



Stance detection using improved whale optimization algorithm

Avinash Chandra Pandey¹ · Vinay Anand Tikkiwal²

Received: 1 October 2020 / Accepted: 1 February 2021 / Published online: 23 February 2021
© The Author(s) 2021

Abstract

News is a medium that notifies people about the events that had happened worldwide. The menace of fake news on online platforms is on the rise which may lead to unwanted events. The majority of fake news is spread through social media platforms, since these platforms have a great reach. To identify the credibility of the news, various spam detection methods are generally used. In this work, a new stance detection method has been proposed for identifying the stance of fake news. The proposed stance detection method is based on the capabilities of an improved whale optimization algorithm and a multilayer perceptron. In the proposed model, weights and biases of the multilayer perceptron are updated using an improved whale optimization algorithm. The efficacy of the proposed optimized neural network has been tested on five benchmark stance detection datasets. The proposed model shows better results over all the considered datasets. The proposed approach has theoretical implications for further studies to examine the textual data. Besides, the proposed method also has practical implications for developing systems that can result conclusive reviews on any social problems.

Keywords Stance detection · Swarm based neural network · Whale optimization · Metaheuristic methods

Introduction

News is a medium that keeps everyone updated about the events that have taken place. It has two parts, one is the headline of the news and the other is the content body. News has the highest reach among all other forms of media [88]. In this era of digitization, news also spread through various social media platforms such as Facebook, Twitter, Whatsapp, etc. Nowadays, doubt regarding the credibility of news has become widespread, since some miscreants are spreading fake news purposefully through different social media platforms. Fake news can have a severe threat to the community due to its extensive reach [86]. In this digital era, everyday terabytes of fake news are created and shared on various online platforms. However, such a high-dimensional data cannot be managed by human beings in real time and human fact-checkers cannot handle such tremendous information in

real time. Thus, to automate the process of fake news detection, several artificial intelligence (AI)-based techniques have been introduced in the literature. One of the popular approach to discover fake news is by identifying the stance of the news [50]. Stance detection involves estimating the relative views (or stance) of two chunks of text relative to a claim, issue, or topic. In general, a stance is defined as a relationship between two textual bodies. In stance detection, a headline and a body text are given and the objective is to classify the headline–body pair into one of the categories, namely, agree, disagree, discuss, and unrelated as discussed below.

1. *Agree stance* A stance is agree stance when the claim of one body is validated by the other body. For example:
Body 1-Mango is known as king of fruits.
Body 2-Mango has a great taste, that is why it is known as king of fruits.
2. *Disagree stance* A stance is disagree stance when the claim of one body is denied by the other body. The same is elaborated in following example:
Body 1-Mango is known as king of fruits.
Body 2-Mango has a great taste but we cannot say it as king of fruits.
3. *Discuss stance* A stance is discuss stance when the claim of one body is neither denied nor validated by the other

✉ Avinash Chandra Pandey
avish.nsit@gmail.com

Vinay Anand Tikkiwal
vat2k1@gmail.com

¹ Department of CSE&IT, Jaypee Institute of Information Technology, Noida, India

² Department of Electronics Engineering, Jaypee Institute of Information Technology, Noida, India

body rather the other body discuss the claim made by the other body. For example:

Body 1-Mango is known as king of fruits.

Body 2-Mango has a great taste and has great texture which makes him stand apart from the other fruits.

4. **Unrelated stance** A stance is an unrelated stance when the claim of one body is neither denied nor validated by the other body rather the other body discuss anything apart form the claim made by the other body. For example:

Body 1-Mango is known as king of fruits.

Body 2-Arvind Kejriwal is the chief minister of the Delhi.

To discover whether a piece of particular news is fake or not, its stance is identified [32]. If the stance of news belongs to an “agreed” category, then it confirms its genuinity, while, if the stance is “disagreed” or “unrelated”, it implies that the news is fake. It is very difficult to discover whether the news is fake or not if the stance belongs to the discussed category. News belonging to the “discussed” category have to be physically analyzed. To discover the stance of fake news, many models such as traditional stance, machine learning, deep learning, and natural language processing (NLP)-based models have been proposed in the literature. The traditional stance-based models compare the body and headlines to check the credibility of news [27]. Shu et al. [71] presented an NLP-based approach for identifying the stance of fake news. The NLP-based approaches investigate fake news from textual and network perspective. The NLP-based approaches cannot capture semantics from textual data; hence, these approaches sometimes do not perform well. Therefore, various word embeddings such as word2vec, Glove, etc., along with machine learning are used for identifying the stance [76]. Ghanem et al. [28] introduced a hybrid model based on the strength of n-grams, lexical representation of indicative words, and word embeddings for stance detection of fake news. In the literature, it has been stated that the performance of stance classification can be improved if appropriate representative features are used. However, the existing approaches use some predefined lexicons and word embeddings to extract features from textual data. Thus, it is possible that the extracted features may be irrelevant. To extract relevant features, deep learning models are generally used.

The deep neural network-based models extract useful features automatically from datasets using backpropagation [63]. However, the performance of backpropagation falls rapidly if the complexity of the problem increases [58]. Moreover, the efficiency of neural network models also depends upon hyper-parameters such as filter window size, learning rate, word embedding techniques, and batch size. Therefore, to improve the performance and optimize the values of hyper-parameters, metaheuristic methods are generally used. The

whale optimization algorithm (WOA) is successfully used in literature for optimizing FFNNs [5]. Moreover, WOA is also used to solve text classification and sentiment analysis problems [49,78]. In this paper, an improved whale optimization algorithm (IWOA) has been proposed for optimizing the hyper-parameters of neural networks. The proposed model uses the word embedding technique for normalizing the textual data followed by an optimized neural network to get the stance. The proposed IWOA optimized FFNN can be used for solving single-objective minimization and maximization optimization problems. Furthermore, the proposed algorithm can also be extended to solve multi-objective optimization problems such as allocating parking lots in a distribution network, optimization of RO desalination plants, etc. The main contributions of this article can be outlined as follows.

- A new variant of whale optimization has been proposed and validated on standard benchmark functions.
- The proposed improved whale optimization algorithm (IWOA) enhances the efficiency by balancing the exploration and exploitation capabilities of WOA.
- The proposed IWOA method are used for optimizing the hyper-parameters of neural network and to identify the stance of textual data.

The remaining paper is ordered as follows. The section “Background study” briefs the related work in the field of stance detection. Preliminaries are discussed in the section “Preliminaries” and the proposed work is presented in the section “Proposed algorithms”. The section “Evaluating IWOA for bias(es)” evaluates for the bias(es) and the section “Experimental results” reports the Experimental outcomes followed by the conclusion in the section “Conclusion and future work”.

Background study

Recently, many deep neural network-based techniques have been proposed for stance detection. Riedel et al. [68] proposed a two-step model for stance detection. The authors first extracted and created a vocabulary set of 5000 most frequent words using TF-IDF which are then passed to a multilayer perceptron having a single hidden layer to get the stance. The proposed technique shows acceptable results for agreeing label stance. However, it does not give satisfactory results for other stances. Therefore, Davis and Proctor [21] presented three different approaches for correctly identifying the stances of the news; the first approach is based on bag-of-words with a three-layer multilayer perceptron (BoW-MLP), the second is a bag of words with bidirectional LSTM (BoW-BiLSTM), and the third is a bag of words with a concatenated multilayer perceptron (BoW-CMLP). The pro-

posed BoW-MLP model outperforms the other two proposed BoW models with a classification accuracy of 93%. However, the proposed models do not perform well with information captured by the BoW model. Thus, Chen et al. [14] used a vectorization technique based on the strength of the LSTM and attention model to predict stances. For the same, they used three different types of neural network architecture with a bag of vectors. The results show that the bag of vector technique performs well with neural network for related and agree news articles. However, it does not perform well in classifying the stance of discuss and unrelated category.

Furthermore, Chaudhry et al. [12] used GloVe word embedding along with LSTMs and three different encoding schemes, namely conditional, independent, and bidirectional conditional for stance detection. In LSTM with conditional encoding, RNN is arranged sequentially, while in LSTM with bidirectional conditional encoding, the softmax layer is used for prediction. The results show that LSTM with bidirectional conditional encoding outperforms the other two algorithms with an accuracy of 97%. However, the accuracy can be further improved by combining an attention mechanism with high-dimensional pre-trained embedding. Mrowca et al. [52] used 100-dimensional GloVe vector representations with bidirectional LSTM for stance detection. Pfohl et al. [65] used the notions of attention model and conditional encoding-based LSTM for identifying the stance of news articles. Furthermore, basic LSTM, LSTM with attention, and conditional encoding LSTM with attention (CEA-LSTM) have been used for stance detection [9,58]. Zeng et al. [86] compared the performance of deep neural network models with a hand-crafted feature-based system and discovered that the deep neural network model outperformed the hand-crafted feature-based system. Sun et al. [74] used a hierarchical attention network for identifying the stance of news. The performance of the attention model is further improved by combining the hand-crafted features and hidden features. Additionally, Yu et al. [85] implemented an RNN encoder–decoder using LSTM and GRU cells with and without having attention for stance detection. In the proposed model, LSTM and GRU cells allow the flexibility to remember or forget the context and also deal with the vanishing gradient problem. Yoon et al. [84] detected incongruity between the body and headline of news articles using a deep hierarchical encoder.

Moreover, Le and Mikolov [42] combined doc2vec with Word2vec word embedding for stance detection. Besides, Lau and Baldwin [41] presented an improved bag-of-words model for stance detection. However, the existing machine learning approaches use some predefined lexicons and word embeddings to extract features from textual data. Thus, it is possible that the extracted features may be irrelevant wherever deep learning models suffer from vanishing gradient descent problem. Therefore, to overcome the problem of

vanishing gradient descent, metaheuristic algorithms are generally used [7,24,57]. Metaheuristic methods normally show better results than the traditional and state-of-the-art methods over NP problems [11,57,59–61]. Mosavi et al. [51] introduced a hybrid model based on gray wolf optimization and neural network for the classification of sonar data. Mukherjee et al. [53] introduced a PSO optimized multilayer perceptron for malignant melanoma detection. Kohli and Arora [38] presented a chaotic GWO algorithm to enhance the global convergence speed of GWO. Krill herd algorithm has been also used to solve a number of optimization problems. Abualigah [2] employed an improved krill herd algorithm for feature selection and document clustering. Cuevas et al. [20] proposed a social spider algorithm based on the behavior of spider to solve many real-world optimization problems like economic dispatch problem [23], transmission expansion planning problem [22], and optimal power flow solution with single-objective optimization [54]. Furthermore, a new metaheuristic algorithm namely symbiotic organisms search (SOS) has been proposed to solve engineering design and various numerical optimization problems [17]. SOS algorithm has been used to solve complex real-world optimization problems such as predicting sea wave height [3], truss optimization with natural frequency [75], optimal operation of reservoir systems [10], and many more. In continuation, Mirjalili et al. [46] proposed a new bio-inspired optimization algorithm, namely salp swarm optimization algorithm (SSA) for engineering design problems. A number of variants of SSA have presented in the literature for solving various NP problems. Yilmaz et al. [82] presented a biobjective optimization model that minimizes the makespan and reduces the workload imbalance among workers for seru production system. Yilmaz and Durmusoglu [83] compared the performance of different metaheuristics for batch scheduling problem in a multihybrid cell manufacturing system. The distribution optimization approaches have also been used for finding optimal parameters for numerous applications. Sun et al. [73] presented a wind forecasting approach based on two-step short-term probabilistic distribution optimization. WOA is also employed in a number of NP problems. WOA has been proposed by Mirjalili and Lewis [44], which is motivated by the hunting mechanism of humpback whales. Mafarja and Mirjalili [43] proposed a feature selection algorithm based on WOA and simulated annealing. Alzaqebah et al. [6] employed WOA to prioritize the software requirements by assuming requirements as a search space and priority as hunting behavior of the whales. Petrović et al. [64] presented a new variant of WOA to find the optimal solution for the NP-hard scheduling problem. Jiang et al. [37] introduced a discrete WOA to solve the green job shop scheduling problem. A new binary WOA based on S-shaped and V-shaped transfer functions has been proposed by Hussien et al. [35] to solve discrete optimization problems. Chen et al. [15]

employed levy flight and chaotic local search synchronously to balance the exploration and exploitation capabilities of standard WOA. The proposed balance WOA has been used to solve complex constrained engineering design problems. Aljarah et al. [5] used a whale optimization algorithm (WOA) along with an MLP for identifying the stance of news. For the same, the authors employed the WOA algorithm for updating the weights and biases for backpropagation instead of gradient descent. The balance WOA proposed by Aljarah et al. [5] for optimizing the MLP mitigates the issue of vanishing gradient problem and also enhances the convergence speed. From the literature, it has been observed that WOA has been successfully used to solve diverse real-world optimization problems including single-objective and multi-objective problems [1, 16], especially in text-mining tasks [4, 36, 49, 78]. Therefore, in this study, WOA has been used for identifying the stance of fake news.

