IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Wind Speed Ensemble Forecasting Based on Deep Learning Using Adaptive Dynamic Optimization Algorithm

ABDELHAMEED IBRAHIM[1] (Member, IEEE), SEYEDALI MIRJALILI[2,3] (Senior Member, IEEE), M. EL-SAID[4,5], SHERIF. S. M. GHONEIM[6] (Senior Member, IEEE), MOSLEH ALHARTHI[6], T. F. IBRAHIM[7,8] and EL-SAYED M. EL-KENAWY[9] (Member, IEEE)

[1]Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt
[2]Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, QLD 4006, Australia
[3]YFL (Yonsei Frontier Lab), Yonsei University, Seoul, Korea
[4]Electrical Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt
[5]Dean of Delta Higher Institute of Engineering and Technology (DHIET), Mansoura, 35111, Egypt
[6]Electrical Engineering Department, College of Engineering, Taif University, Taif 21944, Saudi Arabia
[7]Department of Mathematics, Faculty of Sciences and arts (Mahayel), King Khalid University, Abha , Saudi Arabia
[8]Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt.
[9]Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology (DHIET), Mansoura, 35111, Egypt

Corresponding authors: Abdelhameed Ibrahim (e-mail: afai79@mans.edu.eg) and El-Sayed M. El-kenawy (e-mail: skenawy@ieee.org)

ABSTRACT The development and deployment of an effective wind speed forecasting technology can improve the stability and safety of power systems with significant wind penetration. However, due to the wind's unpredictable and unstable qualities, accurate forecasting of wind speed and power is extremely challenging. Several algorithms were proposed for this purpose to improve the level of forecasting reliability. A common method for making predictions based on time series data is the long short-term memory (LSTM) network. This paper proposed a machine learning algorithm, called adaptive dynamic particle swarm algorithm (AD-PSO) combined with guided whale optimization algorithm (Guided WOA), for wind speed ensemble forecasting. The proposed AD-PSO-Guided WOA algorithm selects the optimal hyper-parameters value of the LSTM deep learning model for forecasting purposes of wind speed. In experiments, a wind power forecasting dataset is employed to predict hourly power generation up to forty-eight hours ahead at seven wind farms. This case study is taken from the Kaggle Global Energy Forecasting Competition 2012 in wind forecasting. The results demonstrated that the AD-PSO-Guided WOA algorithm provides high accuracy and outperforms a number of comparative optimization and deep learning algorithms. Different tests' statistical analysis, including Wilcoxon's rank-sum and one-way analysis of variance (ANOVA), confirms the accuracy of proposed algorithm.

INDEX TERMS Artificial intelligence, Machine learning, Optimization, Forecasting, Guided whale optimization algorithm

I. INTRODUCTION

Due to the intermittence and unpredictability of wind power, the increasing penetration of wind power into power grids might significantly impact the safe functioning of power systems and power quality because the amount of wind energy generated is proportional to the wind speed. As a result, the development and deployment of an effective wind speed forecasting technology can improve the safety and stability of power systems with significant wind penetration. Wind energy is one of the essential low-carbon energy technologies. It can deliver a long-term energy supply and serves as a core component for micro-grids as part of intelligent grid architecture [1].

However, wind power generation is stochastic and intermittent, posing several hurdles to its widespread adoption. With the aid of wind speed and power generation projections, it is possible to reduce energy balancing and make power generating scheduling and dispatch choices. In addition, forecasts can reduce costs involved by mini missing the demand for wind curtailments and, as a result, enhancing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

income in power market operations. Due to the wind's unpredictable and unstable qualities, however, accurate forecasting wind speed and power is extremely difficult. A wind power forecast predicts the projected output of one or more wind turbines, often known as a wind farm. When one talks about production, it usually refers to the amount of power that a wind farm can generate (with unit's kW or MW depending on the nominal capacity of the wind farm). By combining power production throughout each period, forecasts may also be stated in energy [2].

Offer essential information about the projected wind speed and power over the next several minutes, hours, or days is the primary purpose of forecasting wind speed and power. The prediction can be separated based on power system operation requirements into four distinct time frames: long-term (from one day to seven days), medium-term (from six hours to twenty-four hours), short-term (from thirty minutes to six hours), and extremely short-term (from few seconds to thirty minutes. Turbine control and load tracking are based on very short-term estimates. Preload sharing is based on the short-term forecast. The medium-term projections are used for power system management and energy trading. Maintenance schedules for wind turbines are based on long-term forecasts [3].

Wind speed is considered a non-linear and time-relevant forecasting problem. This encourages researchers to make use of the knowledge included in the wind's historical data. Based on time-series data, one of the common methods for making predictions is the long short-term memory (LSTM) network [4]. Marcos et al. in [5] addressed the problem of wind power forecasting based on statistical and numerical weather prediction model models. Two different areas in Brazil were Brazilian developments on the regional atmospheric modeling system is employed to simulate forecasts of seventy-two hours ahead of the wind speed, at every ten minutes.

Liu et al. in [6] employed backpropagation neural network (BPNN), least squares support vector machine (LSSVM), and radial basis function NN (RBFNN) methods to forecast a sixteen MW wind farm that is located in Sichuan, China, based on two months data size at fifteen minutes sampling rate. Recently, Lin et al. in [3] applied Isolation Forest (IF) and deep learning NN for SCADA data of a wind turbine in Scotland to address the problem of wind power forecasting based on data size of twelve months and one-second sampling rate. Another method based on IF and feed-forward NN is applied to a seven MW wind turbine in Scotland (ORE Catapult) using a data size of twelve months and a one-second sampling rate.

To capture the wind speed data's unsupervised temporal features, an interval probability distribution learning (IPDL) model based on rough set theory and restricted Boltzmann machines were proposed in [7]. The IPDL model had a set of interval latent variables which can be tuned to capture the wind speed time series data's probability distribution. A real-valued interval deep belief network (IDBN) was also designed based on a fuzzy type II inference system and IPDL model for the future wind speed values' supervised regression. M. Khodayar et al. [8] proposed a deep neural network (DNN) architecture based on stacked auto-encoder (SAE) and stacked denoising auto-encoder (SDAE) for wind speed forecasting based on short-term and ultra-short-term. The auto-encoders (AEs) are used by the authors in [8] for the unsupervised feature learning from the unlabeled wind data. In addition, a supervised regression layer was employed for wind speed forecasting at the top of the AEs.

A scalable graph convolutional deep learning architecture (GCDLA) was proposed in [9] to learn the powerful Spatio-temporal features from the wind speed and direction data in the neighboring wind farms. GCDLA leveraged the extracted temporal features to forecast the whole graph nodes' wind-speed time series. The rough set theory was incorporated with the GCDLA by introducing lower and upper bound parameter approximations in their model. Authors in [10] proposed a framework based on an enhanced grasshopper optimization algorithm to optimize the hyperparameters and architecture of the LSTM deep learning model for wind speed forecasting. Table 1 shows the recent wind power prediction methods.

Hybrid machine intelligence techniques were proposed recently in the literature for wind forecasting based on different models. Authors in [11] utilized various variants of Support Vector Regression (SVR) and wavelet transform to forecast short-term wind speed. They evaluated their proposed techniques using various performance indices to get the best regressor for wind forecasting applications. A hybrid technique was presented in [12] using learning algorithms such as Twin SVR (TSVR), Convolutional neural networks (CNN), and random forest, in addition to, discrete wavelet transform (DWT) for wind forecasting. The extracted features from wind speed in their work were enhanced based on the wavelet transform. Another hybrid technique was proposed for the anomaly detection problem for wind turbine gearbox in [13] using adaptive threshold and twin SVM (TWSVM) methods.

In this work, a dataset of wind power forecasting is tested as a case study from Kaggle Global Energy Forecasting Competition 2012-Wind Forecasting to predict hourly power generation up to forty-eight hours ahead at seven wind farms. A proposed adaptive dynamic particle swarm algorithm (AD-PSO) with a guided whale optimization algorithm (Guided WOA) improves the forecasting performance by enhancing the parameters of the LSTM classification method. The proposed AD-PSO-Guided WOA algorithm selects the optimal hyper-parameters value of the LSTM deep learning model for forecasting purposes of wind speed. A binary-based algorithm of the AD-PSO-Guided WOA algorithm is used first for the feature selection problem from the wind power forecasting dataset. The evaluation of the binary AD-PSO-Guided WOA algorithm is presented in compared with Particle Swarm Optimization (PSO) [18], Grey Wolf Optimizer (GWO) [19], Stochastic Fractal Search (SFS) [20], WOA [21], [22], Genetic Algorithm (GA) [23], and Firefly

TABLE 1: Recent wind power prediction methods

Ref.	Algorithm	Datasets	Data Size	Sampling rate
Marcos et al. 2017 [5]	Kalman filter, Statistical regression	Palmas and RN05 wind farms in Brazil	7 and 12 months	10 min
Liu et al., 2017 [6]	BPNN, RBFNN and LSSVM	Sixteen MW wind farm in Sichuan, China	2 months	15 min
Bilal et al., 2018 [14]	MLP	Four sites in Senegal.	6-9 months	1 and 10 min
Wang et al., 2018 [15]	ELM optimised by MODA	Two sites of observation in Penglai, China	37 days	10 min
Hong et al. 2019 [16]	CNN, RBFNN, DGF	Historical power data of a wind farm in Taiwan	12 months	60 min
Zhang et al., 2019 [17]	LSTM, Gaussian Mixture Model (GMM)	A 123 units wind farm in north China	3 months	15 min
Khodayar et al., 2019 [7]	IPDL, IDBN, Boltzmann Machines, Rough Set	A wind site in Colorado, US	3 years	10 min
Khodayar et al., 2019 [9]	GCDLA, LSTM, Rough Set	145 wind sites in Northern States, US	6 years	5 min
Lin et al., 2020 [3]	Isolation Forest (IF), Deep learning NN	A wind turbine SCADA data in Scotland	12 months	1 s
	IF, feed-forward NN	Seven MW wind turbine in Scotland	12 months	1 s
Jalali et al., 2021 [10]	LSTM, Enhanced grasshopper optimization algorithm	Two wind stations in Las Vegas and Denver, US	12 months	30 min

Algorithm (FA) [24]. The optimized ensemble method based on the proposed algorithm is tested on the dataset. The results of this scenario are compared with Neural Networks (NN), Random Forest (RF), LSTM, Average ensemble, and k-Nearest Neighbors (k-NN) ensemble-based methods.

The AD-PSO-Guided WOA algorithm ensemble model is compared with the state-of-the-art optimization techniques including PSO [18],WOA [22], GA [23], GWO [19], Harris Hawks Optimization (HHO) [25], [26], Marine Predators Algorithm (MPA) [27], Chimp Optimization Algorithm (ChOA) [28], and Slime Mould Algorithm (SMA) [29]. The AD-PSO-Guided WOA algorithm ensemble model is also compared with the state-of-the-art deep learning techniques including Time delay neural network (TDNN) [30], Deep Neural Networks (DNN) [31], Stacked Denoising Autoencoder (SAE) [32], and Bidirectional Recurrent Neural Networks (BRNN) [33]. The statistical analysis of different tests is performed to confirm the accuracy of the algorithm, including Wilcoxon's rank-sum and one-way analysis of variance (ANOVA). This paper's contributions are summarized as follows.

