Cluster Computing (2024) 27:3341–3362
https://doi.org/10.1007/s10586-023-04165-w

A comparative study of optimization algorithms for feature selection on ML-based classification of agricultural data

Zeynep Garip[1] · Ekin Ekinci[1] · Murat Erhan Çimen[2]

Received: 28 June 2023 / Revised: 5 September 2023 / Accepted: 27 September 2023 / Published online: 3 October 2023

Abstract
In today's world, agricultural production and operation activities generate a lot of data. As a result, computer-aided agriculture applications have become a hot topic in the study, with various machine learning (ML) algorithms being used to classify agricultural data. This paper presents a comparative study consisting of a combination of ML algorithms with meta-heuristic algorithms for feature selection to improve the classification capability of ML algorithms by finding the features that significantly impact accuracy. We have used six different meta-heuristic algorithms for feature selection. Experiments are conducted on four different agricultural datasets with five classification models. To understand the effect of proposed models, the selected features are fed into the ML algorithms. The results prove that combining ML and meta-heuristic algorithms achieves higher classification accuracy with fewer features on agricultural datasets.

Keywords Agricultural data · Machine learning (ML) · Classification · Meta-heuristic optimization · Feature selection

1 Introduction

The science and engineering of making intelligent machines, especially intelligent computer programs, is defined as Artificial Intelligence (AI) [1]. In recent years, many studies have been made on the development of computer-aided agricultural analysis systems using AI algorithms and studies in this area continue today [2–5]. Computer-aided agriculture has increased in importance due to the need for decision support systems for agricultural data, which are necessary for increasing productivity, reducing costs, obtaining fast results, minimizing errors, reducing workload, food safety and sustainability. But before addressing any problems in this area, it is necessary to understand and analyze its basic requirements [6].

Still today, manual methods are being used to classify agricultural products by their types within themselves. However, manual classification poses problems in terms of cost, labour, and waste of time, and is prone to errors [7]. Thus, automated methods are necessary to cope with these problems. The automatic classification of agricultural products minimizes errors arising from manual classification and saves cost, labour and time.

Machine learning (ML) [8, 9] and deep learning (DL) [10] are a subfield of AI that can automatically learn linear and nonlinear relationships in datasets. Agricultural data, on the other hand, is divergent, complex and non-standard [11]. For this reason, ML is predicted to be successful in automatically classifying agricultural data according to their types.

There are many studies in the literature on the classification of agricultural data. A few of them are as follows: classification of apple fruit varieties [12, 13], corn seed [14, 15], wheat varieties [16, 17], pepper seeds [18, 19], sunflower seeds [20], rice seed varieties [21], apricot fruit [22], papaya fruit [23], citrus fruit [24], tomato [25], olive fruit [26, 27], orange varieties [28, 29], jujube fruit [30], pumpkin seed [31], soybean [32], dry bean [33], pistachio [34, 35], date fruit [36] etc. ML algorithms are extremely effective tools for classification of many agricultural products, including extreme learning machine (ELM), VGG, Inception, Logistic Regression (LR), Multilayer

✉ Ekin Ekinci
ekinekinci@subu.edu.tr

1 Faculty of Technology, Computer Engineering Department, Sakarya University of Applied Sciences, Sakarya, Turkey

2 Faculty of Technology, Electrical and Electronics Engineering Department, Sakarya University of Applied Sciences, Sakarya, Turkey

perceptron (MLP), Support Vector Machine (SVM), k-Nearest Neighbours (kNN), Decision Tree (DT), Random Forest (RF), extreme Gradient Boosting (XGB) and other hybrid networks. In the literature, studies for agricultural classification in Table 1 are summarized by giving parameters such as dataset, algorithm, results and performance criteria (accuracy). As a result, meta-heuristic algorithms based ML techniques have been widely used in recent years for agricultural classification. It has recently been demonstrated that the proposed models are useful for agricultural classification.

Although ML is an area of study in the classification of many agricultural products, it can fail due to high-dimensional agricultural datasets and the fact that these datasets contain many irrelevant and unnecessary features. To solve this problem, the need arises to lighten the burden of ML algorithms.

Feature selection is a pre-processing step that improves the performance of ML algorithms by minimizing complexity, irrelevant features, and unnecessary features in datasets. However, finding the optimal subset of features in high-dimensional feature datasets is classified as an NP-hard problem. Since a dataset with N features contains 2^{N-1} feature subsets, the search space will expand exponentially as the number of features increases. As a result, meta-heuristic algorithms were used to determine the subset of features, as the exact algorithms could not produce the desired result in a reasonable time [41]. Literature has developed and used various metaheuristic algorithms to address feature selection. ABC and PSO [37], Cuckoo Search (CS) [42], Firefly Algorithm (FA) [43], Moth Flame Optimization (MFO) [44], Multi Verse Optimizer (MVO) [45], Whale Optimization Algorithm (WOA) [46], Salp Swarm Algorithm (SSA) [47], Grasshopper Optimization Algorithm (GOA) [48], Harris Hawks Optimization (HHO) [49], Equilibrium Optimizer (EO) [50], Marine Predators Algorithm (MPA) [51], and Red Deer Algorithm (RDA) [52] are some of them.

This paper uses WOA, SSA, MFO, MVO, GOA, and MPA algorithms for feature selection. WOA algorithm has the minimum number of control parameters, effective adaptation and simple architecture [46]. The SSA algorithm has few parameters and operators and no control parameters, thus avoiding high sensitivity to unreasonable settings [53]. MFO employs several search techniques, such as attraction, repulsion, and diffusion, to explore the solution space and identify the objective function's overall global optima [54]. MVO algorithm relies on both exploitation and exploration with randomization [55]. GOA has been shown to be superior for determining the global best nonlinear functions in multi-dimensional space [56]. MPA has simplicity, easily adjustable setting

Table 1 Comparison with existing techniques used in agricultural datasets

References	Dataset	Algorithm	Results	Performance criteria (Acc) (%)
[33]	13,611 Dry bean samples	MLP, SVM, kNN, DT	MLP, SVM, kNN, and DT classification models were compared on seven different dry bean seeds, and SVM was the best.	93.19
[35]	2148 Pistachio samples	kNN	kNN model was used to classify the pistachio dataset.	94.18
[37]	33,064 Dry bean samples	SA-ELM, Artificial Bee Colony (ABC)-ELM, Particle Swarm Optimization (PSO)-ELM, HHO-ELM	The effectiveness of standard ELM and optimized ELM models in the classification of dry beans were examined. With the SSA-ELM method, the best classification results were achieved.	91.43
[38]	33,064 Dry bean samples	InceptionV3, VGG16, VGG19, InceptionV3 + SVM, VGG16 + SVM VGG19 + SVM, InceptionV3 + LR VGG16 + LR, VGG19 + LR	The InceptionV3, VGG16, and VGG19 CNN models and SVM and LR were applied for both end-to-end classification and extraction of features. The InceptionV3 was determined to be the best classification model.	84.48
[39]	13,611 Dry bean samples	XGB with ADASYN	The seeds were classified using a genetically varied dry bean dataset and XGBoost, ADASYN and XGBoost + ADASYN algorithms.	95.40
[40]	13,611 Dry bean samples	RF, SVM, and kNN	Three ML algorithms were used for classification and kNN showed better performance.	95

parameters and flexibility in the implementation [51]. Within the scope of this study, the positive characteristics of the above algorithms are why we use these algorithms in the FS problem.

