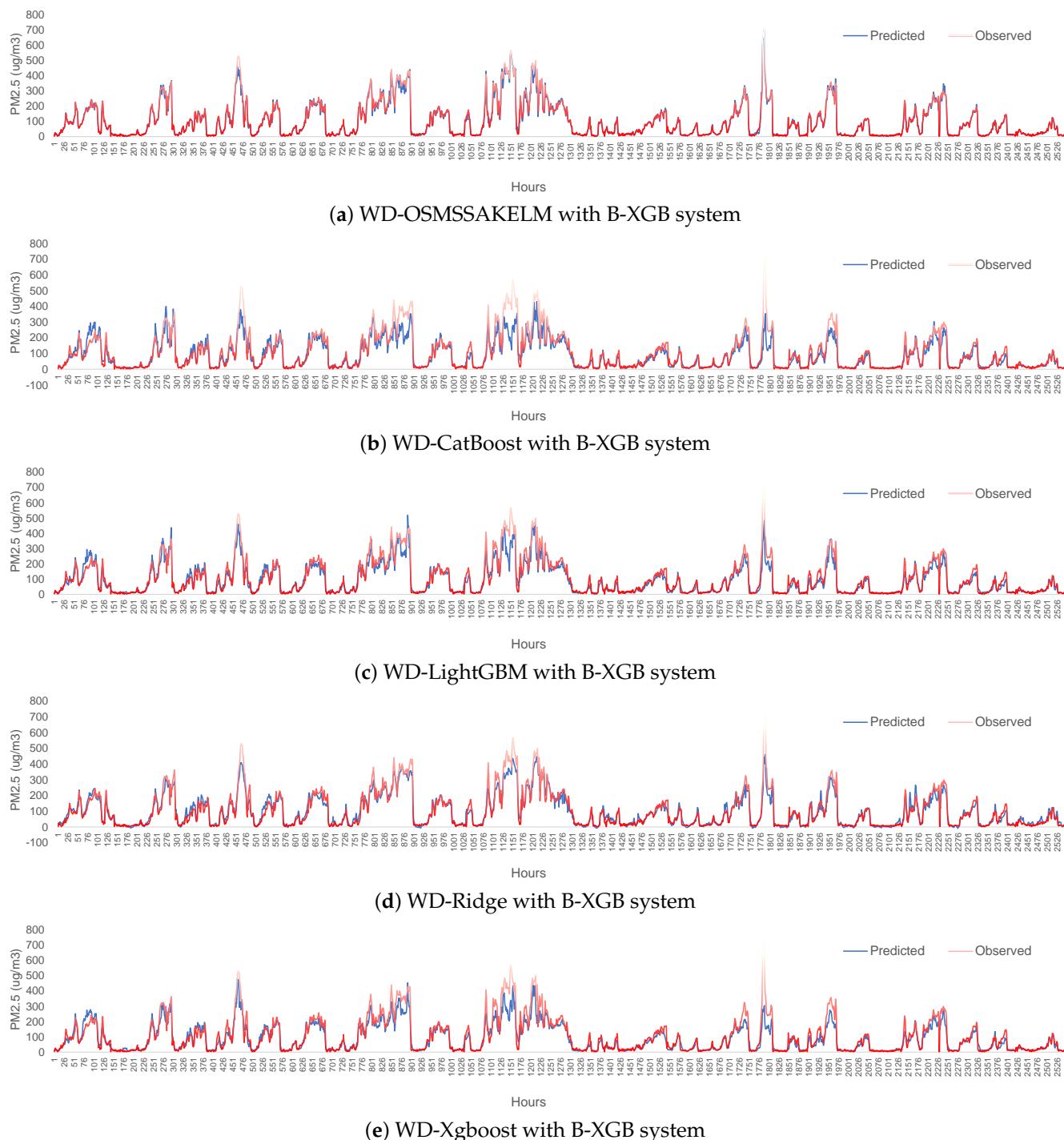


**Figure 22.** Taylor plot of the proposed model versus other regression methods for the training phase.



**Figure 23.** Taylor plot of the proposed model versus other regression methods for the testing phase.

Finally, the predicted time series of the models for the test stage are shown in Figure 24. The plots show that the error was minimal, and the fitting was the best compared to the other prediction models.