From the above discussion, it is evident that the traditional machine learning and deep learning models generally suffer from vanishing gradient descent problems and computationally very expensive. Moreover, these methods also perform badly over NP problems. On the contrary, metaheuristic methods do not suffer from gradient descent problem and also show better results on NP problems. In literature, a number of metaheuristic methods have been introduced for optimizing the different types of neural networks and for solving the various real-world problems. Therefore, in this paper, a new variant of WOA named IWOA has been proposed. The proposed IWOA can be used for solving various engineering design and NP problems like truss optimization, a traveling salesman, cell manufacturing system, text classification, and many more. In this work, the proposed IWOA optimized has been used for optimizing a neural network that has been utilized for stance detection for fake news.

Preliminaries

The proposed approach uses an improved whale optimization algorithm (IWOA) for optimizing the weight and biases of multilayer perceptron which are discussed in the following subsections.

Whale optimization algorithm

Whale optimization algorithm (WOA) is a nature-inspired metaheuristic algorithm which is generally used for solving different optimization problems [44]. WOA is motivated by the hunting mechanism of humpback whales. WOA method uses three phases, i.e., encircling the prey, bubble-net attacking, and searching of prey to find the optimal solution. All the phases of WOA are discussed below.

Encircling the prey

In this phase, each of the search agents update their positions vector concerning the position vector of current best agent using Eqs. (1) and (2):

$$\mathbf{Q} = |\mathbf{B} \cdot \mathbf{W}^*(t) - \mathbf{W}(t)| \quad (1)$$

$$\mathbf{W}(t + 1) = \mathbf{W}^*(t) - \mathbf{A} \cdot \mathbf{Q}, \quad (2)$$

where t denotes the iteration, \mathbf{A} and \mathbf{B} are coefficient vectors, \mathbf{W}^* is the position vector of the best solution obtained so far, and \mathbf{W} denotes the position vector of a whale. The values of \mathbf{A} and \mathbf{B} are computed according to Eqs. (3) and (4):

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{d} - \mathbf{a} \quad (3)$$

$$\mathbf{B} = 2 \cdot \mathbf{d}, \quad (4)$$

where \mathbf{a} is a control parameter and \mathbf{d} is a random vector in the range of $[0, 1]$.

Bubble-net attacking

The attacking mechanism performed by a humpback whale is also known as a bubble-net attack. It is basically an exploitation phase in which the obtained solution is further refined for finding the optimal solution. The bubble-net attacking has two phases, i.e., shrinking encircling medium and spiral updating position mechanism. For the course of the next iteration, one of these mechanisms is chosen based on the probability. For the same, the first probability is randomly generated, and if the probability is less than 0.5, then the shrinking encircling mechanism is used; otherwise, the spiral updating position mechanism is used. Both shrinking encircling and spiral updating position mechanisms are discussed in the following paragraph.

Shrinking encircling mechanism To achieve the shrinking encircling behavior, the value of \mathbf{a} , as defined in Eq. (3), is decreased linearly from 2 to 0. In other words, \mathbf{A} is set to $[-a, a]$, where \mathbf{a} is decreased from 2 to 0 over the course of iterations. Therefore, the new position of a search individual using this approach can be defined anywhere in between the current best individual and the original position of the search individual.

Spiral updating position mechanism In this phase, the search agents move in helix shape toward the prey using Eqs. (5) and (6):

$$\mathbf{Q}' = |\mathbf{W}^*(t) - \mathbf{W}(t)| \quad (5)$$

$$\mathbf{W}(t + 1) = \mathbf{Q}' \cdot e^{bk} \cdot \cos 2\pi k + \mathbf{W}^*(t); \quad (6)$$

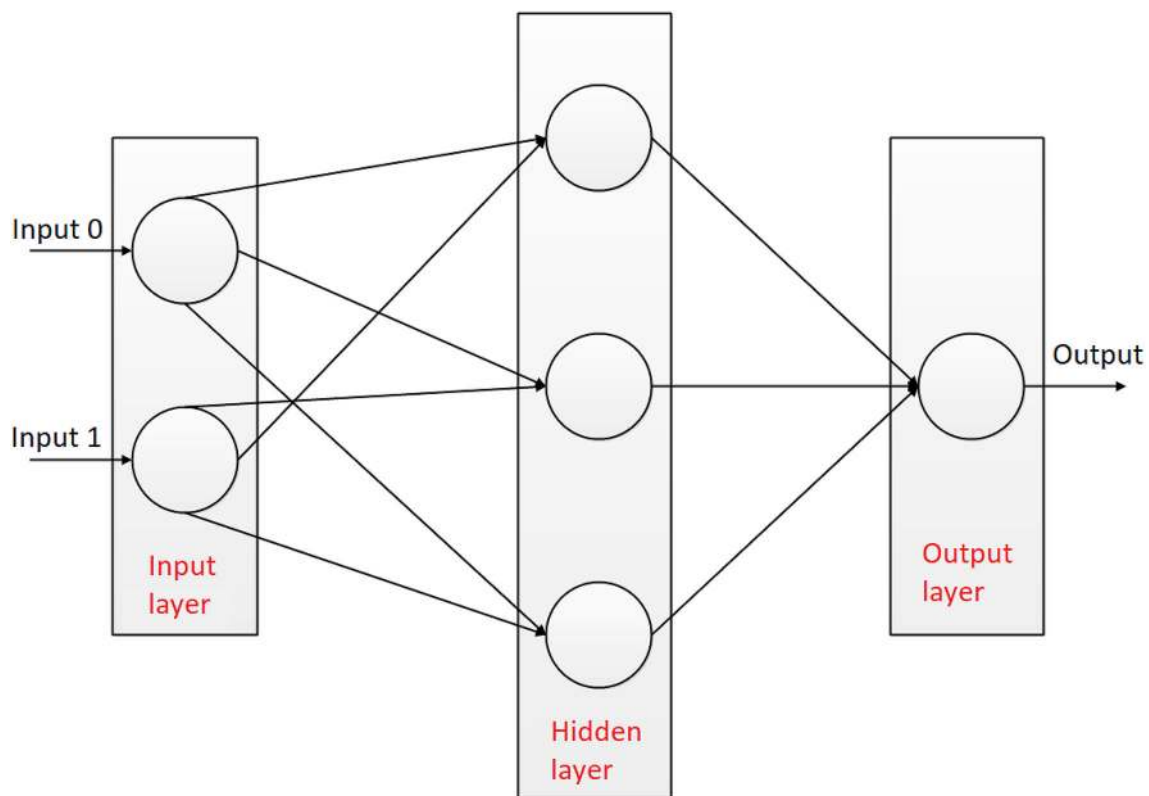


Fig. 1 Block diagram of MLP

here, W^* is position vector of the best search agent, $W(t)$ is position vector of search agent in iteration t , Q' is coefficient vector which is computed using Eq. (5), b is a constant value, and k is a random number in range $[-1, 1]$.

Search for prey (exploration phase)

In this phase, a search agent searches for prey randomly. This random movement serves two purposes; first, it helps in proper investigation of the search region, and the other, it helps to avoid algorithm from being stuck at local optima. Mathematically, it can be formulated using Eqs. (7) and (8):

$$Q = |B \cdot W_{\text{rand}} - W(t)| \quad (7)$$

$$W(t+1) = W_{\text{rand}} - A \cdot Q, \quad (8)$$

where W_{rand} is random vector, $W(t)$ is position vector of search agent at iteration t , A and B are position vectors, and Q is coefficient vector.

Multilayer perceptron neural network

A multilayer perceptron (MLP) is a neural network that consists of multiple perceptrons as depicted in Fig. 1. MLP consists of an input layer to accept the inputs, an output layer that predicts the output, and hidden layers [55]. The hidden

layer is the true computing engine of the MLP which learns more complicated features of data. MLP is generally used for supervised learning problems in which they are trained on a variety of input–output pairs to learn the dependencies (or correlation) between those inputs and outputs. Feedforward neural networks (FFNNs) are a specific form of MLP [25]. In FFNNs, neurons are interlinked in a one-way and one-directional manner. Connections are depicted by weights that are real numbers and fall in the range $[-1, 1]$. Figure 1 illustrates an FFNN with only one hidden layer.

Proposed algorithms

In this paper, weights and hyper-parameters of MLP have been optimized using an improved whale optimization algorithm for enhancing the efficiency of stance detection. The next subsections describe the proposed improved whale optimization algorithm followed by the proposed stance detection method.

Improved whale optimization algorithm

The success of metaheuristic methods depends upon diversification and intensification steps [48,56,61,62]. The metaheuristic methods which maintain the trade-off between

diversification and intensification are considered superior. In whale optimization algorithm (WOA), search individual (humpback whales) that guides the search process is randomly selected. Therefore, WOA usually suffers from slow convergence. Moreover, it also sticks at some local solutions due to hastening exploitation. Thus, in this paper, a new variant of WOA named improved whale optimization algorithm (IWOA) has been introduced to boost the convergence speed and to attain better results. The proposed variant is invigorated with the capabilities of WOA, roulette wheel, and tournament selection. The proposed improved whale optimization algorithm is discussed below.

Improved search for prey using hybrid TS–RS selection

In the intensification (exploitation) phase of WOA, humpback whales that control the entire search process are randomly selected. However, due to this random selection of humpback whales, WOA sometimes takes a longer time to find the optimal solution. Therefore, to equalize the exploration and exploitation and boost the convergence speed, the proposed IWOA uses tournament selection and roulette wheel selection [43] in alternate iterations. In tournament selection, search agents with the best fitness are selected from a group and these search agents guide the whole search process. However, the tournament selection-based WOA sometimes traps in a local solution if local search agents are selected as humpback whales. Thus, to maintain the diversity, roulette wheel selection [49] and tournament selection are used alternately to select the humpback whales. The proposed IWOA uses the following steps to find the searched individual that guides the whole search process. First, tournament selection is used in odd iteration to find the search agents (humpback whales). Tournament selection employs the following steps to select the searched individual:

1. Select few search individuals randomly from the population (a tournament).
2. Compute fitness of each search individual and sort them according to their fitness.
3. The search individual with the best fitness value (the winner) is selected.

Secondly, roulette wheel selection is used to control the trade-off between exploration and exploitation. To understand mathematically, consider there are n search agents in a population $P_s = \{s_1, s_2, \dots, s_n\}$ and fitness value of each search agents s_i is $f(s_i)$. First, tournament selection is used to elect the humpback whales. Second, roulette wheel selection is employed in which selection probability ($P_m(s_i)$) as well as cumulative probability ($P_l(s_k)$) as given in Eqs. (9) and (10) are used to select the search agents:

$$P_m(s_i) = \frac{f(s_i)}{\sum_{i=1}^n f(s_i)}, \quad i = 1, 2, \dots, n \quad (9)$$

$$P_l(s_k) = \sum_{j=1}^k p_l(s_j), \quad \text{for } k = 1, 2, \dots, n. \quad (10)$$

In diversification phase, position of search individual updated according to Eqs. (11) and (12).

$$\mathbf{W}(t+1) = \mathbf{W}_{TS} - \mathbf{A} \cdot \mathbf{Q} \quad \text{if iteration is odd} \quad (11)$$

$$\mathbf{W}(t+1) = \mathbf{W}_{RS} - \mathbf{A} \cdot \mathbf{Q} \quad \text{if iteration is even} \quad (12)$$

$$\mathbf{Q} = |\mathbf{B} \cdot \mathbf{W}_{(RS \text{ or } TS)} - \mathbf{W}(t)| \quad (13)$$

here, \mathbf{W}_{RS} and \mathbf{W}_{TS} is elected by the roulette wheel and tournament selection, respectively. The improved WOA is depicted in Algorithm 1. Furthermore, the proposed IWOA has been used to optimize the neural network for enhancing the efficiency of stance detection.