This study aims to perform feature selection with WOA, SSA, MFO, MVO, GOA, and MPA algorithms with improved classification success on pumpkin seeds, pistachio, dry bean and date fruit datasets. kNN, Classification and Regression Trees (CART), MLP, Multinomial Logistic Regression (MLR), Gaussian Naïve Bayes (GNB) ML algorithms are applied to the datasets created as a result of feature selection. Based on these experiments, we will be answering whether an effective attribute selection can be made with a metaheuristic algorithm, whether a successful performance can be achieved with the selected attributes, and the effect of population size and, therefore, fitness approximation on attribute selection.

The main contributions are as follows:

- To the best of our knowledge, this is the first study that compares the effect of population size and number of iterations on feature selection in the classification of agricultural data.
- Feature selection algorithms, namely WOA, SSA, MFO, MVO, GOA, and MPA, have been utilized to eliminate the irrelevant/redundant features to enhance the performance of the kNN, CART, MLP, MLR, GNB ML algorithms.
- Experiment results demonstrate that metaheuristic algorithms are significantly superior to existing algorithms in fitness, F-score and feature reduction.

The structure of the study is as follows; In the second section, information about the meta-heuristic and ML algorithms and details of the proposed model are given. The third section includes datasets, experimental setup and evaluation metrics, experimental results and interpretation of these results. The last section concludes the paper.

2 Background

2.1 Meta-heuristic algorithms

2.1.1 Marine predators algorithm (MPA)

The marine predator's algorithm was developed by Faramarzi et al. inspired by the prey-predator social relationship between marine predators and their prey [57]. MPA is a heuristic optimization algorithm developed based on the encounter rate of marine predators and their prey [58]. The MPA initial solution starts with a random distribution in the search space. On the basis of MPA, the transition between phases in the structure of the algorithm is provided according to the speed ratio between the prey and the predator [59]. Sea predators complete the step dimension in three phases when hunting their prey [60]. The most obvious feature in the first phase of the algorithm is the great velocity. In other phases, unity and weak ratio come to the fore.

The initial solution is determined by using the MPA random and uniform distribution sample space. The number of hunters n, the number of iterations m, and the optimization parameter size d indicate the initial position of the prey. Xmax and Xmin are the maximum and minimum values in Eq. (1); the random vector is given in rand [0,1].

$$X_0 = X_{\min} + r\,\mathrm{and}\,(X_{\max} - X_{\min}) \tag{1}$$

In this section, the prey matrix holding the positions of the initial population forms the Elite matrix $Elit$.with the best fitness function. Each stage is summarized as follows. In the stage of phase 1, the prey av has to stop the movement of the predator due to its high speed. This process is carried out for only a third of the entire iteration. The behaviour of the prey is determined according to the Brownian motion. Equation 3 updates the matrices used by the prey. In Eqs. 2 and 3, P = 0.5 is determined as a vector containing random numbers with a uniform distribution between R [0,1], while R_B determines the random numbers based on the normal distribution of the Brownian motion.

$$\overrightarrow{adim_i} = \overrightarrow{R_B} \otimes (\overrightarrow{Elite_i} - (\overrightarrow{R_B} \otimes \overrightarrow{Prey_i})) \tag{2}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + (P.\vec{R} \otimes \overrightarrow{step_i}) \tag{3}$$

In phase 2, predator and prey move at the same speed, comprising two-thirds of the algorithm. Here, the prey and the predator use different methods of movement. In this phase, the predator uses the Brownian motion, and the prey uses the Levy motion. In this step, the RL, a vector containing random numbers based on the normal distribution of Levy's, is multiplied by the prey. In this step, the movements of the first half of the population are updated according to Eqs. (4 and 5).

$$\overrightarrow{adim_i} = \overrightarrow{R_L} \otimes (\overrightarrow{Elite_i} - (\overrightarrow{R_L} \otimes \overrightarrow{Prey_i})) \tag{4}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey} + (P.\vec{R} \otimes \overrightarrow{step_i}) \tag{5}$$

The other half of the population is updated according to Eqs. (6 and 7). The Elite matrix is multiplied by RB. Here CF is an adaptive parameter for controlling the step size for predator movement.

$$\overrightarrow{step_i} = \overrightarrow{R_B} \otimes ((\overrightarrow{R_B} \otimes \overrightarrow{Elite_i}) - \overrightarrow{Prey_i}) \tag{6}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Elite_i} + (P.CF \otimes \overrightarrow{step_i}) \tag{7}$$

$$CF = [1 - (Iter./Max.Iter)]^{(2.Iter./Max.Iter.)} \tag{8}$$

In Phase 3, it is assumed that the prey moves more slowly than the predator, and the algorithm uses the Levy motion of the predator for the rest of the iteration. At this point, the Elite matrix is multiplied by R_L. The Prey Matrix is being updated to Eq. (10).

$$\overrightarrow{step_i} = \overrightarrow{R_L} \otimes ((\overrightarrow{R_L} \otimes \overrightarrow{Elite_i}) - \overrightarrow{Prey_i})) \tag{9}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + (P.CF \otimes \overrightarrow{Prey_i}) \tag{10}$$

In MPA, after each iteration, the Elite matrix is replaced by the best solutions. In addition, when the maximum number of iterations is reached, or the stop criterion of the algorithm is met, the resulting solution is the final solution.

2.1.2 Whale optimization algorithm (WOA)

WOA was proposed by Jalili and Lewis for use in optimization problems inspired by the hunting behaviour of humpback whales [61]. The foraging behaviour observed only in humpback whales is bubble-net feeding. Whales form bubbles along a circular path when circling prey during hunting.

2.1.2.1 Encircling prey When hunting, humpback whales can find the location of their prey and surround the prey. Since the location of the optimal design in the search space is not known in advance, the WOA considers the best current candidate solution to be the target prey or close to the optimum solution. Once the best search agent is identified, other search agents will try to update their positions towards the best search agent. The mathematical model of the prey surrounding the behaviour of humpback whales is shown in Eqs. (11 and 12). The in Eqs. (11 and 12) represent the position of the agent, t is the iteration, is the best solution, while represent the convergence values in Eqs. (13 and 14). [0,1] shows the random number, while shows the linearly decreasing vector from 2 to zero along the iteration.

$$\vec{D} = \left|\vec{C}\,\overrightarrow{X^*}(\mathrm{t}) - \overrightarrow{X(\mathrm{t})}\right| \tag{11}$$

$$\overrightarrow{X(\mathrm{t}+1)} = \left|\overrightarrow{X^*}(\mathrm{t}) - \vec{A}.\vec{D}\right| \tag{12}$$

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{13}$$

$$\vec{C} = 2.\vec{r} \tag{14}$$

2.1.2.2 Bubble-net attacking method The bubble-net attacking method of humpback whales involves shrinking, encircling and spiral updating position towards the prey. By lowering the value of in Eq. (8), whales exhibit the behaviour of catching prey by shrinking their search environment. Since value also depends on it decreases linearly from 2 to zero. The mathematical model of the spiral shape formed by humpback whales when catching their prey is given in Eqs. (15 and 16).

$$\overrightarrow{D'} = \left|\overrightarrow{X^*}(\mathrm{t}) - \overrightarrow{X(\mathrm{t})}\right| \tag{15}$$

$$\overrightarrow{X(\mathrm{t}+1)} = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(\mathrm{t}) \tag{16}$$

In Eqs. (15 and 16) D' is the distance between the whale and the best prey, b is the logarithmic spiral constant, and l is a random number between [− 1,1]. When moving towards the prey, humpback whales are 50% likely to choose either the shrinking movement pattern or the spiral movement pattern. The parameter p in Eq. (17) is a random number within the range [0,1].