**Figure 24.** Comparison of the observed trend with the predicted time series (test results) of all models.

## 7. Conclusions

This paper proposed a new efficient wavelet PM2.5 prediction system based on an improved variant of the SSA (OSMSSA), wavelet decomposition, and Boruta-XGBoost (B-XGB) feature selection, which is called WD-OSMSSA-KELM. First, the B-XGB feature selection was applied to remove redundant features. Then, wavelet decomposition (WD) was applied to reach the multi-scale decomposition results and single-branch reconstruction of PM2.5 concentrations and alleviate the prediction error formed by time series data. Then, the new framework optimized the structure of the KELM model under each reconstructed component. To mitigate the premature performance of the SSA, a time-varying version of

the SSA with random leaders was proposed based on OBL and simplex-based search. The optimized model was utilized to predict the PM2.5 data, and 10 error metrics were applied to evaluate the model's performance and accuracy. The experimental results showed that the proposed WD-OLMSSA-KELM model ( $R: 0.995$ ,  $RMSE: 11.906$ ,  $MdAE: 2.424$ ,  $MAPE: 9.768$ ,  $KGE: 0.963$ ,  $R^2: 0.990$ ) can predict the PM2.5 data with superior performance compared to the WD-CatBoost, WD-LightGBM, WD-Xgboost, and WD-Ridge methods.

Despite all the advantages of the proposed model, we also have some limitations in the new system. One is that the user-defined values in the optimization core are chosen by the user and are not fully dynamic. The other limitation is the evolutionary nature of the KELM optimization, which may fall in local optima in some other datasets and needs further tuning and a bigger population size or iteration number. For future works, the proposed model can be extended with evolutionary-based feature selection models, and ensemble models will be investigated. Furthermore, we will compare the performance of the WD-OSMSSA-KELM prediction system with more new models and studies using new datasets in other cities in China. Another future work will be the extension of the proposed model using other variants of the KELM and a multi-objective variant of the OSMSSA.

**Author Contributions:** Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, resources, writing, writing—review and editing, visualization, A.A.H., M.A. and H.C.; supervision, M.A. and H.C.; project administration, M.A. and H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are available upon reasonable request from the corresponding author.

**Acknowledgments:** The first author acknowledges Jin Song Dong at the Department of Computer Science, School of Computing, National University of Singapore (NUS), for the administrative support during his one-year research internship at NUS.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Wen, H.; Dang, Y.; Li, L. Short-Term PM2.5 concentration prediction by combining GNSS and meteorological factors. *IEEE Access* **2020**, *8*, 115202–115216. [[CrossRef](#)]
- Shakoor, A.; Chen, X.; Farooq, T.H.; Shahzad, U.; Ashraf, F.; Rehman, A.; e Sahar, N.; Yan, W. Fluctuations in environmental pollutants and air quality during the lockdown in the USA and China: Two sides of COVID-19 pandemic. *Air Qual. Atmos. Health* **2020**, *13*, 1335–1342. [[CrossRef](#)] [[PubMed](#)]
- Pui, D.Y.; Chen, S.C.; Zuo, Z. PM2.5 in China: Measurements, sources, visibility and health effects, and mitigation. *Particuology* **2014**, *13*, 1–26. [[CrossRef](#)]
- Lin, Y.; Zou, J.; Yang, W.; Li, C.Q. A review of recent advances in research on PM2.5 in China. *Int. J. Environ. Res. Public Health* **2018**, *15*, 438. [[CrossRef](#)]
- Feng, S.; Gao, D.; Liao, F.; Zhou, F.; Wang, X. The health effects of ambient PM2.5 and potential mechanisms. *Ecotoxicol. Environ. Saf.* **2016**, *128*, 67–74. [[CrossRef](#)]
- Wang, D.; Wei, S.; Luo, H.; Yue, C.; Grunder, O. A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine. *Sci. Total. Environ.* **2017**, *580*, 719–733. [[CrossRef](#)]
- Wang, J.; Niu, T.; Wang, R. Research and application of an air quality early warning system based on a modified least squares support vector machine and a cloud model. *Int. J. Environ. Res. Public Health* **2017**, *14*, 249. [[CrossRef](#)]
- Zhang, S.; Li, X.; Li, Y.; Mei, J. Prediction of urban PM2.5 concentration based on wavelet neural network. In Proceedings of the 2018 IEEE Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 5514–5519.
- Anchan, A.; Shedthi, B.S.; Manasa, G.R. Models Predicting PM2.5 Concentrations—A Review. In *Recent Advances in Artificial Intelligence and Data Engineering*; Shetty, D.P., Shetty, S., Eds.; Springer: Singapore, 2022; pp. 65–83.
- Li, G.; Chen, L.; Yang, H. Prediction of PM2.5 concentration based on improved secondary decomposition and CSA-KELM. *Atmos. Pollut. Res.* **2022**, *13*, 101455. [[CrossRef](#)]
- Vijayaraghavan, K.; Cho, S.; Morris, R.; Spink, D.; Jung, J.; Pauls, R.; Duffett, K. Photochemical model evaluation of the ground-level ozone impacts on ambient air quality and vegetation health in the Alberta oil sands region: Using present and future emission scenarios. *Atmos. Environ.* **2016**, *141*, 209–218. [[CrossRef](#)]