- An adaptive dynamic PSO with guided WOA algorithm (AD-PSO-Guided WOA) is suggested.
- For feature selection problem from the wind power forecasting dataset, a binary AD-PSO-Guided WOA, version of the proposed algorithm, is tested.
- A one-sample t-test and ANOVA tests are used to test the binary AD-PSO-Guided WOA algorithm's statistical difference.
- To improve the wind power forecasting accuracy, an optimized ensemble method using the AD-PSO-Guided WOA algorithm is proposed.
- Wilcoxon's rank-sum and ANOVA tests are used to test the proposed optimizing ensemble method's statistical difference.
- The current work's importance is applying a new optimization algorithm to enhance LSTM classifier parameters.
- The proposed algorithms can be generalized and tested for other datasets.

II. PRELIMINARIES

A. MACHINE LEARNING

1) Neural Networks (NNs)

Artificial neural networks (ANNs) are a type of prediction model and classification approach. ANN is used to simulate complicated relationships of finding data patterns or cause-and-effect variable sets. Transient detection, approximation, time-series prediction, and pattern recognition are just a few of the disciplines they may use. ANN is considered an information processing pattern that functions similarly to the human brain. This information processing system comprises highly linked processing pieces called neurons that work together to solve issues in tandem. When formulating an algorithmic solution, a neural network comes in handy and where it is necessary to extract the structure from existing data [34].

A Multilayer perceptron (MLP) has three layers: input, output and one hidden layer. The weighted sum for the node output value is computed as follows [35].

$$S_j = \sum_{i=1}^{n} w_{ij} I_i + \beta_j \tag{1}$$

where input variable i is indicated as I_i. Connection weight between neuron j and I_i is represented as w_{ij}. β_j represents a bias value. Based on using of the sigmoid activation function, the node j output can be calculated as

$$f_j(S_j) = \frac{1}{1 + e^{-S_j}} \tag{2}$$

where the value of $f_j(S_j)$ is then used to get the network output as follows.

$$y_k = \sum_{j=1}^{m} w_{jk} f_j(S_j) + \beta_k \tag{3}$$

where the weights between output node k and neuron j in the hidden layer is defined as w_{jk} and β_k indicates the output layer bias value.

2) Random Forest (RF)

As a method based on statistical learning theory, random forests provide several advantages, including fewer configurable parameters, higher prediction precision, and improved generalization ability. It extracts numerous samples from the original sample using the bootstrap sampling approach,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

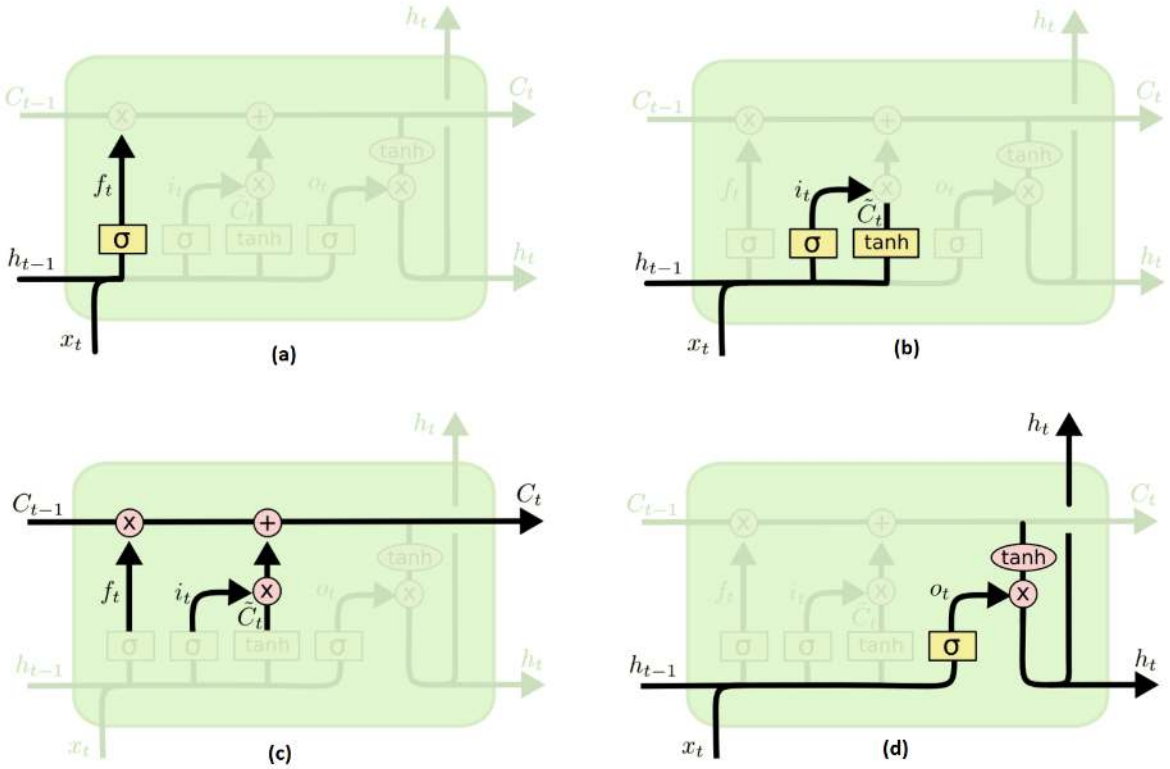

FIGURE 1: LSTM Neural Network Architecture

builds decision tree modeling based on each bootstrap sample, combines the predictions of multiple decision trees, and uses a voting mechanism to determine the outcome.

For the RF training algorithm, the regression/classification tree f_b is trained based on X_b and Y_b training examples for $X = x_1, \ldots, x_n$ and $Y = y_1, \ldots, y_n$. For B times, let $b = 1, ..., B$. After the process of training, the unseen samples predictions x' is calculated by averaging all the predictions of individual regression trees on x' as in equation 4.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x') \tag{4}$$

3) k-Nearest Neighbors (k-NN)

The model's interpretability. The findings of the prediction algorithm using the k-nearest neighbor's technique are based on the previous events that are the most like the current state based on a given distance metric. A simple average of the output values of the k nearest neighbors, or any weighted averaging, is used to make predictions. Thus, experts can analyze the findings of the k-nearest neighbor's method. The object's predictable variable in the k-NN numerical prediction this number is the average of its k closest neighbors' values. The k-NN method is one of the basic and the most powerful machine learning algorithms.

The k-NN model employs a similarity measure, Euclidean distance, to compare the data. Between x_{train} as training data and x_{test} as testing data, calculations of the Euclidean distance are based on the following equation.

$$D\left(x_{train,i}, x_{test,i}\right) = \sqrt{\sum_{i=1}^{k} (x_{train,i} - x_{test,i})^2} \tag{5}$$

To predict the output variables, k-NN determines k training data close to testing data. For unknown testing data to be predicted, the k training data output value is determined to be the nearest neighbours. The following formula is applied for predicting the testing data.

$$\hat{y} = \sum_{j=1}^{k} w_j y_j \tag{6}$$

where the jth neighbor weight is indicated as w_j and it is adjusted by the observed data, for $w_j = j/n$, for n indicates number of training data. This model can be used as a k-NN time series model.

B. DEEP LEARNING

1) Long Short Term Memory (LSTM)

LSTM is an improvement over standard version of ANN, and it is considered as a Recurrent Neural Network (RNN) that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

can be applied for many problems [36]. The main feature of LSTM is to remember the information for a long period of time and it is more suitable for kind of problems where avoidance of long term dependency is required. LSTM architecture is shown in Figure 1. To decide what data should be discarded from the cell state is the the first step in LSTM. A sigmoid layer, named forget gate layer, is used for this as shown in equation 7.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{7}$$

The next step is about deciding the new data that should be stored in the cell state. The values that need an update are decided by an input gate layer, sigmoid layer, and a new candidate values' vector to be added to the generated state by tanh layer as shown in equations 8 and 9.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{8}$$

$$C'_t = \tanh(W_i[h_{t-1}, x_t] + b_i) \tag{9}$$

Now, the old cell state C_{t-1} is updated into the new cell state C_t by equation 10 using equations 7, 8, and 9.

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \tag{10}$$

Output decision based on the cell state is the final step. A sigmoid layer will help to decide about cell state parts that will be moved to output. After that cell state will use tanh will force values between [-1,1] and will multiply it with the output of the sigmoid gate as mentioned in equation 11.

$$h_t = o_t \times \tanh(C_t), o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{11}$$

C. ENSEMBLE TECHNIQUES

The goal such approaches is to combine the capabilities of a variety of single base models to create a predictive model. This concept can be implemented in a variety of ways. For instance, key strategies rely on resampling the training set, while others rely on alternative prediction methods or modifying some predictive technique parameters. Finally, the result of each prediction is combined using an ensemble of approaches.

III. PROPOSED ADAPTIVE DYNAMIC PSO-GUIDED WOA ALGORITHM

This section discusses the presented AD-PSO-Guided WOA algorithm using adaptive dynamic technique, particle swarm algorithm, and modified whale optimization algorithm. Algorithm (1) shows the AD-PSO-Guided WOA algorithm.

A. ADAPTIVE DYNAMIC TECHNIQUE

After the initialization of the optimization algorithm and for each solution in the population, a fitness value is evaluated. For the best fitness value, the optimization algorithm then gets the relevant best agent (solution). To start the adaptive dynamic process, the optimization algorithm starts to split agents of the population into two groups, as in Fig. 2. The two groups are named exploitation group and exploration group. The main target of the individuals (agents) in the exploitation group is to move toward the optimal solution, and the target of the agents in the exploration group is to search the area around the leaders. The change (update) between the agents of the population groups is dynamic. To achieve a balance between the exploitation group and exploration group, the optimization algorithm is initiated with a (50/50) population. Figure 3 explains the balancing and the dynamic change between the number of agents in the groups over different iteration until getting the best solution.

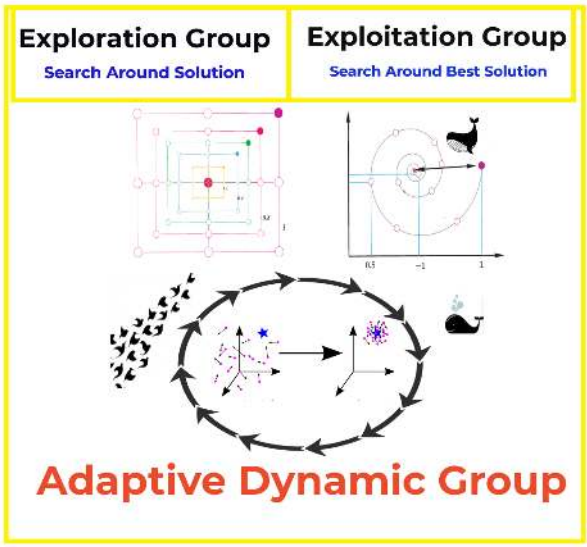

FIGURE 2: Balancing of exploration and exploitation groups in the AD-PSO-Guided WOA algorithm.