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(\mathrm{t}) - \vec{A}.\overrightarrow{D} & p < 0.5 \\ \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) & p \geq 0.5 \end{cases} \tag{17}$$

2.1.3 Moth flame algorithm (MFA)

The MFO algorithm was developed by Mirjalili (2015), inspired by the techniques by which moths navigate around light sources [62]. Using a technique called transverse orientation, moths can travel effectively and easily over long straight distances at a fixed angle relative to the moon. When moths encounter an artificial light, they try to keep a fixed angle between themselves and the light. At the same time, since the light is closer than the moon, the moths follow the path in a spiral way. The algorithm consists of the following four parts: It determines the objective function and produces a random population of moths. It generates a flame sequence, updates the positions of the moths, and finally adjusts the dimension of the flames and gets the best solution [63]. The modelling of the MFO is determined according to the moths and flames. The moths represent individuals searching in a specific area, while the flames show the best positions the moths have obtained so far. The moths update their spiral path by searching around the flames for a better solution. At first, there are the same number of moths and flames. However, to improve exploitation as the process nears its conclusion, the quantity of flames is adaptively reduced [64]. As the number of generations increases, the flame size is determined according to Eq. (18). The moth's location is updated with Eq. (19).

$$FS = round(F_{max} - G_c(F_{max} - 1))/G_{max} \tag{18}$$

$$M_i = \left|F_j - M_j\right|e^{bt}\cos(2\pi t) + F_j \tag{19}$$

FS represents the flame size, Fmax represents the maximum number of flames, Gc and Gmax represent the current and maximum production number. M represents the moth, F represents the flame, b represents the spiral shape, and t represents the random number ranging from − 1 to 1.

2.1.4 Salp swarm algorithm (SSA)

SSA was carried out by Mirjalili et al. by modelling the navigation and foraging characteristics of salp swarms found in the ocean. For the mathematical modelling of salp chains, the population is divided into leaders and followers. In the chain, the salp closest to the food source is determined as the leader, while the others are considered followers [65]. The position of the leader is updated using Eq. (20).

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right)c_3 \geq 0 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right)c_3 < 0 \end{cases} \tag{20}$$

$$m_n^1 = \begin{cases} f_n + \alpha_1((ub_n - lb_n)\alpha_2 + lb_n)\alpha_3 \geq v \\ f_n - \alpha_1((ub_n - lb_n)\alpha_2 + lb_n)\alpha_3 < v \end{cases}$$

The α_1 is computed as follows in (21):

$$\alpha_1 = 2e^{-\left(\frac{4i}{I}\right)^2} \tag{21}$$

The positions of the followers are updated according to the position of the leaders with Eq. (22). The food source is represented by fn, m_n shows the position of the leader, ub_n and lb_n are the highest and lowest values in the search field. The parameters c_2 and c_3 are selected at random within the uniform distribution interval [0,1].

$$m_n^k = \frac{1}{2}\left(m_n^k + m_n^{k-1}\right) \tag{22}$$

2.1.5 Multi verse optimizer (MVO)

The MVO algorithm was developed by Mirjalili et al. by modelling the Multi-Verse theory [66]. The mathematical model is created by including the basic components, white hole, black hole and wormhole. The white hole serves as the primary component in the creation of the cosmos in which we exist. The wormhole serves as a passageway that links various regions of the universe together, while the black hole's strong gravitational pull makes it possible for it to draw in any light beams. The MVO algorithm uses the white hole and black hole in the discovery process, while the wormhole component is applied in the exploitation process [67]. When using the MVO algorithm, the universe symbolizes the solutions, and the components of the universe are taken to be the variables in the solutions. The universe inflation rate is also known as the MVO value of the fitness function [68].

The following regulations are taken into consideration throughout the optimization process:

1. The probability of a white hole or black hole occurring and the inflation rate have direct and inverse proportional connections, respectively.
2. Higher inflation rates tend to send items through white holes, whereas lower inflation rates favour drawing matter through black holes.
3. Wormholes allow for the random movement of objects from all universes in the direction of the best universe [69, 70].

Here are mathematical representations of black holes and white holes while x is an object of the universe, N is the number of worlds, and X is the population of universes [71]. The universes are sorted using the fitness values, and one of them is chosen to be the object sender using a roulette wheel mechanism.

$$x_{ij}(t+1) = \begin{cases} x_{kj}(t), & if\ r_1 < NI\left(x_{ij}(t)\right) \\ x_{ij}(t), & otherwise \end{cases} \tag{23}$$

The universe chosen by a roulette wheel is indicated by kth, where jth stands for a parameter of the ith universe. $NI(x_{ij}(t))$ is the normalized fitness value (i.e. inflation rate) of the ith universe at iteration t, and r_1 is a random number in the range [0,1].

By updating the objects of the universe x_{ij}, which has the best rate as described by the following equation, the wormholes are also used to increase the objects of other universes and the rate of inflation.

$$x_{ij}(t+1) = \begin{cases} Z_{ij}(t), r_2 < WEP \\ x_{ij}(t), r_2 \geq WEP \end{cases} \tag{24}$$

$$Z_{ij} = \begin{cases} X_j^* + TDRx\left(\left(ub_j - lb_j\right)xr_4 + lb_j\right), r_3 < 0.5 \\ X_j^* - TDRx\left(\left(ub_j - lb_j\right)xr_4 + lb_j\right), r_3 \geq 0.5 \end{cases} \tag{25}$$

where X_j represents the object of the best universe, lb_j stands for lower bounds, ub_j for upper limits in the jth parameter (i.e. variable), and r_2, r_3, and r_4 for random integers in the range [0,1].

TDR is a coefficient that works to define the distance needed to send an object via a wormhole to the best universe. It can be defined as:

$$TDR = 1 - \frac{t^{1/p}}{T^{1/p}} \tag{26}$$

where t stands for the current iteration, T is the maximum number of iterations, and p is initially set to 6 to indicate how accurate the exploitation phase should be in the iterations.

The following equation defines WEP, which stands for the wormhole existence probability, and shows how it increases linearly across iterations to maintain the exploitation phase:

$$WEP = WEP_{min} + tx\left(\frac{WEP_{max} - WEP_{min}}{T}\right) \quad (27)$$

where the default values for WEP_{min} and WEP_{max} are 0.2 and 1, respectively.

