Iranian Journal of Fuzzy Systems

Volume 20, Number 3, (2023), pp. 61-74

Original paper

University of Sistan and Baluchestan

Weighted K-nearest neighbors classification based on Whale optimization algorithm

S. Anvari[1], M. Abdollahi Azgomi[2], M. R. Ebrahimi Dishabi [3] and M. Maheri[4]

[1,2,3,4] *Department of Computer Engineering, Miyaneh Branch, Islamic Azad University, Miyaneh, Iran*

shima.anvari@m-iau.ac.ir, azgomi@iust.ac.ir, mrebrahimy@m-iau.ac.ir, maheri@m-iau.ac.ir

Abstract

K-Nearest Neighbors (KNN) is a classification algorithm based on supervised machine learning, which works according to a voting system. The performance of the KNN algorithm depends on different factors, such as unbalanced distribution of classes, the scalability problem, and considering equal values for all training samples. Regarding the importance of the KNN algorithm, different improved versions of this algorithm are introduced, such as fuzzy KNN, weighted KNN, and KNN with variable neighbors. In this paper, a weighted KNN based on Whale Optimization Algorithm is proposed for the objective of increasing the level of detection accuracy. The proposed algorithm devotes a weight to each training sample of every feature by employing the WOA to explore the optimized weight matrix. The algorithm is implemented and experimented on five standard datasets. The evaluation results prove that the proposed algorithm performs better than both weighted KNN based on the Genetic Algorithm (GA) and the classic KNN algorithm.

Keywords: K-nearest neighbors, weighted K-nearest neighbors, whale optimization algorithm, genetic algorithm.

1 Introduction

The knowledge discovery process is an order of tasks and consists of data understanding, data preparation, modeling, evaluation, and deployment [28]. Data mining methods as a powerful tool are used for modeling in the knowledge discovery process [22]. These methods accomplish extracting useful knowledge from the dataset [2]. In general, data mining is the process of pattern discovery and extraction from a large amount of data [17]. Data mining methods are classified into two categories called supervised learning and unsupervised learning [18]. The primary aim of supervised learning is to build a descriptive model representing the labeled data. Classification is the most important technique of supervised learning, and also, the main purpose of unsupervised learning is to find the hidden structure in unlabeled data. Clustering is the most important technique of unsupervised learning.

In this paper, classification is mentioned as one of the effective methods of supervised learning in data mining. Classification is a process where a descriptive model can categorize data into specific classes. Classification is one of the most widely used techniques in the field of data mining. The primary goal of this technique is to classify large data into predefined classes that are described by a set of features [23, 32]. Classification uses the model to predict the class of unknown samples. So far, different classification algorithms in data mining have been proposed, the most important of which are Decision Tree (DT), Random Forest (RF), Bayesian Network (BN), Artificial Neural Network (ANN), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) [8].

KNN is a widely used supervised machine learning-based classification technique that is based on the majority voting principle [20]. KNN is chosen as an efficient classification algorithm for two major reasons: Firstly, the implementation of KNN is so simple. Secondly, the nature of an algorithm is non-parametric and has the capability of classifying a wide range of datasets. KNN is a simple efficient classification algorithm that works based on the distance of samples in the feature space. KNN predicts the class of a sample according to the class of its neighbors. Factors such as imbalanced

Corresponding Author: M. Abdollahi Azgomi
Received: May 2022; Revised: September 2022; Accepted: December 2022.
https://doi.org/10.22111/IJFS.2023.7639

distribution [20], scalability problems [6], and equal value of all training samples [33] affect the performance of the algorithm. To solve these problems, improved versions of the KNN have been proposed. These efficient versions of the KNN can be found in the groups of fuzzy KNN [4, 19, 25], weighted KNN [9, 10, 38], and KNN with variable neighbors [7, 13, 36].

In this paper, a weighted KNN based on the WOA is designed. In the proposed method, the WOA is used to improve the performance of the weighted KNN. The main purpose of developing a weighted KNN based on the WOA is to improve classification criteria such as detection accuracy. The proposed method has been evaluated on different datasets from the UCI machine learning repository [30].

The rest of this paper is organized as follows. The used algorithms are described in Section 2. The related works are presented in Section 3. The motivations and aims are described in Section 4. A weighted KNN based on the WOA is proposed to improve the accuracy in Section 5. Implementation results are detailed in Section 6 and finally, the conclusion and future works are explained in Section 7.

2 Background

2.1 K-nearest neighbor

KNN is a simple and effective algorithm for classification that is based on the proximity of training samples in the feature space [1]. In the KNN algorithm, the test sample class is predicted based on the neighbor's class. In fact, by voting among neighbors, the test sample class is determined. Euclidean distance is used to measure the similarity between the two samples. The Euclidean distance between the considered samples is mathematically formulized by Equation (1). Both samples have n properties that represent the n-dimensional space of the problem.

$$\|X - Y\|_2 = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{1}$$

In the KNN algorithm, the distance between each test sample and all training samples is calculated. Then, K training samples with the minimum distance as neighbors are selected. Finally, the test sample class is predicted based on the voting among the neighbors. This process is applied to all test samples, and the class of all samples is predicted.

2.2 Whale optimization algorithm

The WOA was first proposed in 2016 by Mirjalili [24]. The WOA is inspired by the social behavior of humpback whales. The whales move in a circular orbit to catch small fish and create bubbles along the circular orbit. Algorithm 1 shows the WOA pseudocode. The WOA is a population-based algorithm. The initial population in the WOA is generated randomly. Each whale represents a solution in the problem search space, and the quality of the whales is calculated based on fitness function. The WOA is an iterative algorithm whose purpose is to find the optimal solution. After getting the stop condition, the best whale of the final population is considered as the algorithm's output. In general, the WOA consists of two main phases of exploitation and exploration. In the exploitation step, the solutions are updated based on the best population solution, while in the exploration phase, the solutions are updated based on a random solution.