12. Djalalova, I.; Delle Monache, L.; Wilczak, J. PM2.5 analog forecast and Kalman filter post-processing for the Community Multiscale Air Quality (CMAQ) model. *Atmos. Environ.* **2015**, *108*, 76–87. [[CrossRef](#)]
13. Dastoorpoor, M.; Idani, E.; Goudarzi, G.; Khanjani, N. Acute effects of air pollution on spontaneous abortion, premature delivery, and stillbirth in Ahvaz, Iran: A time series study. *Environ. Sci. Pollut. Res.* **2018**, *25*, 5447–5458. [[CrossRef](#)]
14. Song, Y.; Qin, S.; Qu, J.; Liu, F. The forecasting research of early warning systems for atmospheric pollutants: A case in Yangtze River Delta region. *Atmos. Environ.* **2015**, *118*, 58–69. [[CrossRef](#)]
15. Cheng, Y.; Zhang, H.; Liu, Z.; Chen, L.; Wang, P. Hybrid algorithm for short-term forecasting of PM2.5 in China. *Atmos. Environ.* **2019**, *200*, 264–279. [[CrossRef](#)]
16. Wang, J.; Wang, R.; Li, Z. A combined forecasting system based on multi-objective optimization and feature extraction strategy for hourly PM2.5 concentration. *Appl. Soft Comput.* **2022**, *114*, 108034. [[CrossRef](#)]
17. Banga, A.; Ahuja, R.; Sharma, S.C. Performance analysis of regression algorithms and feature selection techniques to predict PM2.5 in smart cities. *Int. J. Syst. Assur. Eng. Manag.* **2021**, *1*–14. [[CrossRef](#)]
18. Kok, I.; Guzel, M.; Ozdemir, S. Recent trends in air quality prediction: An artificial intelligence perspective. In *Intelligent Environmental Data Monitoring for Pollution Management*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 195–221.
19. Li, J.; Li, M. Prediction of ultra-short-term wind power based on BBO-KELM method. *J. Renew. Sustain. Energy* **2019**, *11*, 056104. [[CrossRef](#)]
20. Luo, H.; Wang, D.; Yue, C.; Liu, Y.; Guo, H. Research and application of a novel hybrid decomposition-ensemble learning paradigm with error correction for daily PM10 forecasting. *Atmos. Res.* **2018**, *201*, 34–45. [[CrossRef](#)]
21. Zhu, S.; Lian, X.; Liu, H.; Hu, J.; Wang, Y.; Che, J. Daily air quality index forecasting with hybrid models: A case in China. *Environ. Pollut.* **2017**, *231*, 1232–1244. [[CrossRef](#)]
22. Niu, M.; Wang, Y.; Sun, S.; Li, Y. A novel hybrid decomposition-and-ensemble model based on CEEMD and GWO for short-term PM2.5 concentration forecasting. *Atmos. Environ.* **2016**, *134*, 168–180. [[CrossRef](#)]
23. Yang, H.; Zhao, J.; Li, G. A new hybrid prediction model of PM2.5 concentration based on secondary decomposition and optimized extreme learning machine. *Environ. Sci. Pollut. Res.* **2022**, *29*, 67214–67241. [[CrossRef](#)]
24. Liu, H.; Jin, K.; Duan, Z. Air PM2.5 concentration multi-step forecasting using a new hybrid modeling method: Comparing cases for four cities in China. *Atmos. Pollut. Res.* **2019**, *10*, 1588–1600. [[CrossRef](#)]
25. Sun, W.; Xu, Z. A novel hourly PM2.5 concentration prediction model based on feature selection, training set screening, and mode decomposition-reorganization. *Sustain. Cities Soc.* **2021**, *75*, 103348. [[CrossRef](#)]