B. GUIDED WOA ALGORITHM

The WOA algorithm shows its advantages for different problems in the area of optimization. WOA is considered in the literature as one of the most effective optimization algorithms [20], [37]. However, it might suffer from a low capability of exploration. The foraging behaviour of whales in nature represents the WOA algorithm inspiration [38]. For mathematical calculations, let's consider n to be the dimension or number of variables of the search space that whales will swim in. If it is considered that the agents (solutions) positions in the space search will be updated over time, the best solution of food will be found.

The following equation can be used in the WOA algorithm for the purpose of updating agents' positions.

$$\overrightarrow{X}(t+1) = \overrightarrow{X}^*(t) - \overrightarrow{A}.\overrightarrow{D}, \overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X}^*(t) - \overrightarrow{X}(t)| \tag{12}$$

where $\overrightarrow{X}(t)$ term represents a solution at an iteration t. The $\overrightarrow{X}^*(t)$ term represents the food or the optimal solution

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

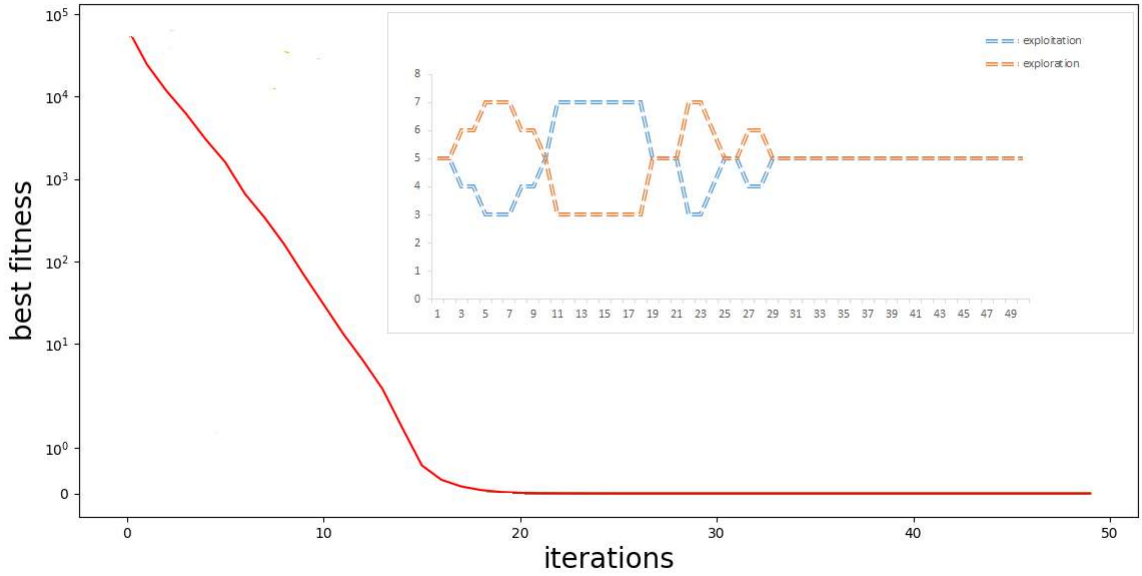

FIGURE 3: Dynamic updating of the exploration and exploitation groups in the AD-PSO-Guided WOA algorithm.

position. The "." indicated in this equation a kind of pairwise multiplication. The $\overrightarrow{X}(t+1)$ represents the changed agent position. The two vectors of $\overrightarrow{A}$ and $\overrightarrow{C}$ will be updated during iterations as $\overrightarrow{A} = 2\overrightarrow{a}.r_1 - \overrightarrow{a}$ and $\overrightarrow{C} = 2.r_2$. The $\overrightarrow{a}$ term will be changed linearly from 2 (maximum value) to 0 (minimum value). The values of r_1 and r_2 are changing randomly between $[0, 1]$.

The term Guided WOA, in this work, indicates a modified version of the original WOA algorithm [37]. In Guided WOA, the drawback of the original WOA is alleviated by updating the search strategy through one agent. The modified algorithm moves the agents toward the prey or best solution based on more than one agent. Equation 12 in the original WOA algorithm forces agents to move randomly around each other to get the global search. In the Guided WOA algorithm, however, the exploration process is enhanced by forcing agents to follow three random agents instead of one. For forcing agents not to be affected by one leader position to get more exploration, equation 12 can be replaced by the following one.

$$\begin{aligned}\overrightarrow{X}(t+1) =& \overrightarrow{w_1} * \overrightarrow{X}_{rand1} + \overrightarrow{z} * \overrightarrow{w_2} * (\overrightarrow{X}_{rand2} - \overrightarrow{X}_{rand3}) + \\ & (1 - \overrightarrow{z}) * \overrightarrow{w_3} * (\overrightarrow{X} - \overrightarrow{X}_{rand1})\end{aligned} \tag{13}$$

where the three random solutions are represented in this equation by $\overrightarrow{X}_{rand1}$, $\overrightarrow{X}_{rand2}$, and $\overrightarrow{X}_{rand3}$. The $\overrightarrow{w_1}$ term value is updated in $[0, 0.5]$. The terms of $\overrightarrow{w_2}$ and $\overrightarrow{w_3}$ are changing in $[0, 1]$. Finally to smoothly the change between exploration and exploitation, the term $\overrightarrow{z}$ is decreasing exponentially instead of linearly and is calculated as follows.

$$\overrightarrow{z} = 1 - \left(\frac{t}{Max_{iter}}\right)^2 \tag{14}$$

where iteration number is represented as t, and Max_{iter} represents the maximum number of iterations.

C. PARTICLE SWARM OPTIMIZATION

Unlike the WOA algorithm, the PSO algorithm simulates the social behaviour of a different kind of swarming pattern of flocks in nature such as birds [19]. The agents in the PSO algorithm search for the best solution or food according to the updated velocity by changing their positions. The algorithm uses particles (agents) and each agent follows these parameters:

- The term ($x^i \in R^n$) indicates a point or position in R^n search space. The agents' positions are calculated by a fitness function.
- the term (v^i) represents velocity or rate of change of agents positions,
- The term (p^i) indicates the last best positions of the particles.

The positions and velocities of agents are updating over iterations. The positions of agents changed using the following equation.

$$x^i_{(t+1)} = x^i_{(t)} + v^i_{(t+1)} \tag{15}$$

where the new agent position is indicated as x^i_{t+1}. Updated velocity of each agent v^i_{t+1} evaluated as in the following form.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

Algorithm 1 :The AD-PSO-Guided WOA algorithm

Initialize population $\overrightarrow{X}_i(i = 1, 2, ..., n)$ with size n, maximum iterations Max_{iter}, fitness function F_n.
Initialize parameters $\overrightarrow{a}$, $\overrightarrow{A}$, $\overrightarrow{C}$, l, $\overrightarrow{r_1}$, $\overrightarrow{r_2}$, $\overrightarrow{r_3}$, $\overrightarrow{w_1}$, $\overrightarrow{w_2}$, $\overrightarrow{w_3}$, $t = 1$
Evaluate fitness function F_n for each $\overrightarrow{X}_i$
Find best individual $\overrightarrow{X^*}$
while $t \leq Max_{iter}$ **do**
  if $(t\%2 == 0)$ **then**
    for $(i = 1 : i < n + 1)$ **do**
      if $(\overrightarrow{r_3} \leq 0.5)$ **then**
        if $(|\overrightarrow{A}| < 1)$ **then**
          Update current search agent position as $\overrightarrow{X}(t+1) = \overrightarrow{X^*}(t) - \overrightarrow{A}.\overrightarrow{D}$
        else
          Select three random search agents $\overrightarrow{X}_{rand1}$, $\overrightarrow{X}_{rand2}$, and $\overrightarrow{X}_{rand3}$
          Update $(\overrightarrow{z})$ by the exponential form of $\overrightarrow{z} = 1 - \left(\frac{t}{Max_{iter}}\right)^2$
          Update current search agent position as $\overrightarrow{X}(t+1) = \overrightarrow{w_1} * \overrightarrow{X}_{rand1} + \overrightarrow{z} * \overrightarrow{w_2} * (\overrightarrow{X}_{rand2} - \overrightarrow{X}_{rand3}) + (1 - \overrightarrow{z}) * \overrightarrow{w_3} * (\overrightarrow{X} - \overrightarrow{X}_{rand1})$
        end if
      else
        Update current search agent position as $\overrightarrow{X}(t+1) = \overrightarrow{D}'.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(t)$
      end if
    end for
    Calculate fitness function F_n for each $\overrightarrow{X}_i$ from Guided WOA
  else
    Calculate fitness function F_n for each $\overrightarrow{X}_i$ from PSO
  end if
  Update $\overrightarrow{a}$, $\overrightarrow{A}$, $\overrightarrow{C}$, l, $\overrightarrow{r_3}$
  Find best individual $\overrightarrow{X^*}$
  Set t = t + 1
end while
return $\overrightarrow{X^*}$

$$v^i_{(t+1)} = C_1 r_1(p^i_(t) - x^i_{(t)}) + C_2 r_2(G - x^i_{(t)}) + \omega v^i_{(t)} \quad (16)$$

where the term ω represents the inertia weight. The terms C_1 and C_2 indicate cognition and social learning factors. The G parameter represents the global best position and the values of r_1 and r_2 are within $[0; 1]$.

D. PROPOSED ALGORITHM COMPLEXITY ANALYSIS

The AD-PSO-Guided WOA algorithm' complexity analysis is presented in this section based on Algorithm (1). Using population number indicated as n iterations number as M_t, the complexity can be defined for each part of the algorithm as

- Initializing of the population: O (1).
- Initializing of parameters $\overrightarrow{a}$, $\overrightarrow{A}$, $\overrightarrow{C}$, l, $\overrightarrow{r_1}$, $\overrightarrow{r_2}$, $\overrightarrow{r_3}$, $\overrightarrow{w_1}$, $\overrightarrow{w_2}$, $\overrightarrow{w_3}$, $t = 1$: O (1).
- Evaluating fitness function F_n: O (n).
- Getting best individual $\overrightarrow{X^*}$: O (n).
- Updating positions: O $(M_t \times n)$.
- Evaluating agents' fitness function using Guided WOA: O $(M_t \times n)$.
- Evaluating agents' fitness function using PSO: O $(M_t \times n)$.
- Updating parameters $\overrightarrow{a}$, $\overrightarrow{A}$, $\overrightarrow{C}$, l, $\overrightarrow{r_3}$: O (M_t).
- Updating best solution: O $(M_t \times n)$.
- Increasing iteration counter: O (M_t).

The complexity of the AD-PSO-Guided WOA algorithm can be considered as O $(M_t \times n)$. For m variables problems, the algorithm complexity can be considered O $(M_t \times n \times m)$.