•Exploitation phase:

In the exploitation phase, the shrinking encircling mechanism and spiral-shaped path are modeled. In this phase, the solutions in the population are updated based on the best solution. Equation (2) is used to model the shrinking encircling mechanism.

$$\overrightarrow{X}(t+1) = \overrightarrow{X^*}(t) - \overrightarrow{A} \cdot \overrightarrow{D}. \tag{2}$$

The variable t represents the current iteration number and $\overrightarrow{X^*}$ represents the best solution. Also, $\overrightarrow{D}$ represents the distance, which Equation (3) shows how to be calculated. Finally, $\overrightarrow{A}$ and $\overrightarrow{C}$ are coefficient vectors that are defined based on Equations (4) and (5), respectively.

$$\overrightarrow{D} = |\overrightarrow{C} \cdot \overrightarrow{X^*}(t) - \overrightarrow{X}(t)|. \tag{3}$$

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r} - \overrightarrow{a}. \tag{4}$$

Algorithm 1 Pseudo-code of the WOA

Input:
Initial population with Size whales
Output:
$\overrightarrow{X^*}$
Calculate the fitness of each whale
Find the best solution among the initial population $\overrightarrow{X^*}$
while $t \leq$ max **do**
  for $i = 1$ to Size **do**
    Update a,l. and P
    Update $\overrightarrow{A}$ and $\overrightarrow{C}$
    if $P < 0.5$ **then**
      if $|\overrightarrow{A}| < 1$ **then**
        Update the position of the current whale by Equation 2
      else
        if $|\overrightarrow{A}| \geq 1$ **then**
          Update the position of the current whale by Equation 10
        end if
      end if
    end if
    if $P \geq 0.5$ **then**
      Update the position of the current whale by Equation 7
    end if
  end for
  Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of each whale
  Update $\overrightarrow{X^*}$
  $t = t + 1$
end while

$$\overrightarrow{C} = 2\overrightarrow{r}. \tag{5}$$

$\overrightarrow{r}$ is a random vector in $[0, 1]$ and $\overrightarrow{a}$ is a vector that is decreased linearly from 2 to 0. How to decrease the $\overrightarrow{a}$ in each iteration is shown in Equation (6). The variable max indicates the highest number of iterations.

$$a = 2 - t\frac{2}{\max}. \tag{6}$$

Equation (7) is used to model the spiral-shaped path. $\overrightarrow{D'}$ represents the distance, which is calculated by Equation (8). Also, b is a fixed number that represents the logarithmic spiral shape, and l is a random number in $[-1, 1]$.

$$\overrightarrow{X}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t), \tag{7}$$

$$\overrightarrow{D'} = |\overrightarrow{X^*}(t) - \overrightarrow{X}(t)|. \tag{8}$$

In the exploitation phase, a selection mechanism is applied between the shrinking encircling operation and the spiral-shaped path operation. For this purpose, a random number in $[0, 1]$ is generated. If the random number generated is less than 0.5, the shrinking encircling operation is used; otherwise, the spiral path operation is used. The selection mechanism in the exploitation phase is modeled based on Equation (9). P is a random number in $[0, 1]$.

$$\overrightarrow{X}(t+1) = \begin{cases} \text{Use Equation (2)} & \text{if } (P < 0.5) \\ \text{Use Equation (7)} & \text{if } (P \geq 0.5) \end{cases} \tag{9}$$

• **Exploration phase:**
In the exploration phase, the solutions are updated based on a random solution. Using a random solution improves the

algorithm's search ability. How to update solutions based on a random solution is shown in Equation (10).

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{rand}} - \overrightarrow{A} \cdot \overrightarrow{D}. \tag{10}$$

Where, $\overrightarrow{X_{rand}}$ indicates a random solution. Also, $\overrightarrow{D}$, which represents the distance, is updated based on Equation (11).

$$\overrightarrow{D} = |\overrightarrow{C} \cdot \overrightarrow{X_{rand}} - \overrightarrow{X}|. \tag{11}$$

3 Related work

Liao and Kuo [21] proposed five discrete Symbiotic Organism Search (SOS) algorithms to simultaneously optimize feature subset and neighborhood size in the KNN algorithm. Each organism represents a solution in the problem's search space that encodes feature subset and neighborhood size simultaneously. Zhang [35] proposed a cost-sensitive KNN algorithm for imbalanced datasets classification. In cost-sensitive learning, the cost of false positive and false negative is considered differently. In this paper, two strategies are proposed to solve this problem that can reduce misclassification costs. Also, these strategies have been further improved by combining smoothing, optimal neighborhood size selection, and feature selection techniques. Hu et al. [14] proposed a KNN algorithm optimized by the P system. The P system is made up of multiple cells that create a computational framework. In system P, two rules of evolution and communication are applied to select the optimal neighborhood size for each test sample. Experimental results demonstrate the effectiveness of the KNN algorithm optimized by the P system compared to the classical KNN algorithm. Jia and Zhang [16] proposed a KNN algorithm to augment the feature in multi-dimensional classification. In the multi-dimensional classification, each training sample is represented by a vector, each element representing the membership of a class. Multi-dimensional classification methods focus on modeling the class variables in the output space. Unlike previous methods, the solution of feature manipulation in input space is proposed in this method. Feature manipulation in the input space is performed using the KNN algorithm.