Algorithm 1 Improved Whale Optimization Algorithm (IWOA)

```

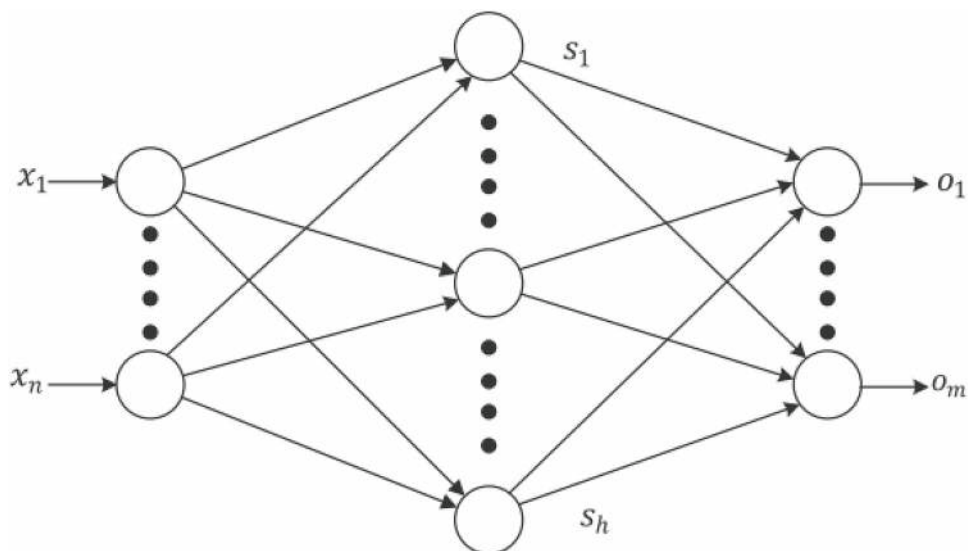
Randomly generate the initial solutions ( $W_i$ )
Evaluate the fitness ( $fit$ ) of all the search agents and find the best search agent ( $Y^*$ )
while Termination condition not satisfied do
  for each individual do
    Update a, A, B, and Q
    if ( $Q < 0.5$ ) then
      if  $|A| < 1$  then
        Update the search individual ( $W_i$ ) using Eq. (2)
        if iteration is odd then
          Select a search agent ( $W_{RS}$ ) using tournament selection and update the current individual by Eq. (11)
        else if iteration is even then
          Select a search agent ( $W_{TS}$ ) using roulette wheel selection and update the current individual by Eq. (12)
        end if
      else if ( $Q \geq 0.05$ ) then
        Update the search individual ( $W_i$ )
      end if
    end if
    Calculate the fitness of all individuals and update the best search individual ( $W^*$ )
    Update a, A, B, and Q
  end for
end while

```

Proposed stance detection method

Nowadays, metaheuristic methods are also used for training multilayer perceptron neural networks (MLPNN) to improve their performance. Metaheuristic-based neural network differs from traditional multilayer perceptron as the weight and biases are updated by the metaheuristic algorithms instead of gradient descent algorithms. Hence, these methods generally do not suffer from vanishing gradient and exploding

Fig. 2 Structure of three-layer FNNs



gradient problems in large neural networks. Metaheuristic methods are applied to three aspects of neural networks. First, metaheuristic methods are applied to find the combination of weights and biases that minimizes the error. Second, metaheuristic methods find a proper structure of FFNN for a problem. Finally, parameters such as momentum and learning rate are tuned for gradient-based learning. In this work, an improved whale optimization algorithm-based stance detection method (IWOASD) has been introduced for finding the optimal values of weights and biases. The proposed IWOASD is based on the capabilities of IWOA and MLP. The most important factor in the proposed IWOASA is to find the optimal values for weights and biases. To achieve the objective, the input values given to the IWOA should be in the form of vector as given in Eq. (14):

$$V = \{\mathbf{W}, \mathbf{b}\} = \{W_{11}, W_{12}, W_{13}, \dots, W_{mm}, h, b_1, b_2, b_3, \dots, b_j\}, \tag{14}$$

where m is the number of the input nodes, W_{ij} is the connection weight from the node i to j , and b_j is the bias.

The next step is to define the objective function of the proposed IWOASD algorithm. The objective function for the proposed IWOASD has been discussed below.

Objective function Consider an FFNN depicted in Fig. 2 with three layers (one hidden, one input, and one output) has been given. The output of each hidden in every epoch is computed according to Eq. (15):

$$f(p_j) = \frac{1}{\left(1 + \exp\left(-\left(\sum_{i=1}^M (W_{ij} \cdot x_j - b_j)\right)\right)\right)}, \quad j = 1, 2, \dots, h, \tag{15}$$

where h is count of hidden layer, W_{ij} is connecting weight from i th node to j th node, x_j is j th input, and b_j is bias.

After computing outcomes of hidden layer, final output O^k of FFNN is computed using Eq. (16):

$$O^k = \sum_{j=1}^h W_{kj} \cdot f(p_j) - b_k, \quad k = 1, 2, \dots, m; \tag{16}$$

here, W_{kj} is the connection weight from k th output node to j th hidden node.

Finally, mean square error (MSE) is computed using Eq. (17). The value of MSE is the difference between MLP output O^k and desired output d^k . Furthermore, for making the FFNN more effective, the efficacy of the FFNN is computed by taking the average value of MSE over every training samples using Eq. (18):

$$MSE = \sum_{j=1}^m (O_j^k - d_j^k)^2 \tag{17}$$

$$\overline{MSE} = \sum_{k=1}^S \frac{\sum_{j=1}^m (O_j^k - d_j^k)^2}{S}, \tag{18}$$

where S is the number of training instances, d_j^k is predicted output, and O_j^k is the actual output of j th input when k th training sample is used.

To find the combination of optimal weights and biases, the proposed IWOASD algorithm uses MSE defined in Eq. (18). To design an optimized neural network, the encoding strategy also needs to be defined.

Encoding strategy In the literature, there are three encoding and representing methods, namely vector, matrix, and binary for representing weights and biases in metaheuristic methods.

In a vector scheme, every agent is encoded as a vector, while in a matrix scheme, every search individual is encoded as a matrix. On the other hand, in the binary encoding scheme, each search individual is represented as strings of binary bits. The decoding particle's vectors of biases and weights in the vector scheme is a complicated task. On the contrary, the binary encoding strategy represents particle variables in binary form in which the length of particle grows for the complex neural network structure, while the decoding strategy of the matrix approach is simple and easy to execute. Therefore, in this work, matrix encoding strategy has been used.

The detailed steps of stance detection using an optimized neural network are presented in Algorithm 2.

Algorithm 2 IWOA-based stance detection algorithm (IWOASD)

```

Initialize the weight matrix and bias for each whale  $W_i (i = 1, 2, 3, \dots, n)$ 

Initialize the position vector  $A$ ,  $B$  according to Eqs. 5 and 6.
Calculate the classification error for each whale.
 $W_{best}$  = best search agent i.e the whale having minimum error.
while  $t \leq$  maximum iteration do
  for each search agent do
    if  $(p \leq 0.5)$  then
      if  $(|A| < 1)$  then
        Perform encircling of prey according to Eqs. 3 and 4 for
        both weight matrices and biases.
      else
        Perform bubble-net attack according to Eqs. 7, 8, 9 and
        10.
      end if
    else
      Perform search for prey according to Eqs. 17 and 18.
    end if
  end for
  Calculate classification error for every search agent.
  Update the best search agent accordingly.
  Update  $A$ ,  $B$ , and  $p$ .
   $t=t+1$ 
end while

```

The workflow of the proposed model can be summarized as follows.

1. First, weights of MLPNN are generated randomly using IWOA
2. Second, fitness of all the individuals are computed using the objective function that minimizes the mean squared error as given in Eq. (18)
3. Weights are updated using IWOA
4. Steps 2–3 are repeated till convergence.

The entire steps of the proposed IWOASD algorithm are depicted in Fig. 3. From the figure and above discussions, it has been observed that the proposed IWOASD algorithm optimizes the weight and biases of MLP by minimizing

the MSEs. Considering MLP part similar for all the algorithms, the computational complexity of the IWOA part can be defined as $O(I \times D \times N^2)$, where I is the current iteration, D is the dimension, and N is the total number of individuals (whales) in population. The computational complexity of FFNN depends upon training and an inference phase. The total time taken to train (backpropagation) an FFNN with n nodes will be $O(n^5)$ and $O(n^4)$ for forward propagation or inference phase. Thus, it can be perceived that the backpropagation phase is much slower than forward propagation. Due to this issue, nowadays, pre-trained neural networks are generally used.

Evaluating IWOA for bias(es)

Generally, the metaheuristic method shows biased results on a number of benchmark functions. Thus, it is better to get an idea of the inherent bias(es) of metaheuristic methods before testing their performance on benchmark functions. Metaheuristic methods may be central bias, edge bias, or/and axial bias. In central bias, solutions closed to the center are explored, while in edge bias, solutions near the boundary (edges) are explored. On the other hand, axial bias search solutions along the x - and y -axis. Therefore, signature test [19] has also been carried out to inspect the bias(es) of the proposed IWOA and standard WOA. For the same, the minimization problem is given in Eq. (19) is used in which every point in the search region represents an optimal solution and an unbiased metaheuristic method produces solutions similar to random search. The outcome of the signature test for WOA and IWOA is depicted in Fig. 4. From the figure, it can be envisioned that the IWOA explores the entire search space evenly and unbiased, while WOA shows center biased results:

$$\text{Minf}(x_1, x_2) = 3; x_1, x_2 \in [-3, 3]. \quad (19)$$

Experimental results

The performance of the proposed stance detection method is discussed in the following subsections. First, the section “Performance analysis of IWOA” investigates the efficacy of the IWOA over unimodal and multimodal benchmark functions. The section “Performance estimation of stance detection method” discusses the performance analysis of IWOA-optimized neural network on five stance detection benchmark datasets. All the experiments have been performed on Matlab 2017a on a computer with 8 GB of RAM and a 2.66 GHz core i3 processor.

Fig. 3 Flow graph for optimized neural network

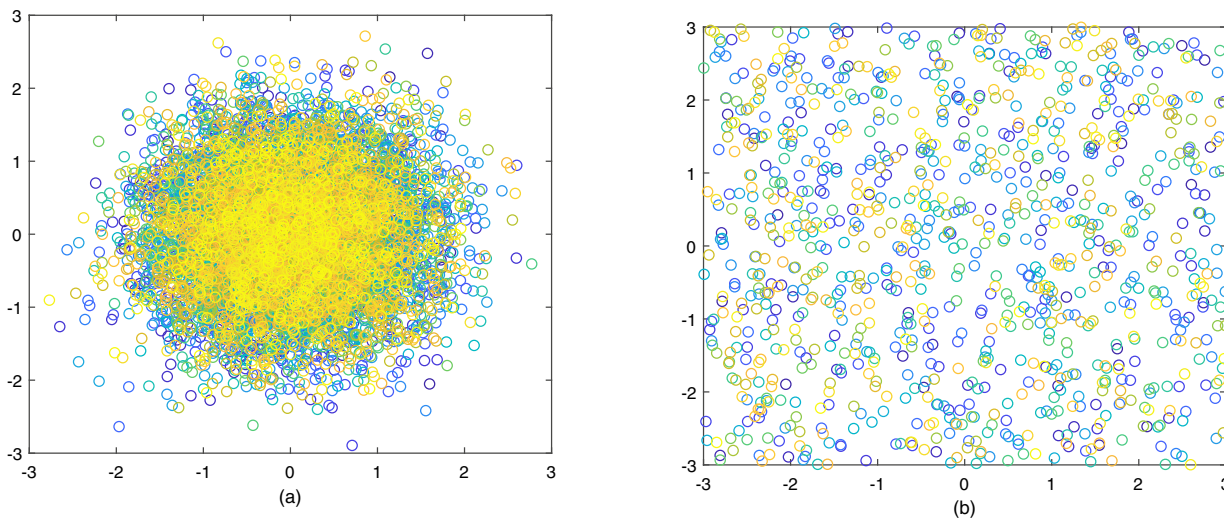
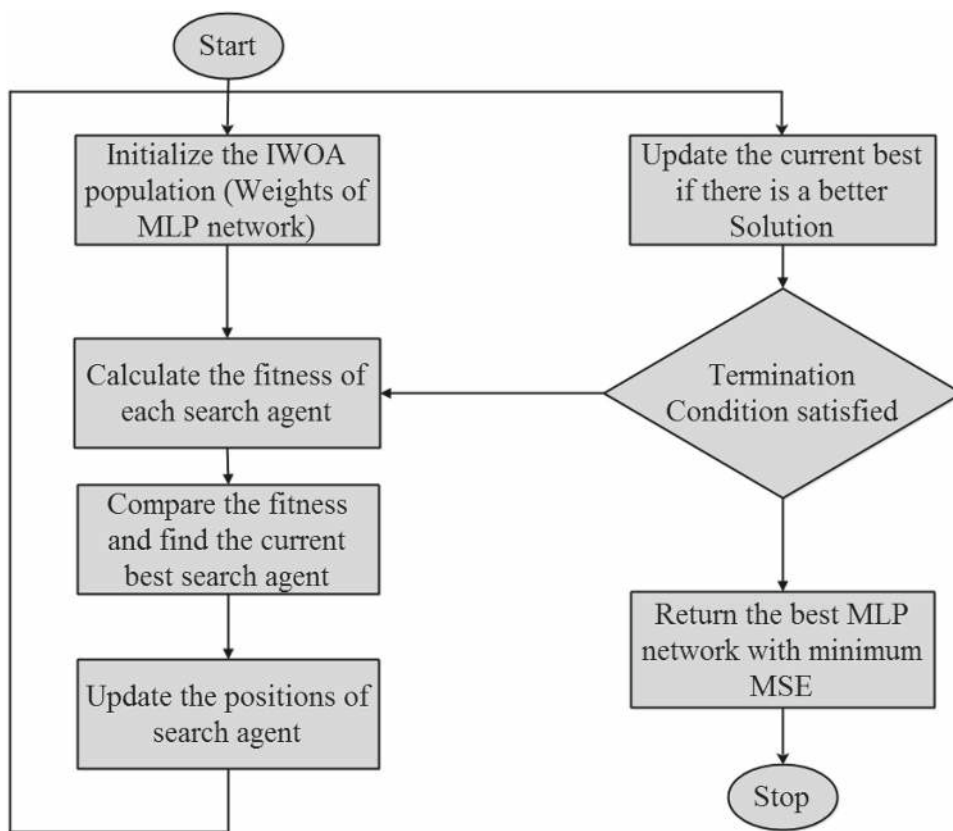