E. BINARY OPTIMIZER

For the feature selection problem, the output solution should be changed to a binary solution using 0 or 1. The sigmoid function is usually employed to change the continuous solution of the optimizer to a binary solution.

$$\overrightarrow{X}_d^{(t+1)} = \begin{cases} 0 & \text{if } Sigmoid(X_{Best}) < 0.5 \\ 1 & otherwise \end{cases}, \qquad Sigmoid(X_{Best}) = \frac{1}{1 + e^{-10(X_{Best} - 0.5)}} \quad (17)$$

where the best position is indicated as X_{Best} for t iteration. The $Sigmoid$ function is used to help in changing the continuous values to be 0 or 1. For $Sigmoid(X_{Best}) \geq 0.5$, the value will change to 1, otherwise, the value will be changed to be 0. Algorithm (2) shows the step by step explanation of the binary AD-PRS-Guided WOA Algorithm.

F. FITNESS FUNCTION

The solutions' quality of an optimizer is measured based on the assigned fitness function. The function is mainly depending on the error rate of classification/regression and the features that have been selected from the input dataset. The best solution is according to the set of features that can give a minimum number of features with a minimum classification error rate. The following equation is applied in this work for the evaluation of solutions' quality.

$$F_n = \alpha Err(O) + \beta \frac{|s|}{|f|} \quad (18)$$

where the optimizer error rate is indicated as $Err(O)$, the selected set of features is denoted as s, f represents total number of existing features. The $\alpha \in [0, 1], \beta = 1 - h_1$ values are responsible of the classification error rate and the number of selected features.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

Algorithm 2 :Binary AD-PSO-Guided WOA algorithm

Initialize AD-PSO-Guided WOA algorithm configuration, including population and parameters
Change current solutions to binary solution (0 or 1)
Evaluate fitness function and determine the best solution
Train k-NN based model and then calculate error
while $t \leq iters_{max}$ **do**
  Apply AD-PSO-Guided WOA algorithm
  Change updated solution to binary solution (0 or 1) based on equation 17
  Evaluate fitness function for each agent
  Update parameters
  Update best solution
end while
Return best solution

IV. EXPERIMENTAL RESULTS

The experimental settings and results for wind power forecasting problems using the presented AD-PSO-Guided WOA algorithm are presented in this section. The dataset is first discussed, and then the experiments are divided into feature selection, ensemble, and comparison scenarios.

A. DATASET DESCRIPTION

A wind power forecasting dataset to predict hourly power generation up to forty-eight hours ahead at seven wind farms is tested in the experiments as a case study. The dataset is published on Kaggle as Global Energy Forecasting Competition 2012 - Wind Forecasting [39]. The presented AD-PSO-Guided WOA algorithm is applied in different scenarios to test the best available accuracy compared to algorithms in the literature. A statistical analysis of different tests is also applied to the tested dataset to show the algorithm's accuracy. Prediction of regression is shown in Fig. 4. The figure shows the actual values from the dataset and the predicted values based on the proposed AD-PSO-Guided WOA algorithm.

B. FEATURE SELECTION SCENARIO

The experiment in this scenario desired to show the feature selection efficiency by the proposed binary AD-PSO-Guided WOA algorithm. The binary AD-PSO-Guided WOA algorithm performance is compared with the binary version of GWO (bGWO) [19], binary PSO (bPSO) [18], binary SFS (bSFS) [20], binary WOA (bWOA) [21], [22], binary FA (bFA) [24], and binary GA (bGA) [23] using performance metrics shown in Table 2. The variables in Table 2 are indicated as follows. An optimizer number of runs is indicated as M, the best solution at the run number j is represented by g_j^*, size of the g_j^* vector is indicated as $size(g_j^*)$, and the number of tested points is N. A classifier's output label for a point i is C_i, a class's label for a point i is L_i, the total number of features is D, and the $Match$ function is used for calculating the matching between two inputs. The metrics include average error and standard deviation fitness.

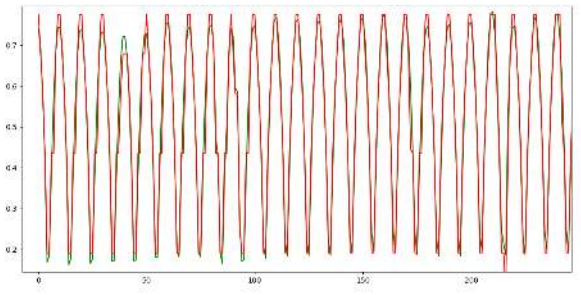

FIGURE 4: The actual (green color) and predicted (red color) values based on the proposed AD-PSO-Guided WOA algorithm.

TABLE 2: Feature selection performance metrics.

Metric	Value
Average Error	$1 - \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} Match(C_i, L_i)$
Average Select Size	$\frac{1}{M}\sum_{j=1}^{M}\frac{size(g_j^*)}{D}$
Average Fitness	$\frac{1}{M}\sum_{j=1}^{M} g_j^*$
Best Fitness	$Min_{j=1}^{M} g_j^*$
Worst Fitness	$Max_{j=1}^{M} g_j^*$
Standard Deviation	$\sqrt{\frac{1}{M-1}\sum(g_j^* - Mean)^2}$

TABLE 3: Configuration of AD-PSO-Guided WOA algorithm.

Parameter	Value
# Whales	20
# Iterations	20
# Runs	20
Dimension	# Features
Inertia W_{max}, W_{min}	[0.9,0.6]
Acceleration constants C_1, C_2	[2,2]
α of F_n	0.99
β of F_n	0.01

TABLE 4: Configuration of compared algorithms for feature selection.

Algorithm	Parameter (s)	Value (s)
GWO	a	2 to 0
	# Wolves	20
	# Iterations	20
PSO	Inertia W_{max}, W_{min}	[0.9,0.6]
	Acceleration constants C_1, C_2	[2,2]
	# Particles	20
	Generations	20
SFS	Maximum diffusion level	1
WOA	a	2 to 0
	r	[0,1]
	# Whales	20
	# Iterations	20
FA	# Fireflies	10
GA	Mutation ratio	0.1
	Crossover	0.9
	Selection mechanism	Roulette wheel
	Population size	20
	Generations	20

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

TABLE 5: Results of feature selection for the presented and compared binary algorithms

	AD-PSO-Guided WOA	bGWO	bPSO	bSFS	bWOA	bFA	bGA
Average error	0.4790	0.5047	0.5385	0.5481	0.5383	0.5369	0.5183
Average Select size	0.2320	0.4275	0.4275	0.5669	0.5909	0.4620	0.3699
Average Fitness	0.3510	0.3541	0.3525	0.3754	0.3603	0.4044	0.3655
Best Fitness	0.2521	0.2744	0.3328	0.2651	0.3244	0.3231	0.2688
Worst Fitness	0.3305	0.3413	0.4005	0.3667	0.4005	0.4207	0.3839
Standard deviation Fitness	0.1635	0.1679	0.1680	0.1742	0.1665	0.2011	0.1665

TABLE 6: Results of ANOVA test for feature selection of the presented and compared binary algorithms

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.07358	6	0.01226	F (6, 133) = 11.75	P < 0.0001
Residual (within columns)	0.1387	133	0.001043	-	-
Total	0.2123	139	-	-	-

TABLE 7: One sample t-test for feature selection of the presented and compared binary algorithms

	AD-PSO-Guided WOA	bGWO	bPSO	bSFS	bWOA	bFA	bGA
Theoretical mean	0	0	0	0	0	0	0
Actual mean	0.479	0.5022	0.5393	0.5481	0.5388	0.5352	0.519
# values	20	r20	20	20	20	20	20
One sample t-test							
t, df		t=54.41, df=19	t=72.46, df=19	t=188.9, df=19	t=210.3, df=19	t=40.08, df=19	t=92.86, df=19
P value (two tailed)		0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
P value summary		****	****	****	****	****	****
Significant (alpha=0.05)?		Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?							
Discrepancy		0.5022	0.5393	0.5481	0.5388	0.5352	0.519
Discrepancy SD		0.04128	0.03328	0.01298	0.01146	0.05972	0.02499
Discrepancy SEM		0.00923	0.007442	0.002902	0.002562	0.01335	0.005589
95% confidence interval		0.4829 to 0.5215	0.5237 to 0.5548	0.5420 to 0.5542	0.5334 to 0.5442	0.5073 to 0.5632	0.5073 to 0.5307
R squared		0.9936	0.9964	0.9995	0.9996	0.9883	0.9978

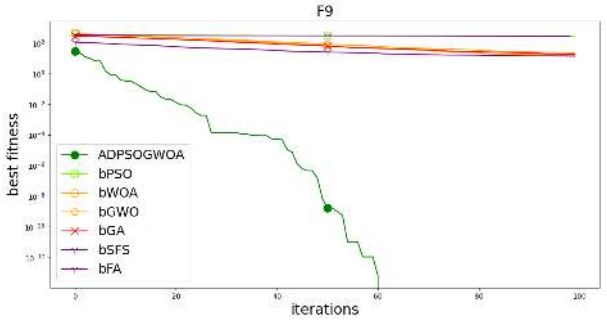

FIGURE 5: The AD-PSO-Guided WOA algorithm convergence curve compared to different algorithms

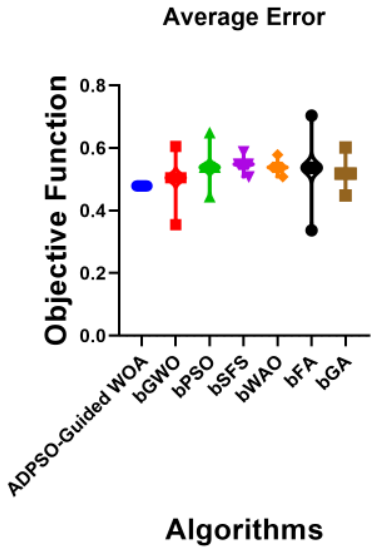

FIGURE 6: The AD-PSO-Guided WOA algorithm average error based on the objective function compared to different binary algorithms

The AD-PSO-Guided WOA algorithm configuration setting in experiments is shown in Table 3. The AD-PSO-Guided WOA algorithm's initial parameters are the number of population equal 20, the maximum number of iterations is set to 20, and the number of runs equals 20 for the dataset. The main parameters for the PSO algorithm are W_{max} and W_{min}, which their values are set to 0.9 and 0.6, respectively. In addition, the α parameter is assigned to be (0.99) and β is assigned to be $(1-\alpha)$. The GWO, PSO, SFS, WOA, FA, and GA algorithms' configuration is shown in Table 4.