Wang et al. [31] developed an Improved n-nearest neighbor algorithm for an epilepsy diagnosis. In this method, a Fourier transform is used to convert the time-domain characteristics of the signal into the frequency domain. Then, a weighted KNN algorithm based on the Bray Curtis distance is designed to classify the information obtained. Delima [5] proposed an improved KNN algorithm based on the GA. The KNN algorithm is vulnerable to dataset noise. In this method, the GA is used to remove the dataset noise to enhance the performance of the KNN algorithm. Imron and Prasetiyo [15] proposed an improved KNN algorithm based on the Particle Swarm Optimization (PSO). In this method, the task of the PSO algorithm is to select the most optimal number of neighbors for the test samples. In fact, for each test sample, the numerical value of the variable K is selected optimally. Bian et al. [3] proposed a fuzzy KNN algorithm with adaptive the number of neighbors. In this method, fuzzy KNN trees such as random forest is used to find the optimal variable numerical value K for each test sample. Also, a strategy is designed to speed up the searching process in the nearest fuzzy neighbor trees. Sang et al. [29] presented a new algorithm for the KNN by a Bergman distance calculation. Utilizing the Bergman distance calculation in the KNN contributes to many advantages in the classification. Indeed, in this paper, a segmentation strategy is presented to address the problem of the KNN by using Bergman distances for high-dimensional datasets.

Harrou et al. [11] introduced an improved version of KNN for supervision of traffic congestion. In this paper, two new KNN-based mechanisms for detecting road traffic congestion are designed. The advantages of the modeling of the piecewise switched linear traffic and Kalman filter have been used in the improved version. Zeraatkar and Afsari [34] presented two generalized versions of the KNN for the classification of an unbalanced dataset. This approach consisted of two steps. In the first step, the data is over-sampled, and then the noisy and borderline samples are chosen and removed. Then, in the second step, two generalized versions of the KNN based on the concepts of interval-valued fuzzy and intuitionistic fuzzy sets are presented for the classification. Zhang et al. [37] introduced an improved fuzzy KNN based on GA. The GA is applied to calculate the initial k value and fuzzy parameters. Rhee and Hwang [27] introduced an interval type-2 fuzzy KNN that has been an advanced version of the fuzzy KNN algorithm. The result of the interval type-2 fuzzy KNN is more confident than the fuzzy KNN. Hashemi et al. [12] presented an improved K-nearest neighbor algorithm with the WOA for predicting liver disease, which is actually designed using a hybrid algorithm including KNN and WOA. Mukherjee et al [26] introduced an improved KNN algorithm based on IoT-cloud classification (eKNN) in order to identify the Covid-19 disease. The method was applied to seven datasets collected from countries such as Brazil, Mexico, etc. To evaluate the proposed method, experiments were conducted on an IoT-based cloud system that was built using disease prediction analysis. Research and results show that the proposed method works better than conventional KNN. Table 1 shows the analytical comparison of previous methods carried out in the field KNN.

Table 1: Analytical comparison of previous methods

Ref	Year	Main Challenges	Feature Selection	Source	Approach	Evaluation
[21]	2018	optimization of feature subset and neighborhood size	√	UCI	SOS	Classification Error, CPU Time
[36]	2020	Reduce costs of imbalanced data classification	√	UCI	Direct-CS-KNN, Distance-CS-KNN	False Positive, False Negative
[14]	2020	optimization of neighborhood size	×	UCI	P Systems	Classification Error, Accuracy
[16]	2020	solve the problem of multi-dimensional classification	×	UCI	feature manipulation in the input space	Standard Deviation
[31]	2020	Design a weighted KNN for disease diagnosis	×	EEG	Fourier transform, Bray Curtis distance	Classification Error, Accuracy
[5]	2020	Resisting dataset noise	√	Local	GA	Accuracy, RMSE, MAE
[15]	2020	Predicting the status of customers	×	UCI	Z-score, PSO	Accuracy
[3]	2022	Fuzzy KNN with Adaptive Nearest Neighbors	×	UCI	A random forest based strategy	Accuracy, Running Time
[29]	2020	improved KNN with Bergman distance	×	UCI	generate a non-metric distance function	Accuracy, Running Time
[11]	2020	Traffic congestion monitoring using an improved KNN	×	Local	switched linear traffic and Kalman filter	Accuracy
[34]	2021	imbalanced data classification	×	UCI	interval-valued fuzzy	The curve (AUC)
[35]	2008	classification of overlapping pattern data	×	UCI	Genetic Algorithm	Classification Error, Accuracy
[27]	2003	Improve fuzzy KNN	×	Local	interval type-2 fuzzy sets	Accuracy
[12]	2019	Detecting liver disease	×	Local	Whale Optimization Algorithm	Classification Error, Accuracy
[26]	2021	IoT-cloud-based COVID - 19 detection.	√	Local	Ant Colony Optimization	Accuracy, Running Time

4 Motivations and aims

The main motivation of designing a weighted KNN based on WOA is to improve classification criteria such as detection accuracy and classification error. In the weighted KNN, solving the weighting problem is considered as the main aim. Nature-inspired metaheuristic algorithms are suitable to solve the weighting problem. WOA is an efficient meta-heuristic algorithm inspired by the social behavior of whales.

- Presenting a new definition of the weighting problem in the weighted KNN.
- Presenting a new definition for distance calculation based on weights matrix
- Optimizing weight matrix based on WOA.

5 The proposed method

In this section, details of the weighted KNN based on WOA are introduced.