Fig. 4 Signature test analysis of **a** WOA and **b** IWOA

Performance analysis of IWOA

The efficiency of IWOA has been evaluated over 17 benchmark functions including both unimodal ($F_1–F_{10}$) and multimodal ($F_{11}–F_{17}$) functions [60,72]. Unimodal functions generally appraise the convergence performance, while multimodal functions investigate the probability of trapping into local solution. The considered unimodal and multimodal

benchmark functions are depicted in Table 1. To assess the efficiency of the proposed IWOA, mean fitness values of IWOA and other state-of-the-art metaheuristic algorithms, namely cuckoo search (CS) [40,80], gray wolf optimization (GWO) [45], whale optimization algorithm (WOA) [44], grasshopper optimization algorithm (GOA) [47], bat algorithm (BA) [81], hybrid cuckoo search (CSK) [61], vector co-evolving particle swarm optimization (VCPSO) [87], global

Table 1 Description of standard benchmark functions

S.no.	Functions	Optimal value	Category
1.	$F_1(X) = \sum_{i=1}^{d-1} [100\epsilon_i(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	0	Unimodal
2.	$F_2(X) = \max_i(x_i : i \in \{1, \dots, d\})$	0	Unimodal
3.	$F_3(X) = \sum_{i=1}^d ix_i^4$	0	Unimodal
4.	$F_4(X) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	0	Unimodal
5.	$F_5(X) = \sum_{i=1}^d x_i^2$	0	Unimodal
6.	$F_8(X) = \sum_{i=1}^{d-1} (x_i^2)^{(x_{i+1}+1)} + (x_{i+1}^2)^{(x_i^2+1)}$	0	Unimodal
7.	$F_6(X) = \sum_{i=1}^{d/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	0	Unimodal
8.	$F_7(X) = \sum_{i=1}^{d-2} (x_{i-1} + 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4$	0	Unimodal
9.	$F_{13}(X) = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2$	0	Unimodal
10.	$F_{17}(X) = \sum_{i=1}^d (x_i)$	0	Unimodal
11.	$F_9(X) = -20e^{-0.02\sqrt{d-1}\sum_{i=1}^d x_i^2} - e^{d-1}\sum_{i=1}^d \cos(2\pi x_i) + 20 + e$	0	Multimodal
12.	$F_{10}(X) = 1 + \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right)$	0	Multimodal
13.	$F_{11}(X) = \sum_{i=1}^d x_i \sin(x_i) + 0.1x_i $	0	Multimodal
14.	$F_{12}(X) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{ x_i })$	0	Multimodal
15.	$F_{13}(X) = \sum_{i=1}^{d-1} \left(0.5 + \frac{\sin^2 \sqrt{100x_i^2 + x_{i+1}^2 - 0.5}}{1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)} \right)$	0	Multimodal
16.	$F_{14}(X) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + \sin^2(2\pi w_d)]$, where $w_i = 1 + \frac{x_i - 1}{4}$, for all $i = 1, \dots, d$	0	Multimodal
17.	$F_{19}(X) = \sum_{i=1}^d x_i^2 + \left(\frac{1}{2} \sum_{i=1}^d ix_i\right)^2 + \left(\frac{1}{2} \sum_{i=1}^d ix_i\right)^4$	0	Multimodal

and local neighborhood-based PSO (GLNPSO) [18], scout particle swarm optimization (ScPSO) [39], random spare reinforced whale optimization algorithm (RDWOA) [16], hybrid whale optimization algorithm (HWOA) [1], naive Bayes-based whale optimization algorithm (NB-WOA) [34], and enhanced whale optimization algorithm (EWOA) [49] have been computed. For a fair comparison, the controlling parameters such as population dimension (P_{dims}) is set to 30 and 50 and max iteration (mitr) is 1000, for all the algorithms. The other parameters are taken from their respective literature and the values of parameters for the proposed algorithm have been decided empirically by testing its performance on different parameter values of standard WOA algorithm.

As nature-inspired algorithms are randomized by their behavior, therefore all the algorithms have been run 30 times and their average values are used for comparison. The average value of fitness of all the algorithm is displayed in Tables 2 and 3. It is observed from the tables that the proposed IWOA demonstrates the best outcomes than the state-of-the-art methods on 85% of the benchmark functions. There are a few functions for which other algorithms show better results than IWOA. EWOA shows comparable performance on F_1 , F_{13} , and F_{15} for only one dimension and on both dimensions

of function F_3 , while WOA performs better than the proposed IWOA for dimensions 30 and 50 over F_7 and F_{14} benchmark functions, respectively. RDWOA achieves the best fitness value for the benchmark functions F_5 and F_6 , respectively, for dimension 50, while for the benchmark functions F_{10} and F_{11} , ScPSO, RDWOA, HWOA, EWOA, and NB-WOA show equivalent performance. Hence, it is evident that the proposed IWOA performs better than the compared methods.

Furthermore, the non-parametric Friedman's test [89] has also been conducted in Table 4 to statistically validate the efficacy of IWOA. Friedman's test assesses the efficiency of each method on every benchmark function and ranks them according to their performance [79]. The best performing method gets the rank of 1, the next best obtain rank 2, then 3, ..., n . If two methods have the same performance, then the rank is computed by averaging the ranks returned in different runs [70]. The p value returned by the Friedman test is 0.0004518, which is much smaller than the threshold ($\alpha = 0.05$) which signifies that the obtained results are significantly different. Table 4 tabulates the ranks of all the methods returned by the Friedman test. From Table 4, it is evident that the proposed IWOA has a minimum ranking value among all the

Table 2 Comparative analysis of existing and proposed algorithms for mean fitness over 30 runs on the benchmark functions (F_1 – F_9) for 30 and 50 dimensions

Funs.	Dims.	CS	GWO	BA	WOA	GOA	CSK	VCPFO	GLNPSO	ScFPO	RDWOA	HWOA	NB-WOA	EWOA	IWOA
F_1	30	4.09E-01	5.98E-01	5.06E-01	1.06E-02	1.64E-01	8.64E-04	4.32E-03	2.22E-03	1.44E-03	1.72E-04	7.98E-04	2.81E-04	8.91E-05	2.97E-06
	50	2.87E+00	8.11E+00	9.50E-03	9.73E-04	8.77E-01	4.66E-02	2.34E-01	7.29E-04	2.27E-03	4.59E-04	2.17E-04	1.59E-04	1.23E-04	2.63E-03
F_2	30	8.99E-06	1.09E-05	8.03E-06	2.83E-05	1.96E-06	3.54E-07	1.77E-06	3.71E-06	1.70E-07	1.24E-07	3.05E-06	3.40E-07	4.83E-08	1.16E-08
	50	2.20E-05	3.69E-05	3.27E-05	5.90E-08	6.58E-06	8.49E-07	5.10E-06	3.29E-05	1.89E-06	1.55E-06	2.08E-05	6.10E-06	9.44E-07	2.68E-08
F_3	30	3.74E-08	7.04E-08	8.47E-08	6.25E-08	3.20E-08	2.00E-08	1.00E-07	3.85E-08	1.39E-09	1.79E-08	2.38E-08	6.72E-10	9.23E-11	2.82E-08
	50	4.11E-08	7.78E-08	8.78E-08	3.94E-07	3.49E-08	1.98E-08	1.19E-07	2.23E-08	5.05E-09	1.35E-09	1.64E-08	2.21E-09	8.40E-12	2.82E-09
F_4	30	2.76E-07	3.24E-07	4.42E-07	1.14E-06	2.14E-07	3.03E-08	1.82E-07	4.09E-07	5.37E-08	3.22E-08	3.69E-07	3.67E-07	2.30E-08	1.87E-09
	50	9.54E-07	1.60E-06	1.20E-06	9.64E-08	6.92E-07	5.42E-08	3.26E-07	6.80E-07	2.56E-08	5.21E-09	9.63E-08	8.31E-08	7.48E-10	4.25E-10
F_5	30	6.04E-08	1.14E-07	1.38E-07	9.64E-08	5.19E-08	1.21E-08	1.64E-07	6.27E-08	5.98E-08	5.53E-08	2.41E-08	2.39E-08	7.22E-09	3.27E-10
	50	6.06E-08	1.14E-07	1.38E-07	1.14E-07	5.20E-08	4.69E-09	1.96E-07	1.11E-07	3.69E-09	7.02E-10	8.77E-08	6.13E-08	9.67E-07	1.27E-08
F_6	30	7.24E-08	1.36E-07	1.67E-07	1.14E-07	6.23E-08	3.95E-08	2.38E-07	1.34E-08	4.20E-08	1.70E-08	9.88E-09	6.47E-10	1.30E-08	9.06E-09
	50	7.25E-08	1.37E-07	1.67E-07	1.33E-07	6.23E-08	3.94E-08	2.37E-07	1.10E-07	4.58E-09	2.76E-09	7.48E-08	6.63E-08	6.33E-08	5.67E-06
F_7	30	8.71E-08	1.63E-07	1.94E-07	1.33E-10	7.41E-08	4.80E-08	2.41E-07	9.88E-08	4.32E-08	3.56E-08	8.73E-08	1.27E-08	1.78E-09	6.59E-08
	50	9.06E-08	1.78E-07	1.94E-07	1.59E-07	7.46E-08	7.86E-08	3.93E-07	1.83E-07	5.46E-08	1.12E-08	5.45E-08	4.32E-08	5.19E-08	6.78E-09
F_8	30	4.99E-07	2.65E-07	2.23E-07	2.22E-07	1.69E-07	1.38E-07	6.88E-07	1.66E-07	1.52E-08	1.60E-07	6.53E-07	2.28E-07	4.58E-07	2.00E-07
	50	9.60E-05	1.67E-03	2.45E-07	1.74E-07	1.31E-05	9.20E-06	4.60E-05	1.30E-07	1.87E-05	1.19E-05	4.10E-08	7.38E-09	2.07E-06	1.85E-09
F_9	30	6.93E-13	3.72E+01	1.66E-17	1.32E-20	6.48E+01	1.21E+01	7.64E-18	5.46E-12	9.07E-33	8.43E-33	1.63E-34	1.01E-35	2.92E-33	1.84E-43
	50	3.46E-13	1.86E+01	5.53E-18	4.40E-21	3.24E+01	4.05E+00	3.39E-18	1.81E-12	1.52E-33	2.71E-34	1.21E-33	6.68E-32	7.50E-33	1.12E-34