2.1.6 Grasshopper optimization algorithm (GOA)

GOA was developed by Saremi et al. inspired by the natural behavior of locust swarms [72]. The grasshopper's movement is influenced by three variables: social interaction (Soi), wind advection (Adi), and gravitational force (Gfi). The behaviour of locust swarms was mathematically modelled using Eq. (28).

$$X_i = r_1S_i + r_2G_i + r_3A_i \quad (28)$$

X_i in Eq. (28) represents the position of i. locusts, r represents the randomly changing numbers of variables in [0,1]. Equation (29) describes the grasshoppers' social behaviour (attraction-repulsion). N denotes the number of grasshoppers. s represents the strength of social forces, l is the attractive length scale, and f is the intensity of attraction. d_{ij} is the absolute distance between ith and the jth grasshopper and $\widehat{d_{ij}}$ is a vector between two grasshoppers. Another variable of X_i is G_i. G stands for the gravitational constant, and e_g for the unity vector pointing toward the earth's centre. The final variable of X_i is called A_i. In Eq. (32), u and $\widehat{e_w}$ represent constant drift and a unity vector in the direction of the wind.

$$S_i = \sum_{j=1}^{N} s\left(d_{ij}\right)\widehat{d_{ij}}, j \neq i \quad (29)$$

$$\left(s_r = fe^{-\frac{r}{l}} - e^{-r}\right), \left(d_{ij} = \left|x_j - x_i\right|\right), \left(\widehat{d_{ij}} = \left|x_j - x_i\right| / d_{ij}\right) \quad (30)$$

N denotes the number of grasshoppers. s represents the strength of social forces, l is the attractive length scale, and f is the intensity of attraction. d_{ij} is the absolute distance between ith and the jth grasshopper and $\widehat{d_{ij}}$ is a vector between two grasshoppers. Another variable of X_i is G_i (gravitational force). G stands for the gravitational constant, and e_g for the unity vector pointing toward the earth's centre.

$$G_i = -g\widehat{e_g} \quad (31)$$

The final variable of X_i is called A_i.

$$A_i = u\widehat{e_w} \quad (32)$$

In Eq. (32), u and $\widehat{e_w}$, respectively, represent constant drift and a unity vector in the direction of the wind.

2.2 Machine learning algorithms

2.2.1 k-nearest neighbours (kNN)

A supervised ML method known as kNN is used for classification tasks and is an effective and simple-to-implement algorithm. The algorithm is based on the parameter called k, which stands for "nearest neighbours". Finding the closest data point(s) or neighbour(s) from a training dataset for a query (data point) from the test set describes how the kNN works. The closest distances from the query are used to determine the nearest data points. It uses a majority vote to determine which class appears the most frequently after determining the k closest data points. Out of the k nearest data points, the class that showed up the most is considered the class label of the query.

This classifier is excellent for classifying feature vectors with high dimensions [73]. The closest neighbour of a query can be determined using the Euclidean distance if it is assumed that data points and query correspond to points in the n-dimensional space. The Euclidean distance between a data point and a query of dimension n, such as DP = [dp_1, dp_2, …, dp_n] and Q = [q_1, q_2, …,q_n] is calculated as:

$$d(DP, Q) = \sqrt{\sum_{i=1}^{n} (dp_i - q_i)^2} \quad (33)$$

2.2.2 Classification and regression trees (CART)

CART stands for Classification and Regression Trees and was devised by Breiman et al. in 1984 [74]. Both classification and regression trees are constructed with CART, which supports continuous attributes, discrete attributes and a combination of both. The classification tree structure constructed by CART is based on the binary division of attributes. Like other classification trees, namely IS3 and C4.5, CART is also based on Hunt's algorithm and can be applied serially [75]. Because CART uses regression analysis with the use of regression trees, it differs from other Hunt's-based algorithms. When choosing the splitting attribute, the Gini index is utilized as the attribute selection measure.

2.2.3 Multi-layer perceptron (MLP)

Structures called Artificial Neural Networks (ANNs) are modelled on how the brain operates. By learning from data and generalizing the unseen situations, these networks can handle linear and nonlinear problems [76]. MLP is one of the well-liked ANNs. MLP consists of an input layer, one or more hidden layers, and an output layer. The hidden layer uses weights and biases to connect the input and output layers. Each layer composes of its own neurons. In the input layer, there are as many neurons as the number of features in the input layer. Although there is no exact way to determine the number of neurons in the hidden layer, various methods are used to determine it. There are as many neurons as the class label in the output layer. Neurons in each layer are completely interconnected with the neurons in the layer next. The input layer of MLP is fed with input data, and hidden layer(s) is used to learn complex relations between input and output. So with the help of input data and hidden layers, using summation and activation, a nonlinear function is created to predict the output data in MLP [77].

2.2.4 Multinomial logistic regression (MLR)

MLR is an extension of linear logistic regression used to classify data that has more than two unordered classes. By measuring each independent variable's distinct contribution, MLR is a quick and effective technique to examine the impact of a group of continuous or categorical independent variables on the output. So, independent variables can be used to predict the value of a dependent variable. While doing this, the MLR predicts a different logistic regression model for each dependent variable based on the reference category [78]. In MLR, one of the classes is accepted as the reference category. If we are to exemplify the dry bean dataset, there are 7 classes (y) in this dataset, namely 0-Seker, 1-Red Bean, 2-Bombay, 3-Cali, 4-Dermosan, 5-Horoz and 6-Sira. We assume class 0 (Seker) as the reference category. For ith data point, the probability of falling into a category is represented with $\pi_i^{(s)} = \Pr(y_i = s), s = 1,2,\ldots,6$ with the reference category $\pi_i^{(0)}$. A multinomial logistic regression model with a logit link can thus be depicted as follows for a straightforward model with a single independent variable, x_i [79]:

$$\log\left(\frac{\pi_i^{(s)}}{\pi_i^{(0)}}\right) = \beta_0^{(s)} + \beta_1^{(s)} x_i \tag{34}$$

Each of the s categories in this model contains the same independent variable, and each contrast's intercept $\beta_0^{(s)}$ and slope $\beta_1^{(s)}$ are typically calculated separately.

2.2.5 Gaussian naïve bayes (GNB)

With the assumption of conditional independence between every pair of features given the value of the class variable, GNB is one of the supervised learning algorithms that apply Bayes' theorem in the Naive form. A common assumption when working with continuous data is that the continuous values corresponding to each class are distributed according to the Gaussian distribution. GNB is Naive Bayes' extended version of a Gaussian probability. In order to implement the classification, GNB assumes that the likelihood of the features is Gaussian [80].

Given a data point X, described by its feature vector (x_{11}, x_{12}, x_{13}, …, x_{1n}), and a target class y, P(X|y) is calculated as:

$$P(X|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{\left(-\frac{(X-\mu_y)^2}{2\pi\sigma_y^2}\right)} \tag{35}$$

where μ_y is the mean, and σ_y^2 is the variance of the data points that their class label is y.

3 Simulation studies

Feature selection outcomes in the selection of the best subset of features with minimum redundancy and maximum recognition power [81]. The FS method removes unwanted features, which results in improved performance, reduced complexity time, and smooth execution of the system with selected subsets from the dataset [82]. The FS method's main objective is to simplify the dataset to remove noisy, irrelevant features that affect the system's performance, thereby reducing the features' dimensionality reduction [83]. Among the many attributes obtained from the pumpkin seeds, pistachio, dry bean and date fruit datasets, it is aimed to achieve a high accuracy rate with fewer attributes. To accomplish this goal, WOA, SSA, MFO, MVO, GOA and DA optimization algorithms are used in attribute selection. The best features obtained are used to carry out a classification process by increasing the accuracy of kNN, CART, MLP, MLR, and MNB ML algorithms. The proposed model is given with Fig. 1.