In this scenario, Table 5 shows the results provided by GWO, PSO, SFS, WOA, FA, and GA algorithms. The AD-PSO-Guided WOA algorithm shows a minimum average error of (0.4790) for feature selection for the presented results. The AD-PSO-Guided WOA algorithm, based on the minimum error of the tested problem, is the best and the SFS algorithm is the worst. In terms of standard deviation, the AD-PSO-Guided WOA algorithm has the lowest value

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

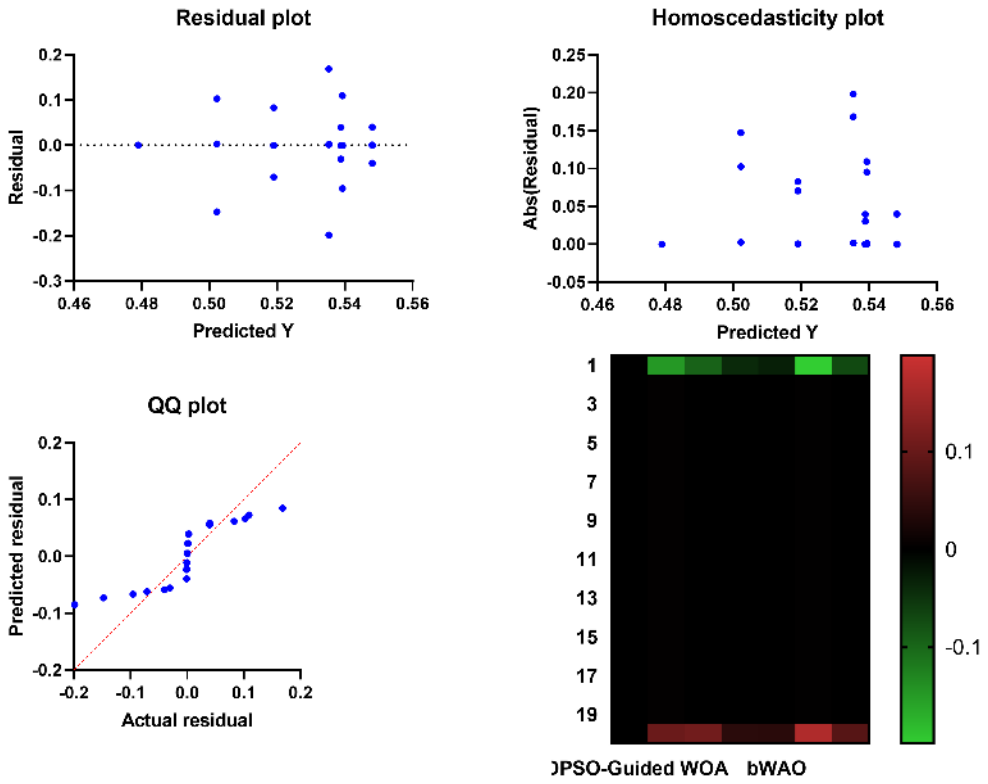

FIGURE 7: Residual, heteroscedasticity, QQ plots and heat map of the presented and compared algorithms for feature selection problem

of (0.1635) which indicates the algorithm's stability and robustness.

The convergence curve of the AD-PSO-Guided WOA algorithm compared to other algorithms is shown in Figure 5. The figure shows the optimizer exploitation capability and the algorithm's ability to avoid possible local optima. Figure 6 shows the AD-PSO-Guided WOA average error based on the objective function compared to different algorithms. The minimum, maximum, and average values for different binary algorithms indicate the advantages of the proposed algorithm. The p-values of the AD-PSO-Guided WOA algorithm are tested compared to GWO, PSO, SFS, WOA, FA, and GA algorithms by ANOVA and t-test tests in Tables 6 and 7, respectively. The statistical analysis results show the superiority and statistical significance of the proposed AD-PSO-Guided WOA algorithm.

The residual values and plots can be useful for some datasets that are not suitable candidates for feature selection. To achieve the ideal case, the residual values should be distributed uniformly around the horizontal axis. Considering that the sum and mean of the residuals are equal to zero, the residual value is computed as the difference between predicted and actual values. The residual plot is shown in Figure 7. A nonlinear and linear model is decided from the residual plot patterns and the appropriate one is determined. The heteroscedasticity plot is shown in Figure 7. Homoscedasticity describes if the error term is the same across the values of independent variables. Figure 7 also shows the quantile-quantile (QQ) plot, probability plot, and heat map. Since the distributions of points in the QQ plot are well fitted on the predetermined line, the actual and predicted residuals are considered to be linearly related. This confirms the presented AD-PSO-Guided WOA algorithm's performance.

C. ENSEMBLE FORECASTING SCENARIO

This scenario is formulated using ensemble-based models of the average ensemble, k-NN ensemble, and the proposed optimizing ensemble model based on the AD-PSO-Guided WOA algorithm. Some ensemble models utilize the training instances of the three base models of NN, RF, and LSTM. This can be used to forecast the unknown observations to the regression of the majority and gives the results to predict wind speed. The hyperparameters that are fed to the AD-PSO-Guided WOA algorithm to train the LSTM model are the number of epochs T_e, encoding length for each attention weights L_e, size of champion attention weights subset W_a, and size of attention weights set N_a.

The evaluation metrics that are used for the experiments in this scenario include Root Mean Squared Error (RMSE), Relative RMSE (RRMSE), Mean Absolute Error (MAE),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

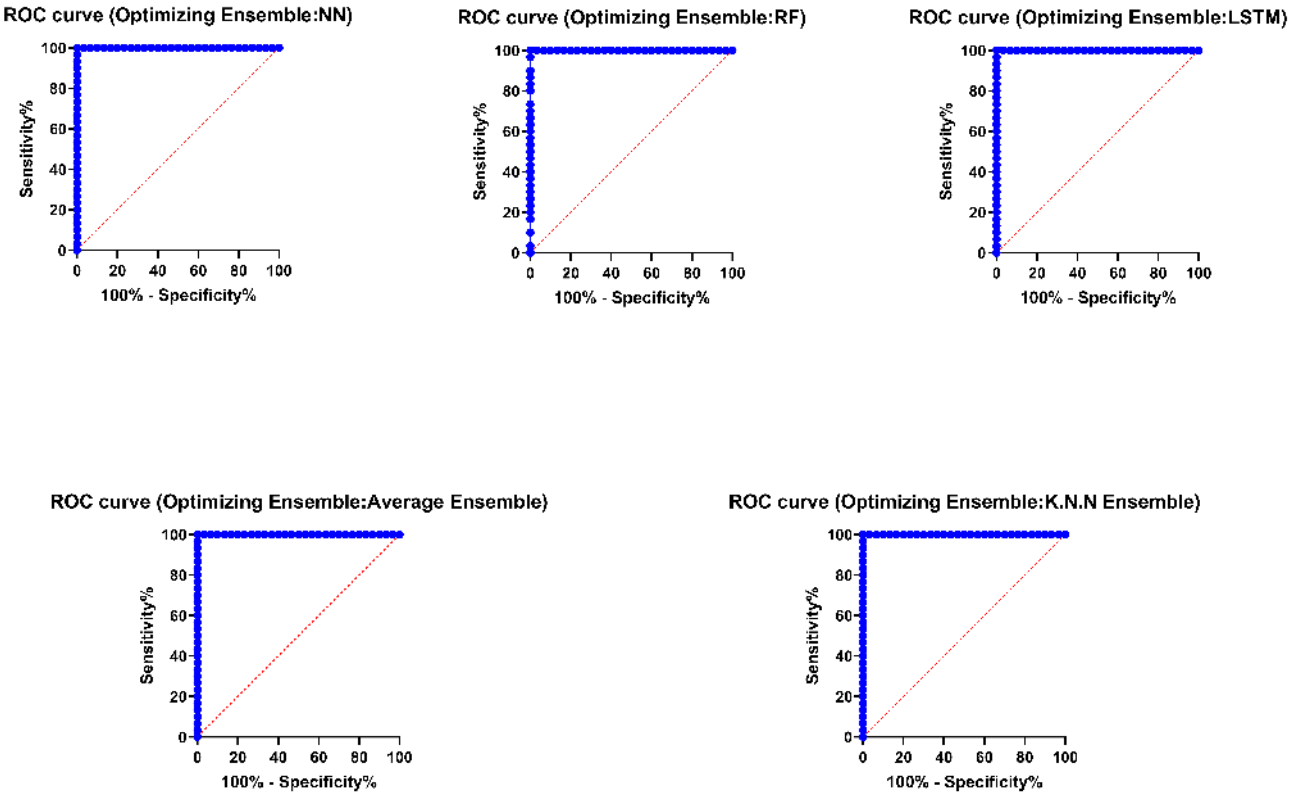

FIGURE 8: The ROC curves of the presented optimizing ensemble model versus other models

TABLE 8: Different ensemble-based and single models wind speed forecasting results

	NN	RF	LSTM	Average Ensemble	k-NN Ensemble	Optimizing Ensemble
MAPE (%)	12.8372	7.3846	6.9924	5.1134	3.6111	1.8755
MAE	0.4961	0.3163	0.2303	0.1013	0.0134	0.00476
RMSE	0.491329643	0.231488171	0.109544	0.091676794	0.01493511	0.003728832

Mean Absolute Percentage Error (MAPE), and the correlation coefficient (r) [40]. The RMSE metric can be calculated as follow to assess the performance.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(H_{p,i} - H_i)^2}{n}} \tag{19}$$

where $H_{p,i}$ represents a predicted value and H_i indicates the actual measured value. The n parameter represents the total number of values. The RRMSE metric is calculated as follow.

$$RRMSE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(H_{p,i} - H_i)^2}}{\sum_{i=1}^{n}(H_{p,i})} \times 100 \tag{20}$$

The MAE is used to calculate, in a set of predictions, the average amount of errors. It can be calculated as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|H_{p,i} - H_i| \tag{21}$$

The MAPE is one of the most commonly used metrics to measure forecast accuracy, which is similar to MAE but normalized by true observation. MAPE can be calculated as follows.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{|H_{p,i} - H_i|}{H_{p,i}} \tag{22}$$

The next metric is the correlation coefficient r which can be calculated as follows.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{23}$$

where x_i represents values of variable x in a sample and y_i represents values of variable y in a sample. $\bar{x}$ and $\bar{y}$ are the mean of the x values and y values, respectively.

Different ensemble-based and single models results are shown in Table 8. It can be seen that the ensemble-based models show promising results than single models of NN, RF, and LSTM. The proposed optimizing ensemble model, based on the deep LSTM learning model, with RMSE of (0.003728832), MAE of (0.00476), and MAPE of (1.8755),

TABLE 9: The proposed optimizing ensemble model' descriptive statistics versus other models.