Best values are present in bold

Table 3 Comparative analysis of existing and proposed algorithms for mean fitness over 30 runs on the benchmark functions ($F_{10} - F_{17}$) for 30 and 50 dimensions

Funs.	Dims.	CS	GWO	BA	WOA	GOA	CSK	VCPSO	GLNPSO	ScPSO	RDWOA	HWOA	NB-WOA	EWOA	IWOA
F_{10}	30	0	0	0	2.56E-75	0	1.70E-75	2.27E-76	5.12E-76	0	0	0	0	0	0
	50	9.98E-79	3.84E-85	0	0	0	0	0	0	0	0	0	0	0	0
F_{11}	30	7.64E-05	3.68E+01	1.52E-05	1.58E-03	2.27E-14	6.80E-04	1.30E-04	9.57E-76	7.94E-05	0	0	0	0	0
	50	8.58E-03	4.38E+03	1.58E-03	2.27E-14	6.80E-04	1.45E-02	0	6.78E-03	0	0	0	0	0	0
F_{12}	30	3.89E-15	1.59E-04	1.67E-13	2.34E-16	1.19E-13	1.31E-14	1.00E-30	1.24E-13	1.05E-30	9.01E-31	2.60E-31	2.34E-31	2.58E-33	9.33E-43
	50	9.72E-16	2.65E-05	5.55E-14	2.93E-17	2.29E-14	3.29E-15	2.66E-32	2.49E-14	3.27E-32	1.72E-32	8.13E-33	2.11E-32	4.05E-33	1.16E-34
F_{13}	30	9.29E-08	1.53E-07	2.15E-07	1.50E-07	2.51E-07	6.30E-08	1.54E-08	8.50E-08	1.86E-08	6.01E-09	3.66E-09	4.26E-09	1.91E-12	8.21E-09
	50	9.46E-08	1.54E-07	2.18E-07	1.45E-04	3.05E-07	1.47E-07	1.55E-08	1.08E-07	2.61E-08	1.71E-08	8.06E-09	1.57E-09	5.67E-09	8.28E-10
F_{14}	30	1.40E-04	2.49E-04	2.06E-05	1.22E-03	1.46E-05	1.06E-05	7.48E-07	1.02E-05	7.52E-06	6.78E-06	4.73E-06	4.48E-07	4.79E-09	1.79E-10
	50	4.00E-03	1.61E-02	4.31E-04	1.83E-10	8.75E-06	2.72E-04	5.35E-07	1.71E-04	1.63E-04	7.32E-06	4.01E-06	0	1.72E-08	2.92E-05
F_{15}	30	1.15E-07	1.89E-07	2.62E-07	1.86E-07	3.10E-07	1.81E-09	1.90E-08	1.04E-07	3.10E-08	5.91E-09	5.66E-09	3.29E-09	3.61E-09	9.50E-10
	50	1.33E-07	2.42E-07	2.68E-07	2.45E-07	3.90E-07	1.29E-07	1.33E-08	1.06E-07	9.45E-09	5.75E-09	4.06E-09	6.13E-09	1.33E-09	1.04E-08
F_{16}	30	1.65E-07	2.67E-07	3.26E-07	3.36E-07	4.72E-07	3.87E-08	2.41E-08	1.29E-07	7.31E-08	3.48E-08	1.29E-08	9.87E-09	1.16E-09	1.10E-10
	50	3.11E-07	6.45E-07	4.11E-07	2.71E-07	4.98E-07	1.25E-07	2.55E-08	1.62E-07	4.84E-08	2.40E-08	2.70E-09	1.15E-09	6.70E-09	1.16E-10
F_{17}	30	1.81E+01	1.62E+02	4.69E-09	1.43E-10	2.70E-09	1.07E+02	3.70E-10	9.94E+01	2.98E-09	1.91E-09	1.47E-09	3.28E-11	2.97E-11	5.37E-12
	50	1.29E+00	2.31E+01	9.37E-10	3.58E-11	3.41E-10	5.47E+00	4.35E-11	2.90E+00	5.32E-11	4.71E-11	1.60E-11	1.48E-11	1.54E-12	1.34E-12

Best values are present in bold

Table 4 Mean ranking of all the considered methods

Rank	Methods	Rank value
1.	IWOA	1.84
2.	EWOA	1.98
3.	RDWOA	2.07
4.	NB-WOA	2.15
5.	ScPSO	2.38
6.	HWOA	2.47
7.	VCPSO	2.72
8.	GLNPSO	2.96
9.	GWO	3.18
10.	CSK	3.37
11.	BA	3.68
12.	WOA	3.83
13.	GOA	4.08
14.	CS	4.25

Best values are present in bold

considered models. Hence, the experimental and statistical outcomes reflect the high efficiency of IWOA.

Performance estimation of stance detection method

The performance of the proposed IWOASD method has been analyzed on five stance datasets namely, argument reasoning comprehension (ARC) dataset [31], headline and article bodies dataset (FNC-1) [52], claim polarity dataset Research [67], perspective dataset [69], and Snopes corpus [30]. The ARC dataset is manually created by Habernal et al. [29] and it consists of 188 debate topics from the user debate section of the New York Times, while the FNC-1 dataset is obtained from Fake News Challenge (FNC-1). Both ARC and FNC-1 datasets are annotated in four categories, namely, unrelated, discuss, agree, and disagree. On the other hand, claim polarity, perspective, and snopes corpus are annotated in two classes, viz, support (agree) and contest (disagree). All the datasets consist of pairs of claims and evidence texts along with stances. Stance in data may belong to any one of the categories, i.e., unrelated, discuss, agree, or disagree. The complete description of datasets is given in the following subsections.

FNC-1 dataset

This dataset is motivated by an Emergent dataset that originated from a digital journalism project at Columbia University [26]. The emergent dataset comprises of 2595 news articles and 300 rumored claims and grouped into true, false, and unverified categories by journalists. The FNC-1 dataset [77] expands the Emergent dataset by allocating four labels to

each headline–body pair, namely, unrelated, discuss, agree, and disagree.

ARC dataset

This dataset contains typical disputed subjects namely, schooling challenges, immigration, and international affairs from different news domains. This dataset is equivalent to the FNC-1 dataset with some considerable differences. In the FNC-1 dataset, the news articles are more balanced and complete, while in the ARC dataset, multisentence statement represents users' perspective of a topic. Table 5 tabulates the complete statistics of FNC-1 and ARC datasets. It can be visualized from the table that both the datasets are skewed. In the literature, it has been shown that the poor models show unsatisfactory results on skewed datasets [13,57]. Therefore, the efficacy of the proposed IWOA has been evaluated on skewed (imbalanced) datasets to show its performance.

Claim polarity dataset

This dataset contains 2394 evidences and claims for 55 topics Research [67]. Topics in the dataset were randomly selected from the debate motions database [8]. In the dataset, all the claims are manually annotated in two classes, namely, support (agree) and contest (disagree). This dataset comprises 1324 support claims and 1070 contest claims.

Perspectrum dataset

This dataset contains the users' views and perspective from various debate websites such as debatewise.org, idebate.com, and procon.org [66]. Each claim in the dataset has different views and stances. The dataset consists of 6125 supporting (agree) claims and 5751 opposing (disagree) claims [66,69]. For a fair comparison, the dataset is divided into three parts, i.e., training data, dev data, and test data. The complete dataset statistics have been tabulated in Table 5.

Snopes dataset

This dataset has been collected from the Snopes platform and consists of 8291 claim [30,33]. The dataset is annotated into two classes, namely, agree and refutes. An instance in the dataset is annotated as agree (support) if the claim is supported by evidence text; otherwise, it is annotated as refute or disagree. There is a total of 6178 agree instances and 2113 refute instances in the dataset [33].

To assess the efficiency of the proposed IWOASD, the first datasets are partitioned into training and testing parts in which each instance of the dataset contains a pair of claim and evidence texts. Since stop words, fuzzy words do not contain relevant information. Therefore, the stance datasets

Table 5 The dataset statistics

Dataset	Data types	Attribute types	Number of samples	Unrelated (%)	Discuss (%)	Agree (%)	Disagree (%)	Date and time	Associated tasks
FNC-1 dataset	Multivariate, text	Real	49,972	73.13	17.83	7.36	1.68	2017-06-22	Classification
ARC dataset	Multivariate, text	Real	17,792	75.0	6.1	8.9	10.0	2017-06-22	Classification
Claim polarity dataset	Multivariate, text	Real	2394	-	-	55.30	44.70	2020-08-01	Classification
Perspectrum dataset	Multivariate, text	Real	11,876	-	-	51.60	48.40	2020-08-01	Classification
Snopes dataset	Multivariate, text	Real	8291	-	-	74.50	24.50	2020-08-01	Classification

are preprocessed for eliminating unwanted words such as ‘a’, ‘is’, ‘are’, etc. from the datasets. Afterward, vocabulary lists are built for distinct terms occurring in headlines and article texts which are further used for extracting pre-trained word embeddings from the Stanford Glove corpus. The pre-trained word embeddings obtained from training sets are used for training the proposed IWOASD method. Finally, the test dataset is fed to a trained model to examine the efficiency of the proposed IWOA-based MLP.

The efficacy of the proposed IWOASD model is evaluated in respect of mean classification accuracy, mean error, standard deviation, and mean computational time for different numbers of hidden nodes (5, 7, 9, . . . , 20), and compared with CS, GWO, BA, WOA, and CSK. Mean classification accuracy is computed using a confusion matrix. In the confusion matrix C_x of size $m \times m$, C_{kk} represents the number of samples of class k predicted to same class, i.e., k . In the confusion matrix, diagonal entries represent correctly predicted entries. The mean accuracy, mean error, standard deviation values, and mean computational time of the proposed and state-of-the-art methods have been tabulated in Tables 6, 7, 8, 9, and 10. From the tables, it can be observed that the proposed IWOA with 20 hidden nodes returns the highest accuracy on the FNC-1 dataset and snopes corpus, while the proposed IWOA with 11 hidden nodes attains the highest accuracy over the ARC, claim polarity, and perspectrum datasets. The proposed IWOA also surpasses other methods in terms of mean error, while for the performance measure standard deviation, other methods such as CS, BA, and CSK show better results than the proposed method. Moreover, the proposed IWOA also performs better than the considered methods for performance criterion and computational time.

Additionally, to validate the efficacy of the proposed IWOASD, it is also compared with baseline along with state-of-the-art and recent variants of WOA such as TF-IDF, Doc2Vec + GRU-GRU, Glove + EWOA, Glove + RDWOA, etc., in terms of mean classification accuracy, mean error, standard deviation, and mean computational time. Table 11 depicts the results of all the considered methods. It is observed from the table that the proposed IWOA with Glove word embedding attains 76.53, 74.72, 78.45, 79.63, and 59.02% accuracy over FNC-1, ARC, claim polarity, perspectrum, and snopes corpus, respectively. Besides, the proposed method also outperforms other methods in terms of mean error. However, Doc2Vec + GRU-GRU shows the least variation over all the considered datasets except the FNC-1 dataset on which Tf-idf with GRU-GRU performs the best.