3.1 Datasets

The datasets used in this study are the pumpkin seeds dataset [33], pistachio dataset [41], dry bean dataset [35] and date fruit dataset [42]. The pumpkin seeds dataset consists of 2500 data points with 12 morphological features and two classes, namely Çerçevelik and Ürgüp Sivrisi. The pistachio dataset consists of 2148 data points with 12 morphological features, four shape features, 12 colour features and two classes, namely Kırmızı Pistachio and Siirt Pistachio. The dry bean dataset consists of 13,611 data points with 12 morphological features, four shape features and seven classes, namely Barbunya, Bombay, Çalı, Dermason, Horoz, Şeker and Sıra. The date fruit dataset consists of 898 data points with 12 morphological features, four shape features, 18 colour features and seven classes,

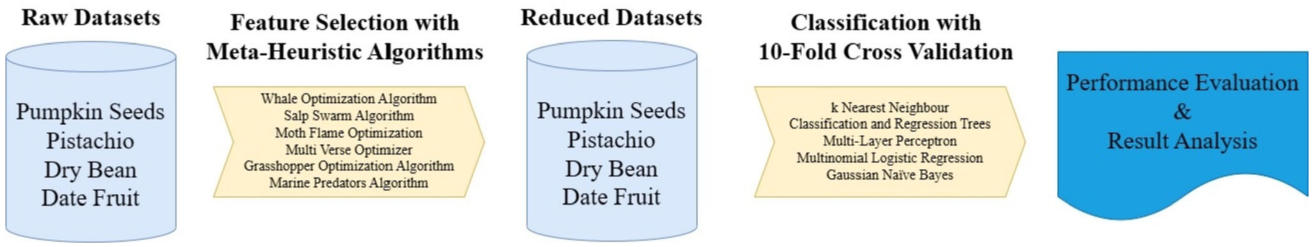

Fig. 1 Proposed model

namely Berhi, Deglet, Dokol, Iraqi, Rotana, Safavi and Sokay. The detail of the feature set with the descriptions is listed in Table 2.

3.2 Experimental setup and evaluation metrics

In order to evaluate the effects of the parameters used in the WOA, SSA, MFO, MVO, GOA, and MPA algorithms, the number of population is chosen as 10 and 20, and the number of iterations is chosen with different values as 10, 50 and 100. To implement meta-heuristic algorithms, MATLAB is used. ML experiments are performed on Google Colab on a system configuration; GPU Tesla k80 with 12 GB of GDDR5 VRAM, and Intel Xeon Processor with two 2.20-GHz cores and 13 GB RAM. We use the Sklearn library for ML algorithms. To perform experiments, model parameters are determined separately for each algorithm. Only the k value needs to be decided when using kNN. Specifying this parameter at random is not a good strategy. The number of nearest neighbours is fixed to k = 5 for all datasets based on the Elbow approach. For CART, in order to select the best splitting attribute for each node, the Gini Index is used as a splitting criterion. The maximum depth of the tree for CART is set to “none”; therefore, the nodes are expanded until all leaves are pruned. Splitting the internal node algorithm requires at least two samples. For each dataset, the input layer of MLP contains as many input neurons as the number of features. In the realized architecture, there are 5 hidden layers and 2 neurons in each layer and one output layer. The strength of the L2 regularization term alpha value is set to 1e-5. For MLR, liblinear is used for coordinate descent-based optimization. To prevent overfitting L2 penalty is preferred, and 1.0 is set to the C parameter. For GNB, the smoothing value is used as 1e-9.

F-score is used in this work to evaluate parameter settings of metaheuristic algorithms on the representative algorithms and the algorithms’ performance on the datasets with 10-fold cross-validation. F-score is the harmonic mean of the Precision and Recall.

For the confusion matrix of Kırmızı Pistachio from the Pistachio dataset, true positive (TP) represents correctly classified Kırmızı Pistachio samples. False positive (FP) is the number of Siirt Pistachio samples which are classified as Kırmızı Pistachio. False negative (FN) shows the number of incorrectly classified Kırmızı Pistachio samples, and True negative (TN) is the number of correctly classified Siirt Pistachio samples. The formulas of Precision and Recall and, depending on these two metrics, F-score is given in terms of TP, FP, FN, and TN are given as follows:

$$Precision = \frac{TP}{TP + FP} \tag{36}$$

$$Recall = \frac{TP}{TP + FN} \tag{37}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{38}$$

3.3 Experimental results

The features selected according to the minimum (Min), mean (Mean) and standard deviation (Std) and minimum fitness value obtained for each of the meta-heuristic algorithms applied to the Pumpkin Seeds dataset are given in Table 3.

Analyzing Table 3 according to the minimum fitness value for population size 10 gives the lowest fitness value for WOA, SSA, MFO, GOA and DA at iteration 50, and MVO gave the lowest fitness value at iteration 100. According to best fitness value, attributes were selected from 25% of attributes (3 out of 12) with WOA and MVO, 41.67% (5 out of 12) with SSA, 50% (12 out of 12) with MFO and GOA 6) and 58.33% (7 out of 12) with DA. When the algorithms are examined in terms of standard deviation, the lowest standard deviation in population size 10 was obtained in WOA, GOA at 10, DA at 100, SSA, and MFO at 50 and in MVO at 10 iterations. It is seen that these algorithms are more stable in terms of the specified population size and the number of iterations compared to other cases. At population size 20, the minimum fitness value was achieved with WOA and SSA at 50 iterations, while with MFO, MVO, GOA and DA, it was achieved at 10 iterations. According to the lowest fitness value, attributes were selected from 41.67% of attributes (5 out of 12) with