	NN	RF	LSTM	Average Ensemble	k-NN Ensemble	Optimizing Ensemble
# values	30	30	30	30	30	30
Minimum	0.4306	0.181	0.08629	0.07737	0.01011	0.003668
25% Percentile	0.4521	0.194	0.09462	0.08521	0.0111	0.003682
Median	0.4696	0.2405	0.1144	0.09076	0.01387	0.003715
75% Percentile	0.4934	0.2598	0.1295	0.1012	0.01712	0.003744
Maximum	0.5354	0.288	0.1403	0.1085	0.01862	0.003918
Range	0.1048	0.107	0.05401	0.0311	0.008508	0.0002499
Mean	0.4749	0.2304	0.1136	0.09341	0.01407	0.003722
Std. Deviation	0.03051	0.03408	0.01845	0.009185	0.002865	0.00004836
Std. Error of Mean	0.005571	0.006222	0.003369	0.001677	0.0005231	0.000008829
Lower 95% CI of mean	0.4635	0.2177	0.1067	0.08998	0.013	0.003704
Upper 95% CI of mean	0.4863	0.2432	0.1204	0.09684	0.01514	0.00374
Coefficient of variation	6.425%	14.79%	16.25%	9.833%	20.37%	1.299%
Geometric mean	0.474	0.2279	0.1121	0.09298	0.01378	0.003722
Geometric SD factor	1.065	1.163	1.18	1.104	1.228	1.013
Lower 95% CI of geo. mean	0.4629	0.2154	0.1053	0.08962	0.01277	0.003704
Upper 95% CI of geo. mean	0.4853	0.2412	0.1192	0.09646	0.01488	0.003739
Harmonic mean	0.4731	0.2254	0.1106	0.09254	0.01351	0.003721
Lower 95% CI of harm. mean	0.4623	0.2133	0.1041	0.08926	0.01255	0.003704
Upper 95% CI of harm. mean	0.4844	0.2391	0.118	0.09607	0.01462	0.003739
Quadratic mean	0.4758	0.2329	0.115	0.09385	0.01435	0.003722
Lower 95% CI of quad. mean	0.4641	0.22	0.108	0.09035	0.01323	0.003704
Upper 95% CI of quad. mean	0.4873	0.245	0.1216	0.09722	0.01538	0.00374
Skewness	0.5804	-0.1167	-0.0517	0.08423	0.1666	2.278
Kurtosis	-0.6107	-1.471	-1.401	-1.346	-1.438	8.546
Sum	14.25	6.913	3.407	2.802	0.422	0.1117

TABLE 10: ANOVA results of the base and ensemble models for the wind speed forecasting

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	4.688	5	0.9377	F (5, 174) = 2228	P < 0.0001
Residual (within columns)	0.07323	174	0.000421	-	-
Total	4.762	179	-	-	-

TABLE 11: Wilcoxon Signed Rank test results of the base and ensemble models for the wind speed forecasting

	NN	RF	LSTM	Average Ensemble	k-NN Ensemble	Optimizing Ensemble
Theoretical median	0	0	0	0	0	0
Actual median	0.4696	0.2405	0.1144	0.09076	0.01387	0.003715
# values	30	r30	30	30	30	30
Wilcoxon Signed Rank Test						
Signed ranks (W) sum	465	465	465	465	465	465
Positive ranks sum	465	465	465	465	465	465
Negative ranks sum	0	0	0	0	0	0
P value (two tailed)	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
Estimate or Exact?	Exact	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?						
Discrepancy	0.4696	0.2405	0.1144	0.09076	0.01387	0.003715

gives competitive results compared to the average ensemble and k-NN ensemble models. The detailed descriptive statistics of the proposed optimizing ensemble model versus other models are shown in Table 9. Figure 8 shows the ROC (Receiver Operating Characteristics) curves of the proposed optimizing ensemble model versus other models. The figures

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

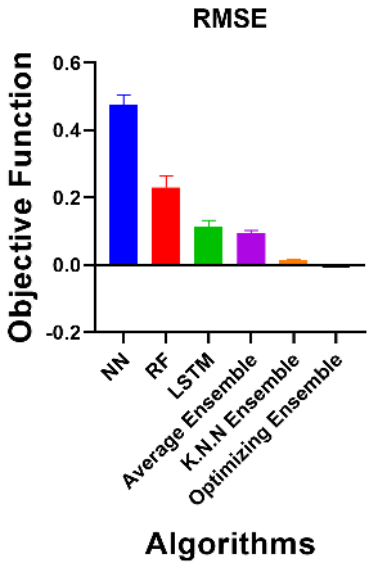

FIGURE 9: RMSE of the presented optimizing ensemble model compared to other models based on objective function

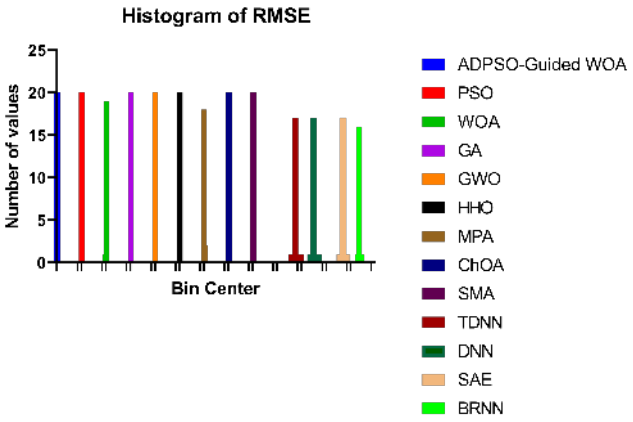

FIGURE 10: Histogram of RMSE of the presented optimizing ensemble model compared to other models based on number of values with Bin Center range (0.00 - 0.59).

show that the proposed ensemble model based on the AD-PSO-Guided WOA algorithm can distinguish data with a high Area Under the Curve (AUC) with a value of 1.0. The RMSE distribution shown in Figure 9, the histogram of RMSE shown in Figure 10, the histogram of RRMSE shown in Figure 11, and the histogram of MAPE shown in Figure 12 confirms the stability of the proposed optimizing ensemble algorithm versus the compared models.

Wilcoxon's rank-sum and ANOVA tests are applied to measure the statistical differences between the proposed and other models that are used for comparison in this scenario. The ANOVA test results are presented in Table 10. Wilcoxon's rank-sum statistical analysis of the proposed ensemble model in comparison to other models is shown

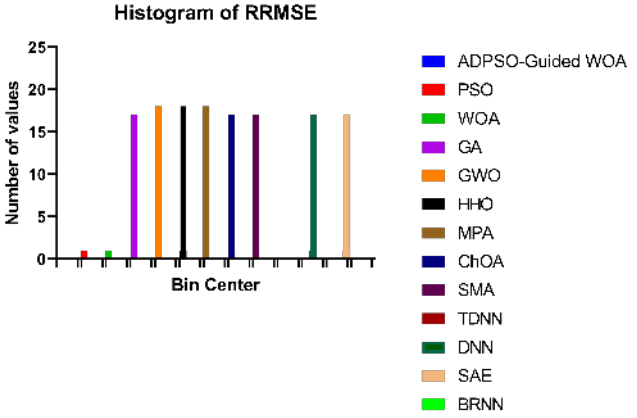

FIGURE 11: Histogram of RRMSE of the presented optimizing ensemble model compared to other models based on number of values with Bin Center range (1.0 - 51.5).

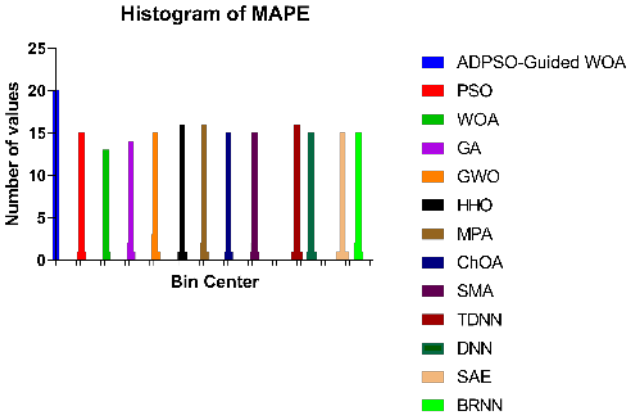

FIGURE 12: Histogram of MAPE of the presented optimizing ensemble model compared to other models based on number of values with Bin Center range (1.8 - 14.8).

in Table 11. Wilcoxon's rank-sum test will determine if the proposed models and other models' results have a significant difference; p-value < 0.05 will show significant superiority. The results explain the superiority of the AD-PSO-Guided WOA based proposed ensemble model and also indicate the statistical significance of the algorithm.

The residual plot in this scenario is shown in Figure 13. The heteroscedasticity plot, QQ plot, and heat map are also shown in Figure 13. Since the distributions of points in the QQ plot are well fitted on the line, the predicted and the actual residuals are considered as linearly related which confirms the proposed AD-PSO-Guided WOA ensemble-based algorithm's performance for the wind speed forecasting problem.

D. COMPARISONS SCENARIO

The third and last scenario is designed to show the performance of the optimizing ensemble-based AD-PSO-Guided WOA algorithm compared with PSO [18], WOA [22], GA

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111408, IEEE Access

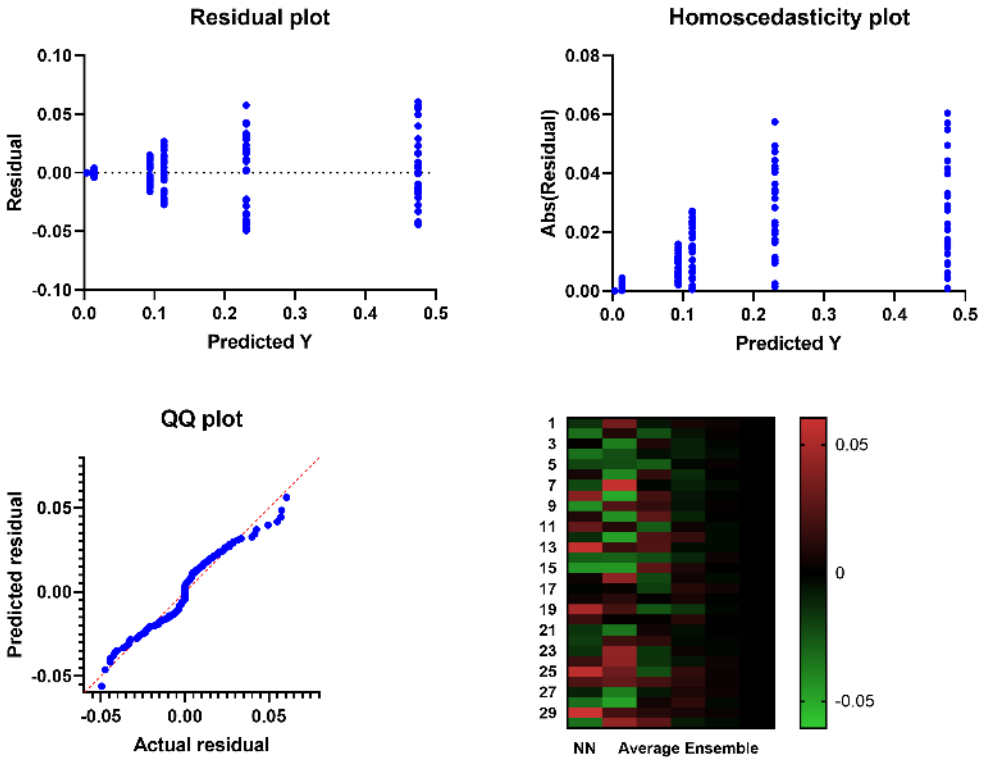

FIGURE 13: Residual, heteroscedasticity, QQ plots and heat map of the presented ensemble-based and compared models for wind speed forecasting problem

TABLE 12: Comparison of wind speed forecasting results using the proposed algorithm compared to other optimization techniques.

Metric	ADPSO-Guided WOA	PSO	WOA	GA	GWO	HHO	MPA	ChOA	SMA
MAPE (%)	1.8755	2.337	2.3559	2.6697	2.5574	4.1236	2.3157	2.44689	3.3312
MAE	0.00476	0.00715	0.007331	0.009123	0.008845	0.0099845	0.005089	0.007402	0.0068875
RMSE	0.003728832	0.00612567	0.006613066	0.008797	0.007802	0.009532	0.0049876	0.006675	0.0058746
RRMSE (%)	1.279369489	7.325162974	8.445619	9.985368	8.886955	10.131245	5.0124517	7.884593	6.991536
r	0.9998878	0.9977836	0.99764592	0.996771	0.997022	0.986612445	0.9945331	0.997536	0.9981303

[23], GWO [19], HHO [25], [26], MPA [27], ChOA [28], and SMA [29]. The AD-PSO-Guided WOA algorithm ensemble model is also compared with four deep learning techniques including TDNN [30], DNN [31], SAE [32], and BRNN [33].