Furthermore, the confusion matrix is also presented in Table 12 to know the number of correctly predicted stances by the proposed method. From the tables, it can be perceived that the proposed IWOASD shows poor performance if all four stances are considered. For the FNC-1 dataset, only 14 instances of the agreed and 5 instances of disagree

Table 6 Comparison of mean accuracy, standard deviation, mean error, and execution time over FNC-1 dataset for the different numbers of hidden nodes

Hidden node(s)	Algorithm	Mean accuracy	Standard deviation	Mean error	Execution time
5	CS	69.17	2.04	0.3484	15 min 58 s
	GWO	70.54	2.32	0.3046	17 min 51 s
	BA	69.54	2.06	0.3237	18 min 27 s
	WOA	70.54	2.96	0.3360	19 min 02 s
	CSK	71.54	2.76	0.2931	18 min 25 s
	Proposed IWOA	75.97	3.06	0.2408	13 min 03 s
7	CS	68.75	2.34	0.3460	26 min 22 s
	GWO	69.71	2.32	0.3022	28 min 10 s
	BA	69.13	2.04	0.3213	28 min 43 s
	WOA	69.71	2.96	0.3336	29 min 19 s
	CSK	71.13	2.76	0.2907	28 min 56 s
	Proposed IWOA	76.04	3.06	0.2384	23 min 27 s
9	CS	69.22	2.32	0.3463	36 min 45 s
	GWO	70.67	2.29	0.3025	38 min 37 s
	BA	69.87	2.02	0.3216	38 min 51 s
	WOA	70.80	2.92	0.3339	39 min 43 s
	CSK	71.52	2.73	0.2910	39 min 06 s
	Proposed IWOA	75.83	3.03	0.2387	33 min 32 s
11	CS	68.99	2.33	0.3462	56 min 34 s
	GWO	70.19	2.31	0.3024	58 min 24 s
	BA	69.50	2.03	0.3215	58 min 47 s
	WOA	70.26	2.94	0.3338	59 min 31 s
	CSK	71.32	2.14	0.2909	59 min 31 s
	Proposed IWOA	75.94	3.04	0.2385	53 min 30 s
13	CS	69.07	2.33	0.3472	66 min 45 s
	GWO	70.28	2.30	0.3034	68 min 37 s
	BA	69.58	2.02	0.3225	68 min 51 s
	WOA	70.34	2.93	0.3348	69 min 43 s
	CSK	71.41	2.73	0.2919	69 min 06 s
	Proposed IWOA	76.03	3.03	0.2395	63 min 32 s
15	CS	69.01	2.33	0.3467	76 min 22 s
	GWO	70.21	2.30	0.3029	78 min 10 s
	BA	69.52	2.03	0.3220	78 min 43 s
	WOA	70.28	2.94	0.3343	79 min 19 s
	CSK	71.34	2.74	0.2914	78 min 56 s
	Proposed IWOA	75.96	3.04	0.2390	73 min 27 s
20	CS	69.60	2.33	0.3467	87 min 15 s
	GWO	70.57	2.30	0.3029	88 min 53 s
	BA	69.98	2.02	0.3220	89 min 15 s
	WOA	70.67	2.93	0.3343	89 min 58 s
	CSK	72.01	2.73	0.2914	89 min 39 s
	Proposed IWOA	76.98	3.03	0.2390	83 min 46 s

Best values are present in bold

Table 7 Comparison of mean accuracy, standard deviation, mean error, and execution time over ARC dataset for the different numbers of hidden nodes

Hidden node(s)	Algorithm	Mean accuracy	Standard deviation	Mean error	Execution time
5	CS	68.54	1.16	0.2084	7 min 29 s
	GWO	67.99	1.07	0.2969	8 min 24 s
	BA	66.11	1.68	0.2754	9 min 12 s
	WOA	64.21	1.96	0.2969	9 min 02 s
	CSK	67.05	1.66	0.2653	9 min 12 s
	Proposed IWOA	74.83	1.89	0.1935	7 min 01 s
7	CS	68.56	1.16	0.2072	17 min 22 s
	GWO	68.00	1.07	0.2951	18 min 12 s
	BA	66.11	1.67	0.2738	19 min 16 s
	WOA	64.22	1.95	0.2951	19 min 12 s
	CSK	67.06	1.65	0.2637	19 min 27 s
	Proposed IWOA	74.73	1.88	0.1924	17 min 10 s
9	CS	68.55	1.15	0.2064	27 min 15 s
	GWO	67.99	1.06	0.2940	28 min 19 s
	BA	66.11	1.66	0.2728	29 min 22 s
	WOA	64.22	1.94	0.2940	29 min 19 s
	CSK	67.05	1.64	0.2627	29 min 04 s
	Proposed IWOA	74.72	1.87	0.1917	27 min 13 s
11	CS	68.70	1.15	0.2064	37 min 16 s
	GWO	68.15	1.06	0.2940	38 min 20 s
	BA	66.26	1.66	0.2728	39 min 31 s
	WOA	64.36	1.94	0.2940	39 min 17 s
	CSK	67.20	1.64	0.2627	39 min 16 s
	Proposed IWOA	74.89	1.87	0.1917	37 min 18 s
13	CS	68.59	1.16	0.2067	47 min 25 s
	GWO	68.03	1.07	0.2945	48 min 19 s
	BA	66.15	1.67	0.2731	49 min 32 s
	WOA	64.25	1.95	0.2945	49 min 20 s
	CSK	67.09	1.65	0.2631	49 min 02 s
	Proposed IWOA	74.76	1.88	0.1920	47 min 11 s
15	CS	68.64	1.15	0.2069	57 min 11 s
	GWO	68.09	1.06	0.2948	58 min 06 s
	BA	66.20	1.66	0.2735	59 min 21 s
	WOA	64.31	1.95	0.2948	59 min 11 s
	CSK	67.15	1.64	0.2634	59 min 21 s
	Proposed IWOA	74.83	1.88	0.1922	57 min 13 s
20	CS	68.60	1.16	0.2070	67 min 15 s
	GWO	68.04	1.06	0.2949	68 min 26 s
	BA	66.16	1.67	0.2735	69 min 10 s
	WOA	64.26	1.95	0.2949	69 min 28 s
	CSK	67.10	1.65	0.2635	69 min 27 s
	Proposed IWOA	74.78	1.88	0.1923	67 min 23 s

Best values are present in bold

Table 8 Comparison of mean accuracy, standard deviation, mean error, and execution time over claim polarity dataset for the different numbers of hidden nodes

Hidden node(s)	Algorithm	Mean accuracy	Standard deviation	Mean error	Execution time
5	CS	71.43	0.92	0.2623	5 min 22 s
	GWO	72.85	1.04	0.2477	5 min 10 s
	BA	71.81	0.93	0.2584	5 min 43 s
	WOA	72.85	1.33	0.2477	5 min 19 s
	CSK	73.88	1.24	0.2370	4 min 56 s
	Proposed IWOA	78.45	1.19	0.1898	4 min 27 s
7	CS	71.00	0.98	0.2668	11 min 45 s
	GWO	71.99	1.11	0.2566	11 min 37 s
	BA	71.39	0.99	0.2627	10 min 51 s
	WOA	71.99	1.42	0.2566	11 min 43 s
	CSK	73.46	1.32	0.2414	10 min 26 s
	Proposed IWOA	78.53	1.27	0.1891	10 min 12 s
9	CS	71.48	0.88	0.2618	22 min 15 s
	GWO	72.98	1.00	0.2463	23 min 53 s
	BA	72.15	0.89	0.2549	22 min 15 s
	WOA	73.12	1.27	0.2449	22 min 58 s
	CSK	73.86	1.19	0.2373	23 min 39 s
	Proposed IWOA	78.31	1.14	0.1913	22 min 26 s
11	CS	71.33	0.96	0.2634	30 min 45 s
	GWO	72.58	1.09	0.2505	30 min 37 s
	BA	71.86	0.97	0.2580	30 min 51 s
	WOA	72.64	1.39	0.2498	30 min 43 s
	CSK	73.75	1.30	0.2384	30 min 06 s
	Proposed IWOA	78.72	1.25	0.1870	29 min 32 s
13	CS	71.25	1.00	0.2642	40 min 45 s
	GWO	72.49	1.14	0.2514	40 min 37 s
	BA	71.77	0.99	0.2588	40 min 51 s
	WOA	72.56	1.45	0.2507	40 min 43 s
	CSK	73.65	1.35	0.2394	40 min 06 s
	Proposed IWOA	78.42	1.30	0.1901	39 min 32 s
15	CS	71.27	0.99	0.2640	55 min 34 s
	GWO	72.51	1.13	0.2512	54 min 24 s
	BA	71.79	1.00	0.2586	54 min 47 s
	WOA	72.58	1.44	0.2505	54 min 31 s
	CSK	73.67	1.35	0.2392	55 min 31 s
	Proposed IWOA	78.33	1.29	0.1911	54 min 30 s
20	CS	71.88	0.95	0.2577	62 min 22 s
	GWO	72.88	1.08	0.2474	62 min 10 s
	BA	72.27	0.96	0.2537	61 min 43 s
	WOA	72.98	1.38	0.2463	62 min 19 s
	CSK	74.36	1.29	0.2320	61 min 56 s
	Proposed IWOA	78.46	1.24	0.1897	61 min 27 s

Best values are present in bold

Table 9 Comparison of mean accuracy, standard deviation, mean error, and execution time over perspectrum dataset for the different numbers of hidden nodes

Hidden node(s)	Algorithm	Mean accuracy	Standard deviation	Mean error	Execution time
5	CS	72.93	1.15	0.2707	7 min 15 s
	GWO	72.34	1.06	0.2766	8 min 26 s
	BA	70.34	1.66	0.2966	8 min 10 s
	WOA	68.32	1.94	0.3168	8 min 28 s
	CSK	71.34	1.64	0.2866	8 min 27 s
	Proposed IWOA	79.62	1.87	0.2038	7 min 23 s
7	CS	72.95	1.15	0.2705	17 min 11 s
	GWO	72.35	1.06	0.2765	18 min 06 s
	BA	70.34	1.65	0.2966	18 min 21 s
	WOA	68.33	1.93	0.3167	19 min 11 s
	CSK	71.35	1.63	0.2865	18 min 21 s
	Proposed IWOA	79.51	1.86	0.2049	17 min 13 s
9	CS	72.94	1.14	0.2706	27 min 25 s
	GWO	72.34	1.05	0.2766	28 min 19 s
	BA	70.34	1.64	0.2966	28 min 32 s
	WOA	68.33	1.92	0.3167	29 min 20 s
	CSK	71.34	1.62	0.2866	28 min 02 s
	Proposed IWOA	79.50	1.85	0.2050	27 min 11 s
11	CS	73.10	1.14	0.2690	37 min 22 s
	GWO	72.51	1.05	0.2749	38 min 12 s
	BA	70.50	1.64	0.2950	39 min 16 s
	WOA	68.48	1.92	0.3152	38 min 12 s
	CSK	71.50	1.62	0.2850	39 min 27 s
	Proposed IWOA	79.68	1.85	0.2032	38 min 10 s
13	CS	72.98	1.15	0.2702	47 min 15 s
	GWO	72.38	1.06	0.2762	48 min 19 s
	BA	70.38	1.65	0.2962	49 min 22 s
	WOA	68.36	1.93	0.3164	49 min 19 s
	CSK	71.38	1.63	0.2862	49 min 04 s
	Proposed IWOA	79.54	1.86	0.2046	47 min 13 s
15	CS	73.03	1.14	0.2697	57 min 29 s
	GWO	72.45	1.05	0.2755	58 min 24 s
	BA	70.44	1.64	0.2956	58 min 12 s
	WOA	68.43	1.93	0.3157	59 min 02 s
	CSK	71.45	1.62	0.2855	58 min 12 s
	Proposed IWOA	79.62	1.86	0.2038	57 min 01 s
20	CS	72.93	1.15	0.2707	67 min 16 s
	GWO	72.34	1.05	0.2766	68 min 20 s
	BA	70.34	1.65	0.2966	68 min 31 s
	WOA	68.32	1.93	0.3168	69 min 17 s
	CSK	71.34	1.63	0.2866	68 min 16 s
	Proposed IWOA	79.62	1.86	0.2038	67 min 18 s

Best values are present in bold

Table 10 Comparison of mean accuracy, standard deviation, mean error, and execution time over snopes annotated corpus for the different numbers of hidden nodes

Hidden node(s)	Algorithm	Mean accuracy	Standard deviation	Mean error	Execution time
5	CS	53.79	1.32	0.4621	5 min 15 s
	GWO	54.86	1.49	0.4514	5 min 53 s
	BA	54.07	1.33	0.4593	5 min 15 s
	WOA	54.86	1.90	0.4514	5 min 58 s
	CSK	55.63	1.77	0.4437	4 min 39 s
	Proposed IWOA	59.07	1.70	0.4093	4 min 26 s
7	CS	53.46	1.40	0.4654	11 min 45 s
	GWO	54.21	1.59	0.4579	11 min 37 s
	BA	53.76	1.42	0.4624	10 min 51 s
	WOA	54.21	2.03	0.4579	11 min 43 s
	CSK	55.32	1.89	0.4468	10 min 06 s
	Proposed IWOA	59.13	1.82	0.4087	10 min 32 s
9	CS	53.82	1.26	0.4618	22 min 22 s
	GWO	54.95	1.43	0.4505	23 min 10 s
	BA	54.33	1.27	0.4567	22 min 43 s
	WOA	55.06	1.82	0.4494	22 min 19 s
	CSK	55.62	1.70	0.4438	23 min 56 s
	Proposed IWOA	58.97	1.63	0.4103	22 min 27 s
11	CS	53.65	1.43	0.4635	33 min 34 s
	GWO	54.58	1.63	0.4542	33 min 24 s
	BA	54.04	1.42	0.4596	32 min 47 s
	WOA	54.64	2.07	0.4536	32 min 31 s
	CSK	55.46	1.93	0.4454	31 min 31 s
	Proposed IWOA	59.05	1.86	0.4095	30 min 30 s
13	CS	53.71	1.37	0.4629	40 min 45 s
	GWO	54.65	1.56	0.4535	40 min 37 s
	BA	54.11	1.39	0.4589	40 min 51 s
	WOA	54.70	1.99	0.4530	40 min 43 s
	CSK	55.53	1.86	0.4447	40 min 26 s
	Proposed IWOA	59.28	1.79	0.4072	39 min 12 s
15	CS	53.67	1.42	0.4633	55 min 22 s
	GWO	54.60	1.62	0.4540	54 min 10 s
	BA	54.06	1.43	0.4594	54 min 43 s
	WOA	54.65	2.06	0.4535	54 min 19 s
	CSK	55.47	1.93	0.4453	55 min 56 s
	Proposed IWOA	58.08	1.84	0.4102	54 min 27 s
20	CS	54.13	1.36	0.4587	62 min 45 s
	GWO	54.88	1.54	0.4512	62 min 37 s
	BA	54.42	1.37	0.4558	61 min 51 s
	WOA	54.95	1.97	0.4505	62 min 43 s
	CSK	55.99	1.84	0.4401	61 min 06 s
	Proposed IWOA	59.48	1.77	0.4092	61 min 32 s