5.1 Concepts

For designing a weighted KNN based on WOA, basic concepts must be defined. The main dataset is composed of m samples where each one is described by n features. Therefore, the main dataset is shown by a $m \times n$ matrix. Moreover, the class of all the samples is determined. The main dataset is divided into training and test datasets. The training

dataset is composed of 80% of samples while the rest 20% of samples are devoted to the test dataset. The training dataset is shown by a $m_1 \times n$ matrix. Also, the test dataset is a $m_2 \times n$ matrix. Finally, the weight matrix is a matrix with the dimensions of m_1 and n that determines the weight of train samples for each feature. The weight matrix is shown with variable W where w_{ij} is the weight of sample number i for feature number j. Each element of the weight matrix is a value in the interval of $[0, 1]$. Figure 1 demonstrates a sample of a weight matrix.

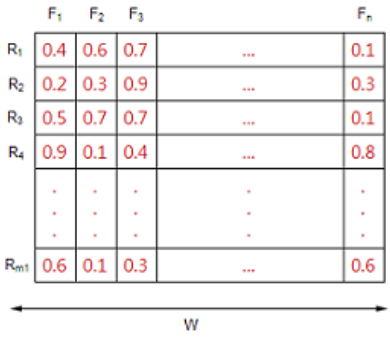

Figure 1: A sample of weight matrix

5.2 Workflow

The flowchart of weighted KNN is illustrated in Figure 2. Weighted KNN implements three main steps for classifying a test sample. These steps are run on all test samples and then, accuracy of detection will be considered as the outcome of the algorithm.

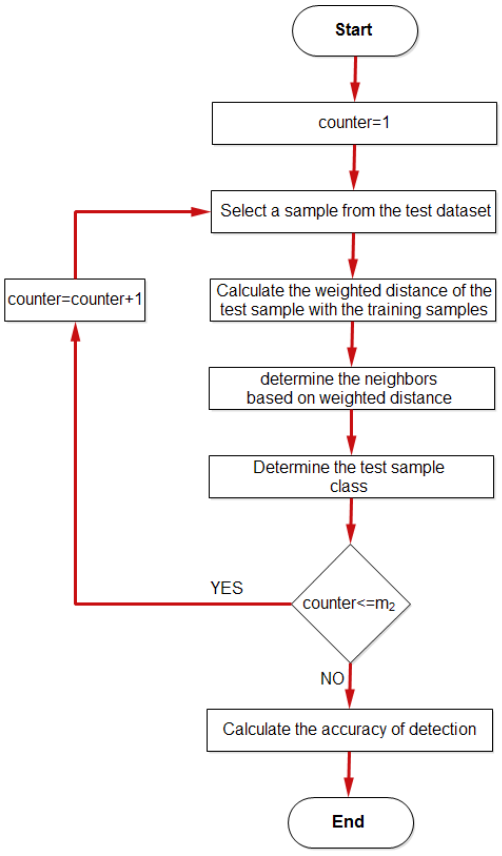

Figure 2: Flowchart of weighted KNN

Measuring the weighted distance between the test sample and the training ones is the very first step of the weighted KNN. Suppose X is a test sample while Y depicts a training sample. Both samples are composed of n features which demonstrate the n-dimensional space of the problem. To measure the distance between these samples, Euclidean distance is used where the weights affect the distance. The weighted distance between X and Y is calculated by Equation (12).

$$\|X - Y\|_2 = \sqrt{\sum_{i=1}^{n} w_{yi}(x_i - y_i)^2}. \tag{12}$$

The second step of weighted-KNN is devoted to determining the neighbors of the test sample according to the value of K. In this step, K training samples that are closer to the test sample must be selected. The proposed method considers the same K for all test samples. In the third step of the weighted KNN, the class of the test sample is determined based on the defined neighbors. To determine the class of the test sample, a voting system takes place among the neighbors. For the purpose of retaining the correct structure of voting, an odd number is considered as the value of K to prevent the system from failure.

5.3 Optimization of weight matrix

The performance of the proposed algorithm strongly depends on the weight matrix. The proposed algorithm uses WOA for optimizing the weight matrix. WOA looks for an optimized weight matrix to improve the level of accuracy of weighted KNN. Each whale in WOA depicts a feasible solution in the search space. In the proposed methodology, each whale shows a weight matrix. A linear vector with $m_1 \times n$ elements is used to encode the solutions where each element is corresponding to a weight in the matrix. Figure 3 illustrates the encoding process. Also, the proposed approach employs a raster method for converting a two-dimensional matrix to a linear vector. In the raster method, each two-dimensional matrix is scanned row by row and from left to right.

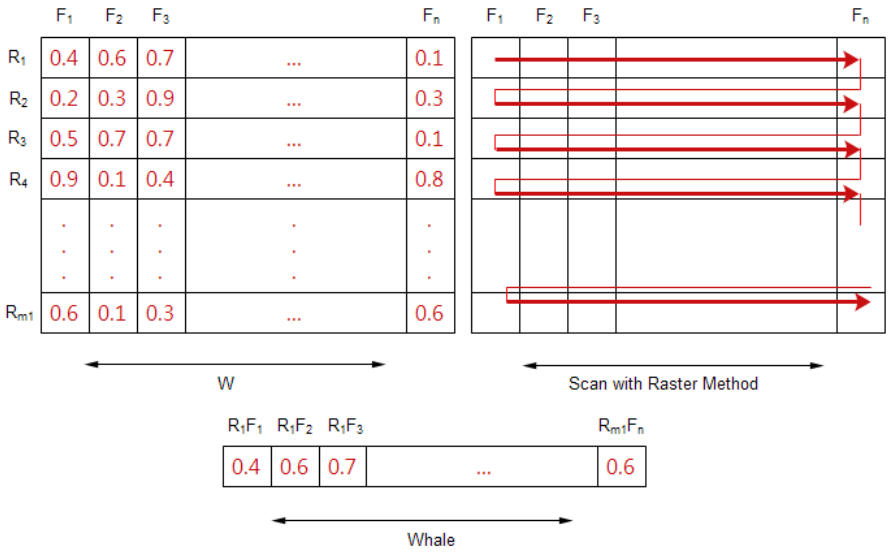

Figure 3: A sample of designed whale

WOA starts with an initial population of random whales. Each whale is a solution in the search space and is defined as a linear vector. After producing the random initial population, the quality of each individual is measured. The level of accuracy of weighted KNN is considered as a fitness function. The best whale is the one in which the corresponding weighted KNN has the highest level of accuracy. The best whale in the initial population is selected as the global solution. Afterwards, an iterative process runs till reaching the stopping criterion. In each iteration of WOA, the position of whales is modified according to the global solution and the random solution. Finally, the best whale of the final population is taken as the final answer. It is straightforward that the final answer shows a matrix with optimal weights.