Table 2 Detailed description of features for datasets

No.	Feature	Description
Morphological Features		
1.	Area (A)	The number of pixels in the boundaries of the pumpkin seed/bean. $A = \sum_{r,c \in R} 1$ where r and c are the sizes of region R
2.	Perimeter (P)	Pumpkin seed's/bean's circumference in pixels
3.	Major axis length (L)	The longest line between the two ends of a pumpkin seed/bean
4.	Minor axis length (l)	The shortest line between the two ends of a pumpkin seed/bean
5.	Eccentricity (Ec)	The eccentricity of the ellipse with the same moments as the region
6.	Equivalent diameter (Ed)	Diameter of a circle with the same area as the pumpkin seed/bean area: $Ed = \sqrt{\frac{4*A}{\pi}}$
7.	Solidity (S)	Also referred to as convexity. The proportion of pumpkin seed/bean pixels to those in the convex shell: $S = \frac{A}{C}$
8.	Convex area (C)	It gave the pixel count of the smallest convex shell in the region formed by the pumpkin seed/bean
9.	Extent (Ex)	It gave back the proportion of the pumpkin seed/bean area to the pixels in the bounding box. $Ex = \frac{A}{A_B}$ where A_B Area of bounding rectangle
10.	Aspect ratio (K)	It gave the pumpkin seeds'/beans' aspect ratio: $K = \frac{L}{l}$
11.	Roundness (R)	Without taking the deformation of the edges into account, it measured the ovality of pumpkin seeds/beans: $R = \frac{4\pi A}{P^2}$
12.	Compactness (Co)	It calculated the ratio between the pumpkin seed's/bean's surface area and the surface area of a circle with the same circumference: $Co = \frac{Ed}{L}$
Shape features		
13.	Shape factor 1 (SF1)	$SF1 = \frac{L}{A}$
14.	Shape factor 2 (SF2)	$SF2 = \frac{l}{A}$
15.	Shape factor 3 (SF3)	$SF3 = \frac{A}{\frac{L}{2}*\frac{L}{2}*\pi}$
16.	Shape factor 4 (SF4)	$SF4 = \frac{A}{\frac{L}{2}*\frac{l}{2}*\pi}$
Color features [84]		
17.	MeanRR	Mean density value of red pixels values
18.	MeanRG	Mean density value of green pixels values
19.	MeanRB	Mean density value of blue pixels values
20.	StdDevRR	Standard deviation of red pixel values
21.	StdDevRG	Standard deviation of green pixel values
22.	StdDevRB	Standard deviation of blue pixel values
23.	SkewRR	Skewness value of red pixel values
24.	SkewRG	Skewness value of green pixel values
25.	SkewRB	Skewness value of blue pixel values
26.	KurtosisRR	Kurtosis value of red pixel values
27.	KurtosisRG	Kurtosis value of green pixel values
28.	KurtosisRB	Kurtosis value of blue pixel values
29.	EntropyRR	Entropy value of red pixel values
30.	EntropyRG	Entropy value of green pixel values
31.	EntropyRB	Entropy value of blue pixel values
32.	ALLdaub4RR	Wavelet decomposition level of the matrix from red pixel value using two-dimensional wavelet (wavelet order db4)
33.	ALLdaub4RG	Wavelet decomposition level of the matrix from green pixel value using two-dimensional wavelet (wavelet order db4)
34.	ALLdaub4RB	Wavelet decomposition level of the matrix from blue pixel value using two-dimensional wavelet (wavelet order db4)

Table 3 Results of meta-heuristic algorithms on pumpkin seeds dataset

Algorithm	Population size	Iteration	min	mean	std	Selected features
WOA	10	10	0.1080	0.1329	0.0108	2, 3, 6, 7, 8, 10, 11, 12
		50	0.1040	0.1266	0.0101	2, 3, 9
		100	0.1107	0.1248	0.0076	2, 3, 6, 8, 9, 11
	20	10	0.1053	0.1304	0.0106	2, 3, 12
		50	0.1013	0.1254	0.0115	2, 3, 7, 8, 12
		100	0.1027	0.1223	0.0124	2, 3, 4, 7, 10, 11, 12
SSA	10	10	0.1133	0.1303	0.0101	2, 3, 4, 6, 7, 8, 9, 10, 11
		50	0.1067	0.1247	0.0085	2, 3, 4, 6, 9
		100	0.1080	0.1272	0.0086	2, 3, 4, 6, 7, 8, 9, 10, 12
	20	10	0.1040	0.1286	0.0108	2, 3, 4, 6, 11, 12
		50	0.1000	0.1272	0.0093	4, 6, 8, 9, 10, 11
		100	0.1040	0.1217	0.0078	2, 3, 7, 9, 11
MFO	10	10	0.1080	0.1264	0.0101	2, 3, 4, 7, 9, 11, 12
		50	0.1053	0.1221	0.0069	2, 3, 4, 8, 9, 11
		100	0.1080	0.1243	0.0080	2, 3, 8, 10
	20	10	0.0960	0.1234	0.0089	2, 3, 9
		50	0.1000	0.1244	0.0086	7, 8, 10
		100	0.1040	0.1223	0.0087	2, 3, 10, 11, 12
MVO	10	10	0.1133	0.1293	0.0104	7, 10, 11
		50	0.1067	0.1274	0.0105	3, 4, 7, 8, 11
		100	0.0960	0.1292	0.0308	2, 3, 9
	20	10	0.1053	0.1236	0.0070	2, 3, 8, 9, 11
		50	0.1147	0.1269	0.0071	2, 3, 8, 10, 12
		100	0.1080	0.1254	0.0095	2, 3, 8, 10
GOA	10	10	0.1120	0.1326	0.0120	2, 3, 4, 6, 7, 8, 9, 10, 11, 12
		50	0.1067	0.1267	0.0098	2, 3, 4, 7, 9, 11
		100	0.1147	0.1267	0.0072	2, 3, 6, 12
	20	10	0.1067	0.1252	0.0104	2, 3, 9, 10, 11
		50	0.1160	0.1266	0.0073	2, 3, 4, 12
		100	0.1080	0.1275	0.0086	2, 3, 4, 6, 7, 8, 9, 10, 12
DA	10	10	0.1067	0.1336	0.0289	2, 3, 8, 10, 11
		50	0.0947	0.1310	0.0339	2, 3, 4, 7, 8, 9, 12
		100	0.0987	0.1270	0.0118	2, 3, 7, 9, 10, 12
	20	10	0.1093	0.1270	0.0109	2, 3, 9, 10, 11
		50	0.1133	0.1273	0.0097	2, 3, 4, 6, 11, 12
		100	0.1120	0.1250	0.0080	2, 3, 8, 11, 12

WOA, MVO, GOA and DA, 50% (6 out of 12) with SSA, and 25% (2 out of 12) with MFO. Considering the smallest value of the standard deviation for 20 iterations, it can be said that WOA and MVO are more stable than others in 10 iterations, SSA and DA in 100, and MFO and GOA in 50 iterations. The F-score value obtained by assigning the selected features to the ML algorithms and the F-score value obtained from the original feature set are compared in Table 4.

For the Pumpkin seed dataset, the F-score values from the original feature set are lower for kNN, and MLP, while the performances from CART, MLR and GNB are comparatively higher. As a result of feature selection, an increase in performance by 17–21% was observed in kNN. In comparison, the CART algorithm achieved a maximum increase of 1%, while for some iterations and populations, there was a 1% decrease. MLP increased the classification success of 48% by 34–35% through feature selection. The datasets created by MLR feature selection did not show an increase but experienced a 1% decrease. GNB, on the other hand, provided a performance increase between 2 and 9% but experienced a 3% decrease in datasets obtained with WOA (50 iterations, population 10) and MVO (100 iterations, population 10).

Table 4 F-score values for the pumpkin seed dataset

Algorithm	Iteration	kNN		CART		MLP		MLR		GNB	
	Original	0.66		0.83		0.48		0.88		0.78	
		Population size									
		10	20	10	20	10	20	10	20	10	20
WOA	10	0.87	0.87	0.84	0.82	0.83	0.87	0.87	0.87	0.86	0.84
	50	0.87	0.83	0.82	0.83	0.87	0.87	0.87	0.87	0.75	0.86
	100	0.87	0.87	0.84	0.83	0.82	0.86	0.87	0.87	0.84	0.87
SSA	10	0.87	0.87	0.84	0.83	0.83	0.82	0.87	0.87	0.86	0.87
	50	0.87	0.87	0.84	0.83	0.84	0.86	0.87	0.87	0.8	0.86
	100	0.87	0.87	0.82	0.83	0.84	0.84	0.87	0.87	0.86	0.87
MFO	10	0.87	0.87	0.83	0.82	0.86	0.87	0.87	0.87	0.87	0.75
	50	0.87	0.87	0.83	0.83	0.82	0.85	0.87	0.85	0.86	0.86
	100	0.87	0.87	0.83	0.83	0.85	0.83	0.87	0.87	0.82	0.86
MVO	10	0.87	0.87	0.82	0.84	0.86	0.85	0.87	0.87	0.86	0.85
	50	0.86	0.87	0.84	0.83	0.87	0.83	0.87	0.87	0.87	0.82
	100	0.87	0.87	0.82	0.83	0.87	0.85	0.87	0.87	0.75	0.82
GOA	10	0.87	0.87	0.84	0.84	0.84	0.85	0.87	0.87	0.87	0.85
	50	0.87	0.87	0.82	0.83	0.82	0.84	0.87	0.87	0.87	0.86
	100	0.87	0.87	0.82	0.82	0.83	0.84	0.87	0.87	0.84	0.86
DA	10	0.87	0.87	0.83	0.84	0.85	0.85	0.87	0.87	0.87	0.85
	50	0.87	0.87	0.83	0.83	0.87	0.82	0.87	0.87	0.87	0.87
	100	0.86	0.87	0.83	0.84	0.86	0.83	0.87	0.87	0.86	0.86