Best values are present in bold

Table 11 Comparison of proposed IWOA-based neural network model with the state-of-the-art models

Dataset	Model	Mean accuracy (%)	Standard deviation	Mean error	Execution time
FNC-1 dataset	Tf-idf + GRU-GRU	67.17	1.34	0.3284	15 min 58 s
	Doc2Vec + GRU-GRU	68.54	2.12	0.3146	18 min 09 s
	Glove + GRU-GRU	68.54	2.12	0.3146	18 min 10 s
	Tf-idf + SMLP	68.54	2.12	0.3146	18 min 11 s
	Doc2Vec + SMLP	68.54	2.12	0.3146	18 min 09 s
	AtheneMLP	71.29	1.87	0.2871	19 min 18 s
	BERT base	75.84	1.49	0.2416	17 min 23 s
	Glove + EWOA	73.26	3.36	0.2832	17 min 26 s
	Glove + RDWOA	72.49	3.42	0.2726	16 min 24 s
	Glove + NB-WOA	72.68	3.13	0.2792	17 min 29 s
ARC dataset	Glove + proposed IWOA	76.53	3.05	0.2347	14 min 56 s
	Tf-idf + GRU-GRU	66.34	1.15	0.3359	8 min 35 s
	Doc2Vec + GRU-GRU	65.81	1.06	0.3334	9 min 24 s
	Glove + GRU-GRU	63.98	1.66	0.3721	8 min 49 s
	Tf-idf + SMLP	62.15	1.94	0.3634	9 min 03 s
	Doc2Vec + SMLP	64.89	1.64	0.3621	8 min 01 s
	AtheneMLP	70.43	1.69	0.2967	8 min 44 s
	BERT base	71.05	1.49	0.3016	8 min 19 s
	Glove + EWOA	72.19	1.96	0.2943	9 min 01 s
	Glove + RDWOA	71.86	2.01	0.3015	8 min 26 s
Claim polarity dataset	Glove + NB-WOA	71.91	2.03	0.2982	8 min 38 s
	Glove + proposed IWOA	74.72	1.87	0.1913	7 min 47 s
	Tf-idf + GRU-GRU	69.66	1.45	0.3142	5 min 12 s
	Doc2Vec + GRU-GRU	70.1	0.94	0.2915	5 min 07 s
	Glove + GRU-GRU	69.18	1.16	0.3126	4 min 59 s
	Tf-idf + SMLP	68.26	1.28	0.3178	5 min 06 s
	Doc2Vec + SMLP	71.13	1.13	0.2916	4 min 14 s
	AtheneMLP	71.33	1.02	0.2866	4 min 37ec
	BERT base	79.18	0.92	0.2108	5 min 11 s
	Glove + EWOA	78.83	1.13	0.2432	5 min 06 s
Perspectrum dataset	Glove + RDWOA	76.28	0.96	0.2506	5 min 16 s
	Glove + NB-WOA	77.06	1.15	0.2496	4 min 38 s
	Glove + proposed IWOA	78.45	1.01	0.2198	4 min 42 s
	Tf-idf + GRU-GRU	69.82	1.15	0.2059	8 min 35 s
	Doc2Vec + GRU-GRU	70.26	1.06	0.2934	9 min 24 s
	Glove + GRU-GRU	65.34	1.66	0.2721	8 min 49 s
	Tf-idf + SMLP	66.41	1.94	0.2934	9 min 03 s
	Doc2Vec + SMLP	70.29	1.64	0.2621	8 min 01 s
	AtheneMLP	70.06	1.43	0.2991	8 min 24 s
	BERT base	79.32	1.41	0.2415	8 min 38 s
Snopes annotated corpus	Glove + EWOA	76.56	1.76	0.2365	8 min 17 s
	Glove + RDWOA	79.65	1.79	0.2265	8 min 23 s
	Glove + NB-WOA	78.11	1.66	0.2162	8 min 11 s
	Glove + proposed IWOA	79.63	1.87	0.1913	7 min 47 s
	Tf-idf + GRU-GRU	51.42	1.08	0.4959	5 min 49 s
	Doc2Vec + GRU-GRU	51.78	1.01	0.4912	5 min 35 s
	Glove + GRU-GRU	52.76	1.53	0.4834	5 min 31 s

Table 11 continued

Dataset	Model	Mean accuracy (%)	Standard deviation	Mean error	Execution time
	Tf-idf + SMLP	50.97	1.14	0.4904	6 min 09 s
	Doc2Vec + SMLP	53.68	1.93	0.4648	5 min 16 s
	AtheneMLP	55.49	1.46	0.4419	5 min 24 s
	BERT base	59.89	1.16	0.4063	5 min 41 s
	Glove + EWOA	56.32	1.36	0.4325	6 min 03 s
	Glove + RDWOA	55.18	1.48	0.4215	5 min 14 s
	Glove + NB-WOA	58.76	1.47	0.4401	5 min 16 s
	Glove + proposed IWOA	59.02	1.72	0.4101	4 min 39 s

Best values are present in bold

Table 12 Stance detection results

Dataset		Agree	Disagree	Discuss	Unrelated	Accuracy
FNC-1 dataset	Agree	14	0	833	72	Accuracy = 76.53
	Disagree	5	5	196	4	
	Discuss	23	2	1858	342	
	Unrelated	3	9	1444	7680	
ARC dataset	Agree	296	67	5	28	Accuracy = 74.71
	Disagree	38	332	45	30	
	Discuss	19	28	202	22	
	Unrelated	152	434	257	2493	
Claim polarity dataset	Agree	289	1035	–	–	Accuracy = 78.46
	Disagree	258	812	–	–	
Perspectrum dataset	Agree	1195	276	–	–	Accuracy = 79.63
	Disagree	289	1013	–	–	
Snopes annotated corpus	Agree	3475	2703	–	–	Accuracy = 59.02
	Disagree	695	1418	–	–	

Best values are present in bold

stances are correctly predicted. On the contrary, the proposed IWOASD has a better recognition rate on the ARC dataset than the FNC-1 dataset. For the two-class stance datasets such as claim polarity, perspectrum, and snopes corpus, the proposed method shows much better results. The performance of the proposed IWOASD method degrades for FNC-1 and ARC dataset, since these datasets contain instances of the discussed category. The above analysis indicates that the efficiency of stance detection algorithms depends upon the number and type of stance categories in the datasets.

Conclusion and future work

This article proposes a new variant of WOA named IWOA for automated stance detection of fake news. The proposed IWOA uses tournament selection and roulette wheel selection in alternate iterations to manage the trade-off between exploration and exploitation. The proposed IWOA has been

validated over 17 benchmark functions. Furthermore, an optimized neural network model based on the strength of improved whale optimization and multilayer perceptron neural network has been presented for the stance detection of fake news. The proposed optimized neural network updates weights and bias of the FFNN using an improved whale optimization algorithm. The performance of the proposed model has been tested on five stance detection datasets and compared with CS, GWO, BA, WOA, and CSK. The proposed IWOA-optimized FFNN model achieves the highest accuracy as compared to the other considered models. Moreover, the proposed model also outperforms other considered models for the performance measures mean error and execution time for more than 80% of the datasets. The proposed IWOA-optimized neural network shows better results; however, improvement is still required. The proposed IWOA employs random choices to find the optimal solution, which means that the computing time and the solution quality are actually random variables. Due to this stochastic nature, its

rigorous analysis would be very difficult. Moreover, it cannot be guaranteed that the solution found in different runs will be globally optimal or of high quality. To mitigate the same, IWOA has been executed 30 times and its mean value has been used for comparison. From the results, it has been found that the proposed IWOA shows much better performance than other algorithms on single-objective problems. However, for the multi-objective problems, the proposed IWOA and other algorithms show the same performance. In future work, feature selection methods and different optimization methods can be explored for improving the accuracy. Furthermore, deep learning models such as CNN, LSTM, and BiLSTM could be also investigated. Besides, transfer learning and multitask methodologies could be also considered to exploit knowledge from other related domains.

Acknowledgements The author is also thankful to Dr. Himanshu Mittal, Assistant Professor at Jaypee Institute of Information Technology, Noida, India for tendering his valuable help for implementing the proposed method.

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

References

1. Abdel-Basset M, Manogaran G, El-Shahat D, Mirjalili S (2018) A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem. *Fut Gener Comput Syst* 85:129–145
2. Abualigah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering. Springer, Berlin
3. Akbarifard S, Radmanesh F (2018) Predicting sea wave height using symbiotic organisms search (SOS) algorithm. *Ocean Eng* 167:348–356
4. Akyol S, Alatas B (2020) Sentiment classification within online social media using whale optimization algorithm and social impact theory based optimization. *Phys A: Stat Mech Appl* 540:123094
5. Aljarah I, Faris H, Mirjalili S (2018) Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Comput* 22:1–15
6. Alzaqebah A, Masadeh R, Hudaib A (2018) Whale optimization algorithm for requirements prioritization. In: Proceedings of 9th International conference on information and communication systems (ICICS), IEEE, pp 84–89
7. Bansal P, Gupta S, Kumar S, Sharma S, Sharma S (2019) MLP-LOA: a metaheuristic approach to design an optimal multilayer perceptron. *Soft Comput* 23:12331–12345
8. Bar-Haim R, Bhattacharya I, Dinuzzo F, Saha A, Slonim N (2017) Stance classification of context-dependent claims. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Long Papers, pp 251–261
9. Borges L, Martins B, Calado P (2019) Combining similarity features and deep representation learning for stance detection in the context of checking fake news. *J Data Inf Qual* 11:1–26
10. Bozorg-Haddad O, Azarnivand A, Hosseini-Moghari SM, Loáiciga HA (2017) Optimal operation of reservoir systems with the symbiotic organisms search (SOS) algorithm. *J Hydroinform* 19(4):507–521
11. Chandra Pandey A, Singh Rajpoot D, Saraswat M (2018) Data clustering based on data transformation and hybrid step size-based cuckoo search. In: Proceedings of the eleventh international conference on contemporary computing (IC3), IEEE, pp 1–6
12. Chaudhry AK, Baker D, Thun-Hohenstein P (2017) Stance detection for the fake news challenge: identifying textual relationships with deep neural nets. In: Proceedings of the CS224n: natural language processing with deep learning, pp 1–10
13. Chawla NV, Japkowicz N, Kotcz A (2004) Special issue on learning from imbalanced data sets. *ACM SIGKDD Explor Newsl* 6:1–6
14. Chen JY, Johnson J, Yennie G (2017) RNNs for stance detection between news articles. Stanford University, California, US, Rep, pp 1–7
15. Chen H, Xu Y, Wang M, Zhao X (2019) A balanced whale optimization algorithm for constrained engineering design problems. *Appl Math Model* 71:45–59
16. Chen H, Yang C, Heidari AA, Zhao X (2020) An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst Appl* 154:113018
17. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
18. Chourasia S, Sharma H, Singh M, Bansal JC (2019) Global and local neighborhood based particle swarm optimization. In: *Harmony Search and Nature Inspired Optimization Algorithms*, Springer pp 449–460
19. Clerc M (2015) *Guided randomness in optimization*, vol 1. Wiley, New York, pp 1–293
20. Cuevas E, Cienfuegos M, Zaldívar D, Pérez-Cisneros M (2013) A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Syst Appl* 40(16):6374–6384
21. Davis R, Proctor C (2017) Fake news, real consequences: recruiting neural networks for the fight against fake news, pp 1–8
22. El-Bages M, Elsayed W (2017) Social spider algorithm for solving the transmission expansion planning problem. *Electr Power Syst Res* 143:235–243
23. Elsayed W, Hegazy Y, Bendary F, El-Bages M (2016) Modified social spider algorithm for solving the economic dispatch problem. *Eng Sci Technol Int J* 19(4):1672–1681
24. Faris H, Aljarah I, Al-Madi N, Mirjalili S (2016) Optimizing the learning process of feedforward neural networks using lightning search algorithm. *Int J Artif Intell Tools* 25:1–32
25. Féraud R, Clérot F (2002) A methodology to explain neural network classification. *Neural Netw* 15:237–246
26. Ferreira W, Vlachos A (2016) Emergent: a novel data-set for stance classification. In: Proceedings of the conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1163–1168
27. Ghai R, Kumar S, Pandey AC (2019) Spam detection using rating and review processing method. In: Panigrahi B, Trivedi M, Mishra K, Tiwari S, Singh P (eds), *Smart Innovations in Communication and Computational Sciences*. Advances in Intelligent Systems and Computing, vol 670. Springer, Singapore. https://doi.org/10.1007/978-981-10-8971-8_18
28. Ghanem B, Rosso P, Rangel F (2018) Stance detection in fake news a combined feature representation. In: Proceedings of the