6 Evaluation of the proposed method

The proposed method is implemented by MATLAB. Moreover, the proposed approach employs UCI datasets for the implementation. In this section, the proposed method is evaluated on quantitative criteria.

6.1 Evaluation metrics

Figure 4 illustrates the confusion matrix. In the confusion matrix, TP shows the number of samples with the positive class which is categorized correctly. Also, TN is the number of samples with the negative class which is detected correctly. In addition, the number of positive samples which are detected wrongly is shown by FN and FP represents the number of negative classes which are detected positive, wrongly. According to the confusion matrix, the evaluation metrics are shown in equations thirteen to sixteen.

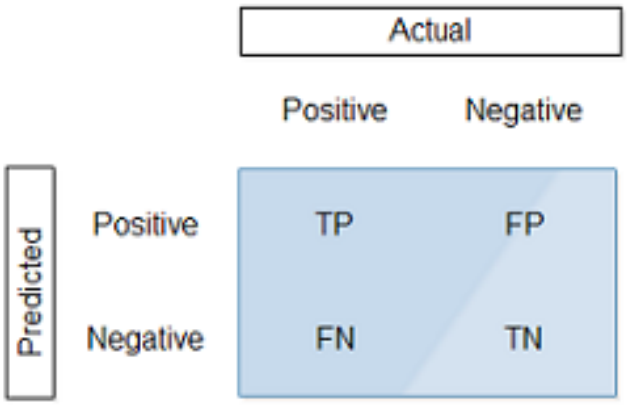

Figure 4: confusion matrix

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100, \tag{13}$$

$$\text{Error\_Rate} = \frac{FP + FN}{TP + TN + FP + FN} \times 100, \tag{14}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100, \tag{15}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100. \tag{16}$$

6.2 Parameters configuration

In this paper, a weighted KNN algorithm based on WOA is designed. Also, the weighted KNN algorithm based on GA is implemented. Table 2 shows initial values for parameters of optimization algorithms. Parameters configuration of the proposed and compared algorithms is provided in Table 3.

Table 2: Initial values for parameters of optimization algorithms

No.	Algorithm	parameter	value
1	WOA	whale length	Based on data set size
2		Size	50
3		Max	100
4		P	Randomly
5	GA	chromosome length	Based on data set size
6		population size	50
7		Imax	100
8		Pc	0.8
9		Pm	0.01

Table 3: Parameters Configuration of the Algorithms

Algorithm	Reference	Parameter
DPSO-KNN	[21]	k value: [1,384]
P-KNN	[14]	k value: 3-5-7-9
FA-FKNN	[3]	k value: 5
GA-FKNN	[35]	k value: 5
IT2-FKNN	[27]	k value: 1-3-5-7-9
Classic KNN	$\cdots$	k value: 3-5-7-9-11
WKNN_GA	$\cdots$	k value: 3-5-7-9-11
WKNN_WOA	$\cdots$	k value: 3-5-7-9-11

6.3 Datasets

In experiments, five datasets from the UCI-Dataset repository are used to evaluate the proposed methods. All the datasets are stored in the format of a Microsoft Excel file. Table 4 shows information on datasets. Each row represents a sample, while each column shows a feature. Also, the last column is responsible for showing the class of each sample. Depending on the type of dataset, preprocessing operations are applied to the dataset. Figure 5 shows the number of train and test samples in the datasets.

Table 4: Information of datasets

Dataset	the numbers of Sample	the numbers of Feature	the numbers of Class	Source
Pima	768	8	2	UCI
Haberman	306	3	2	UCI
Ionosphere	351	34	2	UCI
Mammography	11183	6	2	UCI
Spambase	4601	57	2	UCI

6.4 Evaluation algorithms

Table 5 reports the performance of the classic KNN algorithm on the standard datasets. As the table shows, the classic KNN is run by 3, 5, 7, 9 and 11 neighbors and the best result for each number of neighbors on each dataset is shown.

In Table 6, the performance of the weighted KNN based on GA on standard datasets is shown. Also, in Table 7, the performance of the weighted KNN based on WOA on standard datasets is illustrated. For the implementation of the weighted KNN has been used from the best value for K on each standard dataset.