Table 12 shows the comparison results of the wind speed forecasting based on the proposed algorithm compared to other optimization techniques. The results in the table show that the proposed optimizing ensemble model, based on the LSTM deep learning model and the AD-PSO-Guided WOA algorithm, with MAPE of (1.8755), MAE of (0.00476), RMSE of (0.003728832), RRMSE of (1.279369489), and r of (0.9998878) gives competitive results compared to the PSO, WOA, GA, GWO, HHO, MPA, ChOA, and SMA algorithms for the wind speed forecasting tested problem. Table 13 shows the descriptive statistics of the proposed algorithm compared to other optimization techniques over 20 runs.

The ANOVA test results for wind speed forecasting based on the proposed algorithm compared to other optimization techniques is shown in Table 14. The test of the Wilcoxon Signed-Rank rest of the wind speed forecasting results based on the proposed algorithm compared to other optimization techniques is also shown in Table 15. The results confirm the superiority of the AD-PSO-Guided WOA based proposed ensemble model and indicate the statistical significance of the algorithm for the wind speed forecasting tested problem compared to the PSO, WOA, GA, and GWO algorithms.

Table 16 shows the comparison results of the wind speed forecasting based on the proposed algorithm compared to other deep learning techniques. The results in the table show that the proposed optimizing ensemble model with MAPE of (1.8755), MAE of (0.00476), RMSE of (0.003728832), RRMSE of (1.279369489), and r of (0.9998878) gives competitive results compared to the TDNN, DNN, SAE, and BRNN techniques for the wind speed forecasting tested problem. Table 17 shows the descriptive statistics of the proposed algorithm compared to other deep learning techniques over

TABLE 13: Description statistics of the proposed algorithm compared to other optimization techniques over 20 runs.

	ADPSO-Guided WOA	PSO	WOA	GA	GWO	HHO	MPA	ChOA	SMA
Number of values	20	20	20	20	20	20	20	20	20
Minimum	0.003429	0.005126	0.004613	0.005797	0.00598	0.006532	0.003988	0.005675	0.005075
25% Percentile	0.003729	0.006126	0.006613	0.008797	0.007802	0.009532	0.004988	0.006675	0.005875
Median	0.003729	0.006126	0.006613	0.008797	0.007802	0.009532	0.004988	0.006675	0.005875
75% Percentile	0.003729	0.006126	0.006613	0.008797	0.007802	0.009532	0.004988	0.006675	0.005875
Maximum	0.003829	0.007126	0.008613	0.009997	0.009902	0.01235	0.005988	0.008675	0.007875
Range	0.0004	0.002	0.004	0.0042	0.003922	0.005821	0.002	0.003	0.0028
10% Percentile	0.003639	0.006126	0.005713	0.006897	0.007802	0.009532	0.004988	0.00657	0.005875
90% Percentile	0.003729	0.007026	0.007513	0.008797	0.007802	0.009532	0.005888	0.007665	0.006805
Mean	0.003714	0.006176	0.006613	0.008557	0.007816	0.009523	0.005038	0.006774	0.006
Std. Deviation	0.00007452	0.000394	0.000726	0.000867	0.000638	0.000945	0.000394	0.000563	0.000534
Std. Error of Mean	0.00001666	8.81E-05	0.000162	0.000194	0.000143	0.000211	8.81E-05	0.000126	0.000119
Coefficient of variation	2.006%	6.380%	10.97%	10.13%	8.159%	9.920%	7.821%	8.314%	8.900%
Geometric mean	0.003713	0.006164	0.006574	0.008508	0.007791	0.009476	0.005023	0.006754	0.00598
Geometric SD factor	1.021	1.066	1.12	1.121	1.085	1.111	1.081	1.081	1.086
Harmonic mean	0.003712	0.006152	0.006532	0.008451	0.007766	0.009423	0.005009	0.006735	0.005962
Quadratic mean	0.003715	0.006188	0.006651	0.008599	0.007841	0.009567	0.005053	0.006796	0.006023
Skewness	-3.136	0.5305	5.71E-15	-2.126	0.6519	-0.284	0.5305	2.133	2.475
Kurtosis	12.45	4.985	5.327	5.698	9.699	9.538	4.985	7.604	8.619
Sum	0.07428	0.1235	0.1323	0.1711	0.1563	0.1905	0.1008	0.1355	0.12

TABLE 14: ANOVA test results for wind speed forecasting using the proposed algorithm compared to other optimization techniques over 20 runs.

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.000502	8	6.28E-05	F (8, 171) = 161.7	P < 0.0001
Residual (within columns)	6.64E-05	171	3.88E-07	-	-
Total	0.000569	179	-	-	-

TABLE 15: Wilcoxon Signed Rank Test of wind speed forecasting results using the proposed algorithm compared to other optimization techniques over 20 runs.

	ADPSO-Guided WOA	PSO	WOA	GA	GWO	HHO	MPA	ChOA	SMA
Theoretical median	0	0	0	0	0	0	0	0	0
Actual median	0.003729	0.006126	0.006613	0.008797	0.007802	0.009532	0.004988	0.006675	0.005875
Number of values	20	20	20	20	20	20	20	20	20
Wilcoxon Signed Rank Test									
Sum of signed ranks (W)	210	210	210	210	210	210	210	210	210
Sum of positive ranks	210	210	210	210	210	210	210	210	210
Sum of negative ranks	0	0	0	0	0	0	0	0	0
P value (two tailed)	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?									
Discrepancy	0.003729	0.006126	0.006613	0.008797	0.007802	0.009532	0.004988	0.006675	0.005875

TABLE 16: Comparison of wind speed forecasting results using the proposed algorithm compared to other deep learning techniques.

	ADPSO-Guided WOA	TDNN	DNN	SAE	BRNN
MAPE (%)	1.8755	14.7615	8.1996	12.6632	7.113
MAE	0.00476	0.5377	0.3316	0.4991	0.3226
RMSE	0.003728832	0.5126012	0.3371776	0.478634	0.244604
RRMSE (%)	1.279369489	52.659735	27.887544	48.75756	22.4404
r	0.9998878	0.8766512	0.8963578	0.887633	0.909132

TABLE 17: Description statistics of the proposed algorithm compared to other deep learning techniques over 20 runs.

	ADPSO-Guided WOA	TDNN	DNN	SAE	BRNN
Number of values	20	20	20	20	20
Minimum	0.003429	0.4126	0.2372	0.3786	0.2014
25% Percentile	0.003729	0.5126	0.3372	0.4786	0.2446
Median	0.003729	0.5126	0.3372	0.4786	0.2446
75% Percentile	0.003729	0.5126	0.3372	0.4786	0.2446
Maximum	0.003829	0.6713	0.4937	0.5786	0.2946
Range	0.0004	0.2587	0.2565	0.2	0.09316
10% Percentile	0.003639	0.4934	0.3372	0.4156	0.2356
90% Percentile	0.003729	0.5126	0.3822	0.4786	0.2716
Mean	0.003714	0.5145	0.3425	0.4751	0.2459
Std. Deviation	0.00007452	0.04326	0.04379	0.03602	0.01674
Std. Error of Mean	0.00001666	0.009673	0.009792	0.008055	0.003744
Coefficient of variation	2.006%	8.409%	12.79%	7.582%	6.808%
Geometric mean	0.003713	0.5129	0.34	0.4738	0.2454
Geometric SD factor	1.021	1.083	1.131	1.08	1.07
Harmonic mean	0.003712	0.5113	0.3376	0.4724	0.2449
Quadratic mean	0.003715	0.5162	0.3452	0.4764	0.2465
Skewness	-3.136	2.012	1.654	-0.1042	0.6119
Kurtosis	12.45	11.15	9.226	5.943	5.853
Sum	0.07428	10.29	6.85	9.503	4.919

TABLE 18: ANOVA test results for wind speed forecasting using the proposed algorithm compared to other deep learning techniques over 20 runs.

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	3.357	4	0.8392	F (4, 95) = 781.8	P < 0.0001
Residual (within columns)	0.102	95	0.001073	-	-
Total	3.459	99	-	-	-

TABLE 19: Wilcoxon Signed Rank Test of wind speed forecasting results using the proposed algorithm compared to other deep learning techniques over 20 runs.

	ADPSO-Guided WOA	TDNN	DNN	SAE	BRNN
Theoretical median	0	0	0	0	0
Actual median	0.003729	0.5126	0.3372	0.4786	0.2446
Number of values	20	20	20	20	20
Wilcoxon Signed Rank Test					
Sum of signed ranks (W)	210	210	210	210	210
Sum of positive ranks	210	210	210	210	210
Sum of negative ranks	0	0	0	0	0
P value (two tailed)	0.0001	0.0001	0.0001	0.0001	0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?					
Discrepancy	0.003729	0.5126	0.3372	0.4786	0.2446

20 runs.

The ANOVA test results for wind speed forecasting based on the proposed algorithm compared to other deep learning techniques is shown in Table 18. The test of the Wilcoxon Signed-Rank rest of the wind speed forecasting results based on the proposed algorithm compared to other deep learning techniques is also shown in Table 19. The results confirm the superiority of the AD-PSO-Guided WOA based proposed ensemble model and indicate the statistical significance of the algorithm for the wind speed forecasting tested problem compared to the TDNN, DNN, SAE, and BRNN techniques.

V. CONCLUSIONS

This paper uses a dataset of wind power forecasting as a case study from Kaggle to predict hourly power generation up to forty-eight hours ahead at seven wind farms. A proposed

adaptive dynamic particle swarm algorithm with a guided whale optimization algorithm improves the forecasting performance of the tested dataset by enhancing the parameters of the LSTM classification method. The AD-PSO-Guided WOA algorithm selects the optimal hyper-parameters value of the LSTM deep learning model for forecasting purposes of wind speed. A binary AD-PSO-Guided WOA algorithm is applied for feature selection and it is evaluated in comparison with the GWO, PSO, SFS, WOA, FA, and GA algorithms using the tested dataset. An optimized ensemble method based on the proposed algorithm is tested on the experiments' dataset. The results of this scenario are compared with NN, RF, LSTM, Average ensemble, and k-NN methods. The statistical analysis of different tests is performed to confirm the accuracy of the algorithm, including ANOVA and Wilcoxon's rank-sum tests. The current work's importance is applying a new optimization algorithm to enhance LSTM classifier parameters. In future work, the proposed algorithms will be tested for other datasets. The algorithm will also be applied for other binary problems with a high number of attributes for feature selection, classification problems, and constrained engineering problems. The sparsity of the proposed model will be evaluated and compared with other methods including the sparse autoencoding methods.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support received from Taif University Researchers Supporting Project Number (TURSP-2020/122), Taif University, Taif, Saudi Arabia.