- first workshop on Fact Extraction and Verification (FEVER), pp 66–71
29. Habernal I, Wachsmuth H, Gurevych I, Stein B (2017) The argument reasoning comprehension task: identification and reconstruction of implicit warrants, pp 1–11. arXiv preprint. [arXiv:1708.01425](https://arxiv.org/abs/1708.01425) [cs.CL]
 30. Hanselowski A (2020) Snopes corpus. <https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2081>
 31. Hanselowski A, PVS A, Schiller B, Caspelherr F, Chaudhuri D, Meyer CM, Gurevych I (2018) A retrospective analysis of the fake news challenge stance-detection task. In: Proceedings of the 27th international conference on computational linguistics (COLING 2018). <http://tubiblio.ulb.tu-darmstadt.de/105434/>
 32. Hanselowski A, PVS A, Schiller B, Caspelherr F, Chaudhuri D, Meyer CM, Gurevych I (2018) A retrospective analysis of the fake news challenge stance detection task, pp 1–16. arXiv preprint. [arXiv:1806.05180](https://arxiv.org/abs/1806.05180)
 33. Hanselowski A, Stab C, Schulz C, Li Z, Gurevych I (2019) A richly annotated corpus for different tasks in automated fact-checking, pp 1–11. arXiv preprint. [arXiv:1911.01214](https://arxiv.org/abs/1911.01214)
 34. Hassan MK, El Desouky AI, Elghamrawy SM, Sarhan AM (2019) A hybrid real-time remote monitoring framework with NB-WOA algorithm for patients with chronic diseases. *Future Gener Comput Syst* 93:77–95
 35. Hussien AG, Hassanien AE, Houssein EH, Amin M, Azar AT (2020) New binary whale optimization algorithm for discrete optimization problems. *Eng Optim* 52(6):945–959
 36. Jain L, Katarya R, Sachdeva S (2020) Opinion leader detection using whale optimization algorithm in online social network. *Expert Syst Appl* 142:113016
 37. Jiang T, Zhang C, Sun QM (2019) Green job shop scheduling problem with discrete whale optimization algorithm. *IEEE Access* 7:43153–43166
 38. Kohli M, Arora S (2018) Chaotic grey wolf optimization algorithm for constrained optimization problems. *J Comput Des Eng* 5(4):458–472
 39. Koyuncu H, Ceylan R (2015) Scout particle swarm optimization. In: Lacković I, Vasic D (eds), 6th European Conference of the International Federation for Medical and Biological Engineering. IFMBE Proceedings, vol 45. Springer, Cham. https://doi.org/10.1007/978-3-319-11128-5_21
 40. Kulhari A, Pandey A, Pal R, Mittal H (2016) Unsupervised data classification using modified cuckoo search method. In: Proceedings of the ninth international conference on contemporary computing (IC3). IEEE, pp 1–5
 41. Lau JH, Baldwin T (2016) An empirical evaluation of doc2vec with practical insights into document embedding generation, pp 1–9. arXiv preprint. [arXiv:1607.05368](https://arxiv.org/abs/1607.05368) [cs.CL]
 42. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the international conference on machine learning, pp 1188–1196
 43. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* 260:302–312
 44. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
 45. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
 46. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
 47. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. *Appl Intell* 48:805–820
 48. Mittal H, Saraswat M (2018) An optimum multi-level image thresholding segmentation using non-local means 2d histogram and exponential kbest gravitational search algorithm. *Eng Appl Artif Intell* 71:226–235
 49. Mohammed AS, Shukla V, Pandey AC (2020) Enhancing sentiment analysis using enhanced whale optimisation algorithm. *Int J Intell Inf Database Syst* 13(2–4):208–230
 50. Mohtarami M, Baly R, Glass J, Nakov P, Màrquez L, Moschitti A (2018) Automatic stance detection using end-to-end memory networks, pp 1–10. arXiv preprint. [arXiv:1804.07581](https://arxiv.org/abs/1804.07581) [cs.CL]
 51. Mosavi M, Khishe M, Ghamgosar A (2016) Classification of sonar data set using neural network trained by gray wolf optimization. *Neural Netw World* 26:393
 52. Rajamohan S, Romanella A, Ramesh A (2019) A weakly-supervised attention-based visualization tool for assessing political affiliation, pp 1–8, arXiv preprint arXiv
 53. Mukherjee S, Adhikari A, Roy M (2019) Malignant melanoma detection using multilayer perceptron with optimized network parameter selection by PSO. In: Proceedings Of Contemporary advances in innovative and applicable information technology, Springer, pp 101–109
 54. Nguyen TT (2019) A high performance social spider optimization algorithm for optimal power flow solution with single objective optimization. *Energy* 171:218–240
 55. Orhan U, Hekim M, Ozer M (2011) EEG signals classification using the k-means clustering and a multilayer perceptron neural network model. *Expert Syst Appl* 38:13475–13481
 56. Pal R, Saraswat M (2017) Data clustering using enhanced biogeography-based optimization. In: Proceedings of IEEE international conference on contemporary computing. IEEE, pp 1–6
 57. Pandey AC, Rajpoot DS (2019) Spam review detection using spiral cuckoo search clustering method. *Evol Intell* 12:147–164
 58. Pandey AC, Rajpoot DS (2020) Improving sentiment analysis using hybrid deep learning model. *Recent Adv Comput Sci Commun* 13:627–639
 59. Pandey AC, Rajpoot DS, Saraswat M (2016) Data clustering using hybrid improved cuckoo search method. In: Proceedings of ninth international conference on contemporary computing (IC3). IEEE, pp 1–6
 60. Pandey AC, Rajpoot DS, Saraswat M (2017) Hybrid step size based cuckoo search. In: Proceedings of tenth international conference on contemporary computing (IC3). IEEE, pp 1–6
 61. Pandey AC, Rajpoot DS, Saraswat M (2017) Twitter sentiment analysis using hybrid cuckoo search method. *Inf Process Manag* 53:764–779
 62. Pandey AC, Pal R, Kulhari A (2018) Unsupervised data classification using improved biogeography based optimization. *Int J Syst Assur Eng Manag* 9:821–829
 63. Pandey AC, Garg M, Rajput S (2019) Enhancing text mining using deep learning models. In: Proceedings of twelfth IEEE international conference on contemporary computing (IC3). IEEE, pp 1–5
 64. Petrović M, Miljković Z, Jokić A (2019) A novel methodology for optimal single mobile robot scheduling using whale optimization algorithm. *Appl Soft Comput* 81:105520
 65. Pfohl SR, Triebe O, Legros F (2017) Stance detection for the fake news challenge with attention and conditional encoding. CS224n: natural language processing with deep learning, pp 1–14
 66. Popat K, Mukherjee S, Yates A, Weikum G (2019) Stancy: Stance classification based on consistency cues, pp 1–6. arXiv preprint. [arXiv:1910.06048](https://arxiv.org/abs/1910.06048)
 67. Research I (2020) Claim polarity dataset. https://www.research.ibm.com/haifa/dept/vst/debating_data.shtml/
 68. Riedel B, Augenstein I, Spithourakis GP, Riedel S (2017) A simple but tough-to-beat baseline for the fake news challenge stance detection task, pp 1–6. arXiv preprint. [arXiv:1707.03264](https://arxiv.org/abs/1707.03264) [cs.CL]
 69. Schiller B, Daxenberger J, Gurevych I (2020) Perspectrum stance dataset. <https://www.groundai.com/project/stance-detection-benchmark-how-robust-is-your-stance-detection/>

70. Sepanski JH (2007) A modification on the Friedman test statistic. *Commun Stat Simul Comput* 36:783–790
71. Shu K, Sliva A, Wang S, Tang J, Liu H (2017) Fake news detection on social media: a data mining perspective. *ACM SIGKDD Explor Newsl* 19:22–36
72. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12:702–713
73. Sun M, Feng C, Chartan EK, Hodge BM, Zhang J (2019) A two-step short-term probabilistic wind forecasting methodology based on predictive distribution optimization. *Appl Energy* 238:1497–1505
74. Sun Q, Wang Z, Zhu Q, Zhou G (2018) Stance detection with hierarchical attention network. In: *Proceedings of 27th international conference on computational linguistics*, pp 2399–2409
75. Tejani GG, Savsani VJ, Patel VK, Mirjalili S (2018) Truss optimization with natural frequency bounds using improved symbiotic organisms search. *Knowl Based Syst* 143:162–178
76. Thorne J, Chen M, Myrianthous G, Pu J, Wang X, Vlachos A (2017) Fake news stance detection using stacked ensemble of classifiers. In: *Proceedings of EMNLP workshop: natural language processing meets journalism*, pp 80–83
77. Tosik M, Mallia A, Gangopadhyay K (2018) Debunking fake news one feature at a time, pp 1–5. arXiv preprint. [arXiv:1808.02831](https://arxiv.org/abs/1808.02831) [cs.CL]
78. Tubishat M, Abushariah MA, Idris N, Aljarah I (2019) Improved whale optimization algorithm for feature selection in arabic sentiment analysis. *Appl Intell* 49(5):1688–1707
79. Xu J, Shan G, Amei A, Zhao J, Young D, Clark S (2017) A modified Friedman test for randomized complete block designs. *Commun Stat Simul Comput* 46:1508–1519
80. Yang XS, Deb S (2014) Cuckoo search: recent advances and applications. *Neural Comput Appl* 24:169–174
81. Yang XS, Hossein Gandomi A (2012) Bat algorithm: a novel approach for global engineering optimization. *Eng Comput* 29:464–483
82. Yılmaz ÖF (2020) Operational strategies for seru production system: a bi-objective optimisation model and solution methods. *Int J Prod Res* 58(11):3195–3219
83. Yılmaz OF, Durmusoglu MB (2018) A performance comparison and evaluation of metaheuristics for a batch scheduling problem in a multi-hybrid cell manufacturing system with skilled workforce assignment. *J Ind Manag Optim* 14(3):1219
84. Yoon S, Park K, Shin J, Lim H, Won S, Cha M, Jung K (2019) Detecting incongruity between news headline and body text via a deep hierarchical encoder. In: *Proceedings of the AAAI conference on artificial intelligence*, pp. 791–800
85. Yu H, Yue C, Wang C (2016) News article summarization with attention-based deep recurrent neural networks
86. Zeng Q, Zhou Q, Xu S (2017) Neural stance detectors for fake news challenge. *CS224n: natural language processing with deep learning*, pp 1–9
87. Zhang Q, Liu W, Meng X, Yang B, Vasilakos AV (2017) Vector coevolving particle swarm optimization algorithm. *Inf Sci* 394:273–298
88. Zhang Q, Yılmaz E, Liang S (2018) Ranking-based method for news stance detection. In: *Proceedings of companion web conference, ACM*, pp 41–42
89. Zimmerman DW, Zumbo BD (1993) Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks. *J Exp Educ* 62:75–86

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