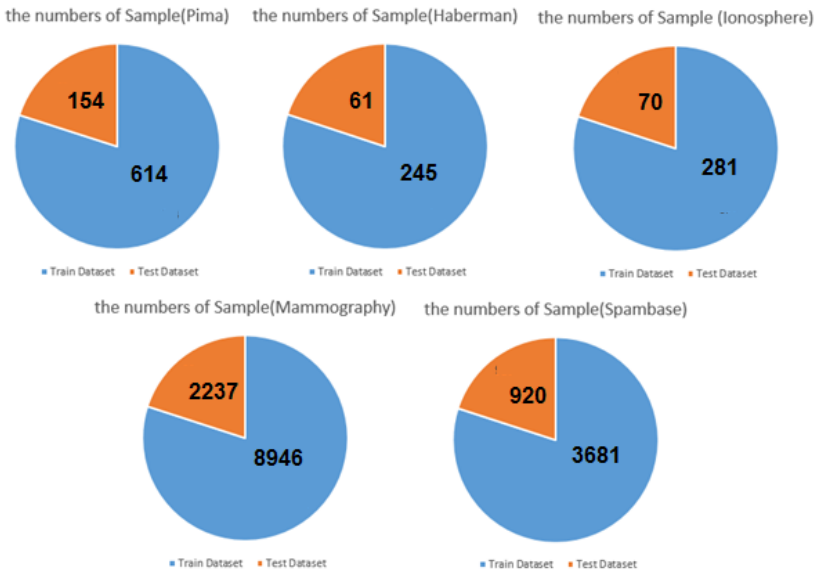

Figure 5: The number of train and test samples in the datasets

Table 5: The performance of classic KNN on standard datasets

Dataset	K=3		K=5		K=7		K=9		K=11	
	ACC	ER	ACC	ER	ACC	ER	ACC	ER	ACC	ER
Pima	65.58	34.42	64.93	35.07	**72.73**	**27.27**	68.83	31.17	69.48	30.52
Haberman	72.13	27.87	75.41	24.59	**77.05**	**22.95**	75.41	24.59	77.05	22.95
Ionosphere	82.86	17.14	82.86	17.14	82.86	17.14	84.28	15.72	**85.71**	**14.29**
Mammography	98.35	01.65	98.35	01.65	**98.43**	**01.57**	98.39	01.61	98.26	01.74
Spambase	69.13	30.87	**72.39**	**27.61**	65.65	34.35	70.11	29.89	66.85	33.15

Table 6: Performance of the weighted KNN based on GA

Dataset	K	Accuracy	Error_Rate	Sensitivity	Specificity
Pima	7	79.22	20.78	90.10	58.49
Haberman	7	81.97	18.03	80.00	82.61
Ionosphere	11	87.14	12.86	61.11	96.15
Mammography	7	98.84	01.16	99.08	88.68
Spambase	5	86.41	13.59	86.45	86.39

Table 7: Performance of the weighted KNN based on WOA

Dataset	K	Accuracy	Error_Rate	Sensitivity	Specificity
Pima	7	80.52	19.48	86.14	69.81
Haberman	7	83.61	16.39	86.67	82.61
Ionosphere	11	88.57	11.43	66.67	96.15
Mammography	7	98.93	01.07	99.13	90.57
Spambase	5	88.04	11.96	90.00	87.05

The evaluation of the accuracy of the KNN algorithms is shown in Figure 6. It is evident that the weighted KNN based on WOA has had high accuracy on all datasets.

The sensitivity of the weighted KNN based on WOA and GA is shown in Figure 7. The detection rate of positive samples in the weighted KNN based on WOA has been improved compared with the weighted KNN based on GA. The

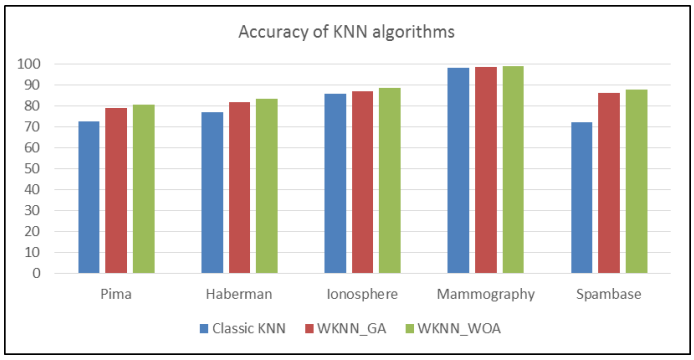

Figure 6: Evaluation of the accuracy of the KNN algorithms

specificity of the weighted KNN algorithms is shown in Figure 8. As can be seen, the detection rate of negative samples in the weighted KNN based on WOA has been improved compared with the weighted KNN based on GA.

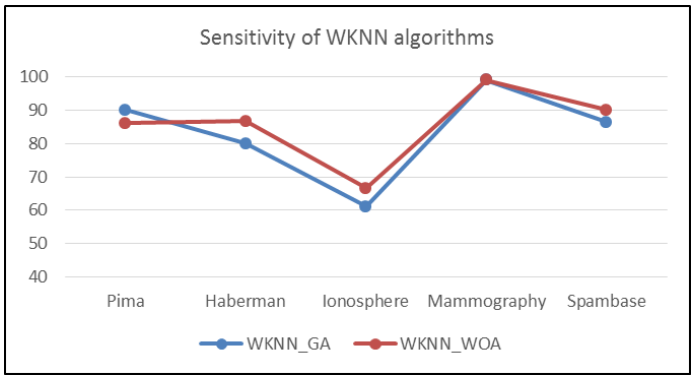

Figure 7: Sensitivity of the weighted KNN algorithms

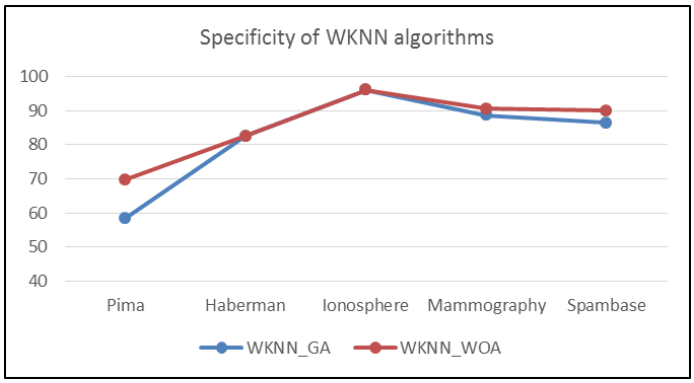

Figure 8: Specificity of the Weighted KNN algorithms

6.5 Time complexity

The time complexity of intermediate states of weighted KNN based on WOA is given below.