REFERENCES

[1] M. Santhosh, C. Venkaiah, and D. M. V. Kumar, "Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review," Engineering Reports, vol. 2, no. 6, May 2020. [Online]. Available: https://doi.org/10.1002/eng2.12178

[2] B. Kosovic, S. E. Haupt, D. Adriaansen, S. Alessandrini, G. Wiener, L. D. Monache, Y. Liu, S. Linden, T. Jensen, W. Cheng, M. Politovich, and P. Prestopnik, "A comprehensive wind power forecasting system integrating artificial intelligence and numerical weather prediction," Energies, vol. 13, no. 6, p. 1372, Mar. 2020. [Online]. Available: https://doi.org/10.3390/en13061372

[3] Z. Lin, X. Liu, and M. Collu, "Wind power prediction based on high-frequency SCADA data along with isolation forest and deep learning neural networks," International Journal of Electrical Power & Energy Systems, vol. 118, p. 105835, Jun. 2020. [Online]. Available: https://doi.org/10.1016/j.ijepes.2020.105835

[4] M. Ibrahim, A. Alsheikh, Q. Al-Hindawi, S. Al-Dahidi, and H. ElMoaqet, "Short-time wind speed forecast using artificial learning-based algorithms," Computational Intelligence and Neuroscience, vol. 2020, pp. 1–15, Apr. 2020. [Online]. Available: https://doi.org/10.1155/2020/8439719

[5] J. M. Lima, A. K. Guetter, S. R. Freitas, J. Panetta, and J. G. Z. de Mattos, "A meteorological–statistic model for short-term wind power forecasting," Journal of Control, Automation and Electrical Systems, vol. 28, no. 5, pp. 679–691, Jul. 2017. [Online]. Available: https://doi.org/10.1007/s40313-017-0329-8

[6] J. Liu, X. Wang, and Y. Lu, "A novel hybrid methodology for short-term wind power forecasting based on adaptive neuro-fuzzy inference system," Renewable Energy, vol. 103, pp. 620–629, Apr. 2017. [Online]. Available: https://doi.org/10.1016/j.renene.2016.10.074

[7] M. Khodayar, J. Wang, and M. Manthouri, "Interval deep generative neural network for wind speed forecasting," IEEE Transactions on Smart Grid, vol. 10, no. 4, pp. 3974–3989, Jul. 2019. [Online]. Available: https://doi.org/10.1109/tsg.2018.2847223

[8] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 2770–2779, Dec. 2017. [Online]. Available: https://doi.org/10.1109/tii.2017.2730846

[9] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," IEEE Transactions on Sustainable Energy, vol. 10, no. 2, pp. 670–681, Apr. 2019. [Online]. Available: https://doi.org/10.1109/tste.2018.2844102

[10] S. M. J. Jalali, S. Ahmadian, M. Khodayar, A. Khosravi, V. Ghasemi, M. Shafie-khah, S. Nahavandi, and J. P. S. Catalão, "Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting," Engineering with Computers, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s00366-021-01356-0

[11] H. S. Dhiman, D. Deb, and J. M. Guerrero, "Hybrid machine intelligent SVR variants for wind forecasting and ramp events," Renewable and Sustainable Energy Reviews, vol. 108, pp. 369–379, Jul. 2019. [Online]. Available: https://doi.org/10.1016/j.rser.2019.04.002

[12] H. S. Dhiman and D. Deb, "Machine intelligent and deep learning techniques for large training data in short-term wind speed and ramp event forecasting," International Transactions on Electrical Energy Systems, Feb. 2021. [Online]. Available: https://doi.org/10.1002/2050-7038.12818

[13] H. S. Dhiman, D. Deb, S. M. Muyeen, and I. Kamwa, "Wind turbine gearbox anomaly detection based on adaptive threshold and twin support vector machines," IEEE Transactions on Energy Conversion, pp. 1–1, 2021. [Online]. Available: https://doi.org/10.1109/tec.2021.3075897

[14] B. Bilal, M. Ndongo, K. H. Adjallah, A. Sava, C. M. F. Kebe, P. A. Ndiaye, and V. Sambou, "Wind turbine power output prediction model design based on artificial neural networks and climatic spatiotemporal data," in 2018 IEEE International Conference on Industrial Technology (ICIT). IEEE, Feb. 2018. [Online]. Available: https://doi.org/10.1109/icit.2018.8352329

[15] J. Wang, W. Yang, P. Du, and Y. Li, "Research and application of a hybrid forecasting framework based on multi-objective optimization for electrical power system," Energy, vol. 148, pp. 59–78, Apr. 2018. [Online]. Available: https://doi.org/10.1016/j.energy.2018.01.112

[16] Y.-Y. Hong and C. L. P. P. Rioflorido, "A hybrid deep learning-based neural network for 24-h ahead wind power forecasting," Applied Energy, vol. 250, pp. 530–539, Sep. 2019. [Online]. Available: https://doi.org/10.1016/j.apenergy.2019.05.044

[17] J. Zhang, J. Yan, D. Infield, Y. Liu, and F. sang Lien, "Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and gaussian mixture model," Applied Energy, vol. 241, pp. 229–244, May 2019. [Online]. Available: https://doi.org/10.1016/j.apenergy.2019.03.044

[18] R. Bello, Y. Gomez, A. Nowe, and M. M. Garcia, "Two-step particle swarm optimization to solve the feature selection problem," in Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), Oct 2007, pp. 691–696.

[19] E.-S. El-Kenawy and M. Eid, "Hybrid gray wolf and particle swarm optimization for feature selection," International Journal of Innovative Computing, Information and Control, vol. 16, no. 3, pp. 831–844, 2020.

[20] E.-S. M. El-kenawy, A. Ibrahim, S. Mirjalili, M. M. Eid, and S. E. Hussein, "Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images," IEEE Access, vol. 8, pp. 179 317 – 179 335, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3028012

[21] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in Engineering Software, vol. 95, pp. 51 – 67, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997816300163

[22] E. M. Hassib, A. I. El-Desouky, L. M. Labib, and E.-S. M. El-kenawy, "Woa+brnn: An imbalanced big data classification framework using whale optimization and deep neural network," Soft Computing, vol. 24, no. 8, pp. 5573–5592, Mar. 2019. [Online]. Available: https://doi.org/10.1007/s00500-019-03901-y

[23] M. M. Kabir, M. Shahjahan, and K. Murase, "A new local search based hybrid genetic algorithm for feature selection," Neurocomputing, vol. 74, no. 17, pp. 2914 – 2928, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231211002748

[24] I. Fister, X.-S. Yang, I. Fister, and J. Brest, "Memetic firefly algorithm for combinatorial optimization," Tech. Rep. arXiv:1204.5165, Apr 2012, comments: 14 pages; Bioinspired Optimization Methods

and their Applications (BIOMA 2012). [Online]. Available: https://cds.cern.ch/record/1443422

[25] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," Future Generation Computer Systems, vol. 97, pp. 849–872, Aug. 2019. [Online]. Available: https://doi.org/10.1016/j.future.2019.02.028

[26] A. Ibrahim, H. A. Ali, M. M. Eid, and E.-S. M. El-kenawy, "Chaotic harris hawks optimization for unconstrained function optimization," in 2020 16th International Computer Engineering Conference (ICENCO). IEEE, Dec. 2020. [Online]. Available: https://doi.org/10.1109/icenco49778.2020.9357403

[27] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," Expert Systems with Applications, vol. 152, p. 113377, Aug. 2020. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.113377

[28] M. Khishe and M. Mosavi, "Chimp optimization algorithm," Expert Systems with Applications, vol. 149, p. 113338, Jul. 2020. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.113338

[29] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," Future Generation Computer Systems, vol. 111, pp. 300–323, Oct. 2020. [Online]. Available: https://doi.org/10.1016/j.future.2020.03.055

[30] F. Noman, G. Alkawsi, A. A. Alkahtani, A. Q. Al-Shetwi, S. K. Tiong, N. Alalwan, J. Ekanayake, and A. I. Alzahrani, "Multistep short-term wind speed prediction using nonlinear auto-regressive neural network with exogenous variable selection," Alexandria Engineering Journal, vol. 60, no. 1, pp. 1221–1229, Feb. 2021. [Online]. Available: https://doi.org/10.1016/j.aej.2020.10.045

[31] X. Liu, H. Zhang, X. Kong, and K. Y. Lee, "Wind speed forecasting using deep neural network with feature selection," Neurocomputing, vol. 397, pp. 393–403, Jul. 2020. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.08.108

[32] T. Su, Y. Liu, J. Zhao, and J. Liu, "Probabilistic stacked denoising autoencoder for power system transient stability prediction with wind farms," IEEE Transactions on Power Systems, vol. 36, no. 4, pp. 3786–3789, Jul. 2021. [Online]. Available: https://doi.org/10.1109/tpwrs.2020.3043620

[33] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: A survey," Big Data, vol. 9, no. 1, pp. 3–21, Feb. 2021. [Online]. Available: https://doi.org/10.1089/big.2020.0159

[34] M. S. Nazir, F. Alturise, S. Alshmrany, H. M. J. Nazir, M. Bilal, A. N. Abdalla, P. Sanjeevikumar, and Z. M. Ali, "Wind generation forecasting methods and proliferation of artificial neural network: A review of five years research trend," Sustainability, vol. 12, no. 9, p. 3778, May 2020. [Online]. Available: https://doi.org/10.3390/su12093778

[35] E.-S. M. El-Kenawy, S. Mirjalili, A. Ibrahim, M. Alrahmawy, M. El-Said, R. M. Zaki, and M. M. Eid, "Advanced meta-heuristics, convolutional neural networks, and feature selectors for efficient COVID-19 x-ray chest image classification," IEEE Access, vol. 9, pp. 36 019–36 037, 2021. [Online]. Available: https://doi.org/10.1109/access.2021.3061058

[36] A. A. Nasser, M. Z. Rashad, and S. E. Hussein, "A two-layer water demand prediction system in urban areas based on micro-services and LSTM neural networks," IEEE Access, vol. 8, pp. 147 647–147 661, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3015655

[37] S. S. M. Ghoneim, T. A. Farrag, A. A. Rashed, E.-S. M. El-Kenawy, and A. Ibrahim, "Adaptive dynamic meta-heuristics for feature selection and classification in diagnostic accuracy of transformer faults," IEEE Access, vol. 9, pp. 78 324–78 340, 2021. [Online]. Available: https://doi.org/10.1109/access.2021.3083593

[38] S. Mirjalili, S. M. Mirjalili, S. Saremi, and S. Mirjalili, Whale Optimization Algorithm: Theory, Literature Review, and Application in Designing Photonic Crystal Filters. Cham: Springer International Publishing, 2020, pp. 219–238. [Online]. Available: https://doi.org/10.1007/978-3-030-12127-3_13

[39] "Global energy forecasting competition 2012 - wind forecasting," https://www.kaggle.com/c/GEF2012-wind-forecasting, note = Accessed: 2021-06-19.

[40] R. Al-Hajj, A. Assi, and M. M. Fouad, "Stacking-based ensemble of support vector regressors for one-day ahead solar irradiance prediction," in 2019 8th International Conference on Renewable Energy Research and Applications (ICRERA). IEEE, Nov. 2019. [Online]. Available: https://doi.org/10.1109/icrera47325.2019.8996629