F-score values of pumpkin seed dataset for MLR with population size 10, are given in Fig. 2, for each feature selection algorithms.

The features selected for each of the meta-heuristic algorithms applied to the Pistachio dataset are given in Table 5.

Analyzing Table 5 according to the minimum fitness value for population size 10 gives the lowest fitness value for WOA MVO, GOA and DA at iteration 50. In contrast, MFO gave the lowest fitness value at iteration 100. According to best fitness value, attributes were selected from 57.16% of attributes (16 out of 28) with WOA, 50% (14 out 28) with SSA and GOA, 64.29% (18 out of 28) with MFO, 46.43% (13 out of 28) with MVO, and 25% (7 out of 28) with DA. Analyzing algorithms with regard to standard deviation, it was determined that WOA, SSA and MFO for

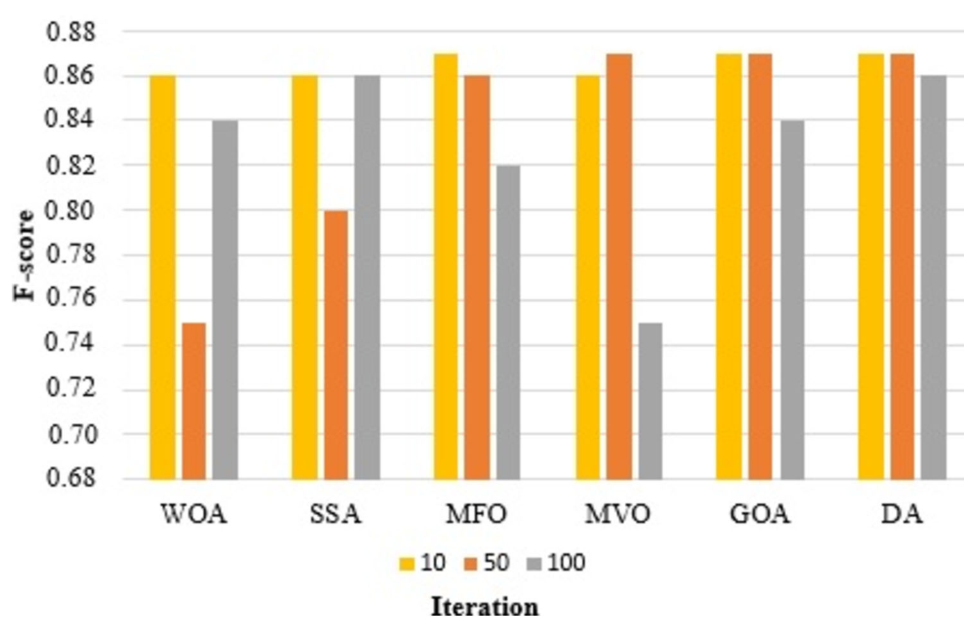

Fig. 2 F-score values for pumpkin seed dataset

Table 5 Results of meta-heuristic algorithms on the pistachio dataset

Algorithm	Population size	Iteration	Min	Mean	Std	Selected features
WOA	10	10	0.1134	0.1440	0.0249	4, 6, 9, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
		50	0.1025	0.1344	0.0150	4, 6, 9, 10, 14, 15, 16, 17, 19, 20, 21, 22, 23, 25, 27, 28
		100	0.1040	0.1304	0.0183	3, 4, 5, 6, 7, 9, 10, 12, 14, 16, 17, 18, 19, 20, 21, 22, 24, 26, 27, 28
	20	10	0.1040	0.1308	0.0155	3, 4, 5, 6, 9, 10, 12, 13, 14, 15, 17, 18, 20, 21, 22, 23, 24, 25, 27, 28
		50	0.1009	0.1220	0.0111	3, 4, 5, 6, 9, 10, 11, 14, 15, 17, 20, 21, 22, 23, 24, 25, 26
		100	0.0901	0.1214	0.0114	3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
SSA	10	10	0.1118	0.1364	0.0223	3, 4, 5, 6, 9, 12, 17, 18, 20, 22, 24, 25, 26, 28
		50	0.1009	0.1185	0.0078	4, 6, 7, 9, 12, 13, 16, 17, 19, 20, 22, 23, 24, 27, 28
		100	0.0901	0.1172	0.0100	3, 4, 6, 12, 15, 16, 17, 20, 21, 22, 23, 24, 26, 28
	20	10	0.1009	0.1275	0.0122	4, 6, 9, 10, 13, 14, 17, 19, 20, 22, 24, 25
		50	0.0994	0.1160	0.0094	4, 5, 6, 7, 11, 12, 13, 14, 16, 17, 20, 23, 25, 26
		100	0.0947	0.1140	0.0108	4, 5, 6, 11, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24
MFO	10	10	0.1087	0.1264	0.0108	3, 4, 6, 7, 11, 12, 17, 18, 19, 21, 22, 24, 25, 26, 28
		50	0.0947	0.1170	0.0107	3, 4, 5, 6, 7, 9, 13, 14, 17, 18, 20, 21, 22, 24, 25, 26
		100	0.0916	0.1169	0.0110	4, 5, 6, 7, 11, 12, 13, 14, 15, 17, 18, 20, 21, 22, 23, 24, 25, 26
	20	10	0.0978	0.1219	0.0132	3, 4, 6, 7, 9, 10, 12, 13, 14, 15, 17, 20, 21, 22, 23, 28
		50	0.0978	0.1134	0.0082	3, 6, 12, 17, 18, 21, 22, 23, 26, 27, 28
		100	0.0901	0.1106	0.0104	3, 4, 5, 9, 12, 15, 16, 17, 18, 21, 22, 25, 27
MVO	10	10	0.1056	0.1368	0.0305	4, 5, 7, 10, 13, 14, 17, 18, 19, 20, 22, 23, 24, 26, 28
		50	0.0932	0.1258	0.0237	3, 4, 6, 7, 12, 13, 16, 18, 20, 21, 25, 26, 28
		100	0.0963	0.1248	0.0124	3, 4, 6, 9, 12, 13, 17, 18, 20, 21, 23, 25, 26, 27
	20	10	0.1040	0.1214	0.0092	4, 6, 7, 9, 10, 14, 15, 17, 19, 20, 21, 23, 25
		50	0.0994	0.1186	0.0106	3, 6, 7, 11, 12, 13, 17, 18, 19, 21, 22, 25, 26, 27, 28
		100	0.0994	0.1179	0.0089	3, 4, 6, 10, 11, 13, 14, 15, 16, 18, 20, 21, 22, 27, 28
GOA	10	10	0.1087	0.1408	0.0292	3, 6, 9, 10, 11, 12, 13, 15, 18, 19, 22, 23, 24, 28
		50	0.1009	0.1331	0.0181	3, 4, 5, 6, 9, 11, 12, 13, 14, 16, 17, 18, 20, 22, 23, 24, 25, 27, 28
		100	0.1009	0.1266	0.0114	4, 6, 10, 11, 12, 14, 17, 18, 19, 20, 22, 23, 24, 26
	20	10	0.0901	0.1281	0.0146	3, 4, 5, 6, 7, 9, 10, 12, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 28
		50	0.1056	0.1246	0.0106	3, 4, 5, 6, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 22
		100	0.1118	0.1256	0.0083	3, 6, 7, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 23, 25, 28
DA	10	10	0.1080	0.1298	0.0280	2, 3, 6, 7, 9, 10, 12
		50	0.1040	0.1309	0.0318	2, 3, 4, 6, 7, 11, 12
		100	0.1080	0.1289	0.0113	2, 3, 4, 6, 8, 11
	20	10	0.1053	0.1266	0.0096	2, 3, 8, 11, 12
		50	0.1093	0.1264	0.0080	2, 3, 6, 9, 11, 12
		100	0.1120	0.1248	0.0071	2, 3, 9, 10, 11