26. Yin, S.; Liu, H.; Duan, Z. Hourly PM2.5 concentration multi-step forecasting method based on extreme learning machine, boosting algorithm and error correction model. *Digit. Signal Process.* **2021**, *118*, 103221. [[CrossRef](#)]
27. Liu, B.; Ye, S. Research on Seasonal PM2.5 Predication in Hangzhou City Based on SSA-ELM Model. In Proceedings of the 2021 IEEE 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), Guiyang, China, 23–25 July 2021; pp. 515–524.
28. Du, P.; Wang, J.; Hao, Y.; Niu, T.; Yang, W. A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2.5 and PM10 forecasting. *Appl. Soft Comput.* **2020**, *96*, 106620. [[CrossRef](#)]
29. Du, P.; Wang, J.; Yang, W.; Niu, T. A novel hybrid fine particulate matter (PM2.5) forecasting and its further application system: Case studies in China. *J. Forecast.* **2022**, *41*, 64–85. [[CrossRef](#)]
30. Jiang, F.; He, J.; Tian, T. A clustering-based ensemble approach with improved pigeon-inspired optimization and extreme learning machine for air quality prediction. *Appl. Soft Comput.* **2019**, *85*, 105827. [[CrossRef](#)]
31. Xing, G.; Zhao, E.I.; Zhang, C.; Wu, J. A Decomposition-Ensemble Approach with Denoising Strategy for PM2.5 Concentration Forecasting. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 5577041. [[CrossRef](#)]
32. Jiang, F.; Qiao, Y.; Jiang, X.; Tian, T. MultiStep Ahead Forecasting for Hourly PM10 and PM2.5 Based on Two-Stage Decomposition Embedded Sample Entropy and Group Teacher Optimization Algorithm. *Atmosphere* **2021**, *12*, 64. [[CrossRef](#)]
33. Liu, H.; Yin, S.; Chen, C.; Duan, Z. Data multi-scale decomposition strategies for air pollution forecasting: A comprehensive review. *J. Clean. Prod.* **2020**, *277*, 124023. [[CrossRef](#)]
34. Adam, S.P.; Alexandropoulos, S.A.N.; Pardalos, P.M.; Vrahatis, M.N. No free lunch theorem: A review. In *Optimization and Its Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 57–82.
35. Ren, H.; Li, J.; Chen, H.; Li, C. Adaptive Lévy-assisted salp swarm algorithm: Analysis and optimization case studies. *Math. Comput. Simul.* **2021**, *181*, 380–409. [[CrossRef](#)]
36. Çelik, E.; Öztürk, N.; Arya, Y. Advancement of the search process of salp swarm algorithm for global optimization problems. *Expert Syst. Appl.* **2021**, *182*, 115292. [[CrossRef](#)]
37. Aljarah, I.; Habib, M.; Faris, H.; Al-Madi, N.; Heidari, A.A.; Mafarja, M.; Elaziz, M.A.; Mirjalili, S. A dynamic locality multi-objective salp swarm algorithm for feature selection. *Comput. Ind. Eng.* **2020**, *147*, 106628. [[CrossRef](#)]
38. Salgotra, R.; Singh, U.; Singh, G.; Singh, S.; Gandomi, A.H. Application of mutation operators to salp swarm algorithm. *Expert Syst. Appl.* **2021**, *169*, 114368. [[CrossRef](#)]
39. Liu, Y.; Shi, Y.; Chen, H.; Heidari, A.A.; Gui, W.; Wang, M.; Chen, H.; Li, C. Chaos-assisted multi-population salp swarm algorithms: Framework and case studies. *Expert Syst. Appl.* **2021**, *168*, 114369. [[CrossRef](#)]
40. Tubishat, M.; Ja’afar, S.; Alsawatti, M.; Mirjalili, S.; Idris, N.; Ismail, M.A.; Omar, M.S. Dynamic Salp swarm algorithm for feature selection. *Expert Syst. Appl.* **2021**, *164*, 113873. [[CrossRef](#)]