- Initialization of weighted KNN based on WOA requires O($Size \times m_1 \times n$) time where Size denotes the number of whales and $m_1 \times n$ is the size of whales.
- Computation of fitness function requires O($Size \times m_1 \times n \times k$) time where k denotes the number of neighbors.

- Computation of control parameters and position update steps of weighted KNN based on WOA requires O($Size \times m_1 \times n$).

Therefore, the total time complexity of weighted KNN based on WOA becomes O($Size \times m_1 \times n \times k$) per iteration/generation. The final time complexity of weighted KNN based on WOA for the maximum of iterations is O(max $\times Size \times m_1 \times n \times k$). Considering that the time complexity of the classical KNN is O($m_1 \times n \times k$), it can be said that the proposed method has a higher time complexity than the classical KNN.

6.6 Space complexity

The space complexity of intermediate states of weighted KNN based on WOA is given below.

- The space required for population storage and initialization process is O($Size \times m_1 \times n$).
- The space required for training dataset storage is O($m_1 \times n$).

Therefore, the final space complexity of weighted KNN based on WOA becomes O($Size \times m_1 \times n$). Considering that the space complexity of the classical KNN is O($m_1 \times n$), it can be said that the proposed method has a higher space complexity than the classical KNN.

6.7 Comparison

Table 8 shows the comparison results of the proposed algorithm with other algorithms. To compare the proposed algorithm with other algorithms, five standard datasets have been used. Accuracy is considered a criterion for evaluating classification algorithms. The weighted KNN based on WOA on the three datasets had the highest detection accuracy.

Table 8: Comparison of the proposed method with other methods

Dataset	DPSO-KNN [21]	P-KNN [14]	FA-FKNN [3]	GA-FKNN [35]	IT2-FKNN [27]	WKNN_WOA	Rank
Pima	76.32	76.67	74.35	74.36	74.23	**80.52**	**1 of 6**
Haberman	——	75.00	74.83	70.30	68.98	**83.61**	**1 of 5**
Ionosphere	——	92.35	——-	95.33	**96.00**	88.57	4 of 4
Mammography	——	81.82	———	79.53	80.25	**98.93**	**1 of 4**
Spambase	——	81.67	——-	81.26	91.23	**88.04**	**2 of 4**

7 Conclusions

This paper presents a weighted KNN algorithm based on WOA for solving classification problems. In this method, WOA is employed for the goal of improving the performance of weighted KNN algorithm. The performance of weighted KNN algorithm massively depends on the values of the weight matrix. The proposed method uses WOA to reach the optimum weight matrix. To evaluate the performance of the proposed method, standard datasets from UCI are used. The implementation results show that weighted KNN algorithm based on WOA performs better than weighted KNN algorithm based on GA and the classic KNN algorithm, as well. Also, the comparison results of the weighted KNN algorithm based on WOA with other methods show that it has good performance. WOA has a high ability to find optimal values for the weights matrix. WOA has the ability to discover the optimal matrix of weights by using the selection mechanism in the exploitation phase and the use of random solutions in the exploration phase. The main advantage of the proposed method is to improve evaluation criteria such as accuracy and classification error. The performance of the weighted KNN algorithm based on WOA is better than the classical KNN on all datasets. The main disadvantage of the weighted KNN algorithm based on WOA is that it has more time and space complexity than the classical KNN. Finding the optimal weight matrix using other novel metaheuristic algorithms such as Butterfly Optimization Algorithm (BOA) and Dragonfly Optimization Algorithm (DOA) are suggested for future works.

References

[1] A. A. Aburomman, M. B. I. Reaz, *A novel SVM-KNN-PSO ensemble method for intrusion detection system*, Applied Soft Computing, **38** (2016), 360-372.

[2] S. Bandaru, A. H. Ng, K. Deb, *Data mining methods for knowledge discovery in multi-objective optimization: Part A-survey*, Expert Systems with Applications, **70** (2017), 139-159.

[3] Z. Bian, C. M. Vong, P. K. Wong, S. Wang, *Fuzzy KNN method with adaptive nearest neighbors*, IEEE Transactions on Cybernetics, **52** (2022), 5380-5393.

[4] H. L. Chen, C. C. Huang, X. G. Yu, X. Xu, X. Sun, G. Wang, S. J. Wang, *An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach*, Expert Systems with Applications, **40** (2013), 263-271.

[5] A. J. P. Delima, *An enhanced K-nearest neighbor predictive model through metaheuristic optimization*, International Journal of Engineering and Technology Innovation, **10** (2020), 280-292.

[6] Z. Deng, X. Zhu, D. Cheng, M. Zong, S. Zhang, *Efficient KNN classification algorithm for big data*, Neurocomputing, **195** (2016), 143-148.

[7] N. García-Pedrajas, J. A. R. Del-Castillo, G. Cerruela-García, *A proposal for local K-values for k-nearest neighbor rule*, IEEE Transactions on Neural Networks and Learning Systems, **28** (2015), 470-475.

[8] G. V. Gayathri, S. C. Satapathy, *A survey on techniques for prediction of asthma*, Smart Intelligent Computing and Applications, **159** (2020), 751-758.

[9] Z. Geler, V. Kurbalija, M. Ivanović, M. Radovanović, *Weighted KNN and constrained elastic distances for time-series classification*, Expert Systems with Applications, **162** (2020). DOI:10.1016/j.eswa.2020.113829.