41. Kansal, V.; Dhillon, J.S. Emended salp swarm algorithm for multiobjective electric power dispatch problem. *Appl. Soft Comput.* **2020**, *90*, 106172. [CrossRef]
42. Zhang, H.; Wang, Z.; Chen, W.; Heidari, A.A.; Wang, M.; Zhao, X.; Liang, G.; Chen, H.; Zhang, X. Ensemble mutation-driven salp swarm algorithm with restart mechanism: Framework and fundamental analysis. *Expert Syst. Appl.* **2021**, *165*, 113897. [CrossRef]
43. Elaziz, M.A.; Li, L.; Jayasena, K.P.N.; Xiong, S. Multiobjective big data optimization based on a hybrid salp swarm algorithm and differential evolution. *Appl. Math. Model.* **2020**, *80*, 929–943. [CrossRef]
44. Tu, Q.; Liu, Y.; Han, F.; Liu, X.; Xie, Y. Range-free localization using Reliable Anchor Pair Selection and Quantum-behaved Salp Swarm Algorithm for anisotropic Wireless Sensor Networks. *Ad Hoc Netw.* **2021**, *113*, 102406. [CrossRef]
45. Salgotra, R.; Singh, U.; Singh, S.; Singh, G.; Mittal, N. Self-adaptive salp swarm algorithm for engineering optimization problems. *Appl. Math. Model.* **2021**, *89*, 188–207. [CrossRef]
46. Ren, H.; Li, J.; Chen, H.; Li, C. Stability of salp swarm algorithm with random replacement and double adaptive weighting. *Appl. Math. Model.* **2021**, *95*, 503–523. [CrossRef]
47. Chouhan, N.; Bhatt, U.R.; Upadhyay, R. Weighted Salp Swarm and Salp Swarm Algorithms in FiWi access network: A new paradigm for ONU placement. *Opt. Fiber Technol.* **2021**, *63*, 102505. [CrossRef]
48. Wang, Z.; Ding, H.; Yang, Z.; Li, B.; Guan, Z.; Bao, L. Rank-driven salp swarm algorithm with orthogonal opposition-based learning for global optimization. *Appl. Intell.* **2022**, *52*, 7922–7964. [CrossRef]
49. Majhi, S.K.; Mishra, A.; Pradhan, R. A chaotic salp swarm algorithm based on quadratic integrate and fire neural model for function optimization. *Prog. Artif. Intell.* **2019**, *8*, 343–358. [CrossRef]
50. Neggaz, N.; Ewees, A.A.; Elaziz, M.A.; Mafarja, M. Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113103. [CrossRef]
51. Ewees, A.A.; Al-qaness, M.A.A.; Abd Elaziz, M. Enhanced salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times. *Appl. Math. Model.* **2021**, *94*, 285–305. [CrossRef]
52. Saafan, M.M.; El-Gendy, E.M. IWOSSA: An improved whale optimization salp swarm algorithm for solving optimization problems. *Expert Syst. Appl.* **2021**, *176*, 114901. [CrossRef]
53. Ibrahim, R.A.; Ewees, A.A.; Oliva, D.; Abd Elaziz, M.; Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3155–3169. [CrossRef]
54. Zhang, Y.; Liu, R.; Wang, X.; Chen, H.; Li, C. Boosted binary Harris hawks optimizer and feature selection. *Eng. Comput.* **2021**, *37*, 3741–3770. [CrossRef]
55. Zhang, S.; Guo, B.; Dong, A.; He, J.; Xu, Z.; Chen, S.X. Cautionary tales on air-quality improvement in Beijing. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2017**, *473*, 20170457. [CrossRef]
56. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B Cybern. A Publ. IEEE Syst. Man Cybern. Soc.* **2012**, *42*, 513–529. [CrossRef]
57. Yaseen, Z.M.; Faris, H.; Al-Ansari, N. Hybridized extreme learning machine model with salp swarm algorithm: A novel predictive model for hydrological application. *Complexity* **2020**, *2020*, 8206245. [CrossRef]
58. Heidari, A.A.; Abbaspour, R.A.; Chen, H. Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training. *Appl. Soft Comput.* **2019**, *81*, 105521. [CrossRef]
59. Wang, M.; Chen, H.; Li, H.; Cai, Z.; Zhao, X.; Tong, C.; Li, J.; Xu, X. Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction. *Eng. Appl. Artif. Intell.* **2017**, *63*, 54–68. [CrossRef]
60. Zhang, L.; Zhou, W.; Jiao, L. Wavelet support vector machine. *IEEE Trans. Syst. Man, Cybern. Part B* **2004**, *34*, 34–39. [CrossRef]
61. Debnath, L.; Shah, F.A. *Wavelet Transforms and Their Applications*; Springer: Berlin/Heidelberg, Germany, 2002.
62. Kursa, M.B.; Rudnicki, W.R. Feature selection with the Boruta package. *J. Stat. Softw.* **2010**, *36*, 1–13. [CrossRef]
63. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
64. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’06), Vienna, Austria, 28–30 November 2005; Volume 1, pp. 695–701.
65. Nelder, J.A.; Mead, R. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308–313. [CrossRef]
66. Lagarias, J.C.; Reeds, J.A.; Wright, M.H.; Wright, P.E. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM J. Optim.* **1998**, *9*, 112–147. [CrossRef]
67. Abbassi, A.; Abbassi, R.; Heidari, A.A.; Oliva, D.; Chen, H.; Habib, A.; Jemli, M.; Wang, M. Parameters identification of photovoltaic cell models using enhanced exploratory salp chains-based approach. *Energy* **2020**, *198*, 117333. [CrossRef]
68. Płońska, A.; Płoński, P. MLJAR: State-of-the-Art Automated Machine Learning Framework for Tabular Data. Version 0.10.3. 2021. Available online: <https://github.com/mljar/mljar-supervised> (accessed on 22 June 2022).
69. Junninen, H.; Niska, H.; Tuppurainen, K.; Ruuskanen, J.; Kolehmainen, M. Methods for imputation of missing values in air quality data sets. *Atmos. Environ.* **2004**, *38*, 2895–2907. [CrossRef]
70. Tiyasha, T.; Tung, T.M.; Bhagat, S.K.; Tan, M.L.; Jawad, A.H.; Mohtar, W.H.M.W.; Yaseen, Z.M. Functionalization of remote sensing and on-site data for simulating surface water dissolved oxygen: Development of hybrid tree-based artificial intelligence models. *Mar. Pollut. Bull.* **2021**, *170*, 112639. [CrossRef] [PubMed]