[10] J. Gou, L. Du, Y. Zhang, T. Xiong, *A new distance-weighted k-nearest neighbor classifier*, Journal of Information and Computational Science, **9** (2012), 1429-1436.

[11] F. Harrou, A. Zeroual, Y. Sun, *Traffic congestion monitoring using an improved KNN strategy*, Measurement, **156** (2020). DOI:10.1016/j.measurement.2020.107534.

[12] V. Hashemi, Z Hasani, I. Sahraei, K. Borna, *Hybrid algorithms of Whale optimization algorithm and k-nearest neighbor to predict the liver disease*, EAI Endorsed Transaction on Context-Aware Systems and Applications, **6** (2019), 1-5.

[13] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, A. A. Alhasanat, *Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach*, International Journal of Computer Science and Information Security, **12** (2014), 33-39.

[14] J. Hu, H. Peng, J. Wang, W. Yu, *KNN-P: A KNN classifier optimized by P systems*, Theoretical Computer Science, **817** (2020), 55-65.

[15] M. A. Imron, B. Prasetyo, *Improving algorithm accuracy k-nearest neighbor using z-score normalization and particle swarm optimization to predict customer churn*, Journal of Soft Computing Exploration, **1** (2020), 56-62.

[16] B. B. Jia, M. L. Zhang, *Multi-dimensional classification via KNN feature augmentation*, Pattern Recognition, **106** (2020). DOI:10.1016/j.patcog.2020.107423.

[17] N. Jothi, N. A. Rashid, W. Husain, *Data mining in healthcare-a review*, Procedia Computer Science, **72** (2015), 306-313.

[18] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, I. Chouvarda, *Machine learning and data mining methods in diabetes research*, Computational and Structural Biotechnology Journal, **15** (2017), 104-116.

[19] P. Kumar, R. S. Thakur, *Liver disorder detection using variable-neighbor weighted fuzzy K nearest neighbor approach*, Multimedia Tools and Applications, **80** (2021), 16515-16535.

[20] M. M. Kumbure, P. Luukka, M. Collan, *A new fuzzy K-nearest neighbor classifier based on the Bonferroni mean*, Pattern Recognition Letters, **140** (2020), 172-178.

[21] T. W. Liao, R. J. Kuo, *Five discrete symbiotic organisms search algorithms for simultaneous optimization of feature subset and neighborhood size of KNN classification models*, Applied Soft Computing, **64** (2018), 581-595.

[22] M. M. Mafarja, S. Mirjalili, *Hybrid Whale optimization algorithm with simulated annealing for feature selection*, Neurocomputing, **260** (2017), 302-312.

[23] N. Mastrogiannis, B. Boutsinas, I. Giannikos, *A method for improving the accuracy of data mining classification algorithms*, Computers and Operations Research, **36** (2009), 2829-2839.

[24] S. Mirjalili, A. Lewis, *The Whale optimization algorithm*, Advances in Engineering Software, **95** (2016), 51-67.

[25] T. M. Mohamed, *Pulsar selection using fuzzy KNN classifier*, Future Computing and Informatics Journal, **3** (2018), 1-6.

[26] R. Mukherji, A. Kundu, I. Mukherji, D. Gupta, P. Tiwari, A. Khanna, M. Shorfuzzaman, *LOT-cloud based healthcare model for COVID-19 detection: An enhanced k-nearest neighbor classifier based approach*, Computing, (2021), 1-21. DOI:10.1007/s00607-021-00951-9.

[27] F. H. Rhee, C. Hwang, *An interval type-2 fuzzy K-nearest neighbor*, International Conference on Fuzzy Systems, Louis, MO, USA, (2003), 25-28.

[28] S. Sharma, K. M. Osei-Bryson, G. M. Kasper, *Evaluation of an integrated knowledge discovery and data mining process model*, Expert Systems with Applications, **39** (2012), 11335-11348.

[29] Y. Song, Y. Gu, R. Zhang, G. Yu, *Bre-Partition: Optimized high-dimensional KNN search with Bregman distances*, IEEE Transactions on Knowledge and Data Engineering, **34** (2020), 1053-1065.

[30] UCI Machine Learning Repository [Online], URL: https://archive.ics.uci.edu/ml/index.php.

[31] Z. Wang, J. Na, B. Zheng, *An improved KNN classifier for epilepsy diagnosis*, IEEE Access, **8** (2020), 100022-100030.

[32] Y. L. Yang, X. Y. Bai, *A research on classification performance of fuzzy classifiers based on fuzzy set theory*, Iranian Journal of Fuzzy Systems, **16** (2019), 15-27.

[33] H. Yigit, *A weighting approach for KNN classifier*, International Conference on Electronics, Computer and Computation, Ankara, Turkey, (2013), 228-231.

[34] S. Zeraatkar, F. Afsari, *Interval-valued fuzzy and intuitionistic fuzzy-KNN for imbalanced data classification*, Expert Systems with Applications, **184** (2021). DOI:10.1016/j.eswa.2021.115510.

[35] S. Zhang, *Cost-sensitive KNN classification*, Neurocomputing, **391** (2020), 234-242.

[36] S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, *Efficient KNN classification with different numbers of nearest neighbors*, IEEE Transactions on Neural Networks and Learning Systems, **29** (2018), 1774-1785.

[37] J. Zhang, Y. Niu, W. He, *Using genetic algorithm to improve fuzzy KNN*, International Conference on Computational Intelligence and Security, Suzhou, China, (2008), 475-479.

[38] C. Zhang, J. Yao, G. Hu, T. Schott, *Applying feature-weighted gradient decent k-nearest neighbor to select promising projects for scientific funding*, Computers, Materials and Continua, **64** (2020), 1741-1753.