71. Alsahaf, A.; Petkov, N.; Shenoy, V.; Azzopardi, G. A framework for feature selection through boosting. *Expert Syst. Appl.* **2022**, *187*, 115895. [[CrossRef](#)]
72. Ahmadianfar, I.; Shirvani-Hosseini, S.; He, J.; Samadi-Koucheksaraee, A.; Yaseen, Z.M. An improved adaptive neuro fuzzy inference system model using conjoined metaheuristic algorithms for electrical conductivity prediction. *Sci. Rep.* **2022**, *12*, 1–34. [[CrossRef](#)]
73. Barzegar, R.; Asghari Moghaddam, A.; Adamowski, J.; Ozga-Zielinski, B. Multi-step water quality forecasting using a boosting ensemble multi-wavelet extreme learning machine model. *Stoch. Environ. Res. Risk Assess.* **2018**, *32*, 799–813. [[CrossRef](#)]
74. Yang, X.S.; Hossein Gandomi, A. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* **2012**, *29*, 464–483. [[CrossRef](#)]
75. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
76. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
77. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98.
78. Taylor, K.E. Summarizing multiple aspects of model performance in a single diagram. *J. Geophys. Res. Atmos.* **2001**, *106*, 7183–7192. [[CrossRef](#)]
79. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. *J. Big Data* **2020**, *7*, 1–45. [[CrossRef](#)]
80. Fan, J.; Ma, X.; Wu, L.; Zhang, F.; Yu, X.; Zeng, W. Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. *Agric. Water Manag.* **2019**, *225*, 105758. [[CrossRef](#)]
81. Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K. Xgboost: Extreme Gradient Boosting. R Package Version 0.4-2. 2015; pp. 1–4. Available online: <https://cran.r-project.org/web/packages/xgboost/vignettes/xgboost.pdf> (accessed on 22 June 2022).
82. McDonald, G.C. Ridge regression. *Wiley Interdiscip. Rev. Comput. Stat.* **2009**, *1*, 93–100. [[CrossRef](#)]