population size 10 were more stable at 50 iterations, while MVO, GOA and DA were more stable at 100 iterations. At population size 20, the minimum fitness value was achieved with WOA, SSA, and MFO at 100 iterations, while GOA and DA achieved that at 10 iterations. According to best fitness value, attributes were selected from 57.16% of attributes (25 out of 28) with WOA, 50% (15 out 28) with SSA and MVO, 64.29% (13 out of 28) with MFO, 46.43% (19 out of 28) with GOA, and 25% (5 out of 28) with DA. Considering the smallest value of the standard deviation for 20 iterations, it can be said that WOA and MVO are most stable than others 10 iterations, SSA and DA at 100, and MFO and GOA at 50 iterations. The F-score value obtained by assigning the selected features to the ML algorithms and the F-score value obtained from the original feature set are given in Table 6.

Table 6 F-score values for the pistachio dataset

Algorithm	Iteration	kNN		CART		MLP		MLR		GNB	
	Original	0.76		0.85		0.56		0.87		0.82	
		10	20	10	20	10	20	10	20	10	20
WOA	10	0.87	0.88	0.83	0.84	0.84	0.85	0.87	0.88	0.85	0.88
	50	0.86	0.87	0.84	0.84	0.85	0.84	0.89	0.89	0.88	0.89
	100	0.87	0.87	0.84	0.84	0.85	0.86	0.88	0.88	0.88	0.88
SSA	10	0.87	0.87	0.84	0.83	0.83	0.83	0.88	0.85	0.88	0.88
	50	0.87	0.86	0.84	0.83	0.84	0.84	0.89	0.89	0.86	0.88
	100	0.87	0.87	0.85	0.83	0.85	0.83	0.9	0.85	0.89	0.88
MFO	10	0.87	0.87	0.84	0.84	0.84	0.85	0.86	0.89	0.85	0.88
	50	0.88	0.86	0.83	0.82	0.83	0.84	0.87	0.87	0.86	0.83
	100	0.87	0.86	0.83	0.84	0.83	0.82	0.87	0.84	0.88	0.86
MVO	10	0.88	0.86	0.84	0.83	0.86	0.83	0.89	0.87	0.86	0.88
	50	0.87	0.86	0.83	0.82	0.83	0.84	0.89	0.86	0.87	0.82
	100	0.87	0.88	0.84	0.84	0.83	0.84	0.86	0.88	0.87	0.88
GOA	10	0.86	0.87	0.82	0.84	0.86	0.86	0.89	0.88	0.87	0.88
	50	0.88	0.86	0.85	0.84	0.85	0.83	0.89	0.86	0.87	0.87
	100	0.86	0.86	0.83	0.84	0.82	0.84	0.88	0.88	0.88	0.86
DA	10	0.82	0.75	0.81	0.8	0.78	0.6	0.86	0.84	0.86	0.75
	50	0.84	0.82	0.81	0.8	0.76	0.79	0.87	0.87	0.86	0.85
	100	0.75	0.58	0.8	0.78	0.61	0.51	0.86	0.85	0.81	0.81

For the Pistachio dataset, the F-score values from the original feature set were low for MLP, while the remaining algorithms were found to be relatively successful. Feature selection has shown that kNN's performance increases from 6 to 12%, but in some cases, the DA algorithm has also experienced a decrease of up to 18%. The CART algorithm did not provide an increase but a decrease of 1–7%. MLP showed a performance increase between 26 and 30% in feature sets from other algorithms except DA. Classification with features from the DA algorithm resulted in a performance increase between 4 and 23% and a decrease of 5% (Population size 20, iteration 100) at the same time. MLR achieved the highest classification performance of 90% for the Pistachio dataset with features from the SSA algorithm (population size 10, iteration 100). However, some algorithms have experienced decreases of up to 3%. GNB, on the other hand, provided a performance increase between 2 and 7% but experienced a 7% decrease in datasets obtained with DA (10 iterations, population 20).

F-score values of pistachio dataset for MLR with population size 10, are given in Fig. 3, for each feature selection algorithms.

The features selected for each of the meta-heuristic algorithms applied to the Dry bean dataset are given in Table 7.

Analysing Table 7 according to the minimum fitness value for population size 10 gives the lowest fitness value for WOA, SSA, MFO and MVO at iteration 50, GOA at iteration 100, while DA gave the lowest fitness value at iteration 10. According to best fitness value, attributes were selected from 81.25% of attributes (13 out of 16) with WOA, 56.25% (9 out 16) with SSA, 68.75% (11 out of 16) with MFO, 62.5% (10 out of 16) with MVO, and 37.5% (6 out of 16) with GOA and DA. Analyzing algorithms with regards to standard deviation, it was determined that WOA, MFO, MVO and DA for population size 10 were more stable at 50 iterations, while SSA at iteration 100, and GOA at iteration 10 were more stable compared to others. At population size 20, the minimum fitness value was achieved with WOA and MVO at 50 iterations, while it was achieved with SSA and GOA at 100 iterations and with MFO and DA at 10 iterations. According to low fitness value, attributes were selected from 62.5% of attributes (10 out of 16) with WOA, 43.75% (7 out of 16) with SSA, 50% (8 out of 16) with MFO and MVO, 75% (12 out of 16) with GOA, and 31.25% (5 out of 16) with DA. Considering the smallest value of the standard deviation for 20 iterations, it can be said that WOA, MVO and DA are most stable than others at 50 iterations and SSA and GOA at 100 iterations. The F-score value obtained by assigning the selected features to the ML algorithms and the F-score value obtained from the original feature set are given in Table 8.

For the Dry bean dataset, the highest F-score value taken from the original feature set was 0.72, achieved through GNB. Classification on the original dataset through kNN with an F-score value of 17% increased success to 77, 78%

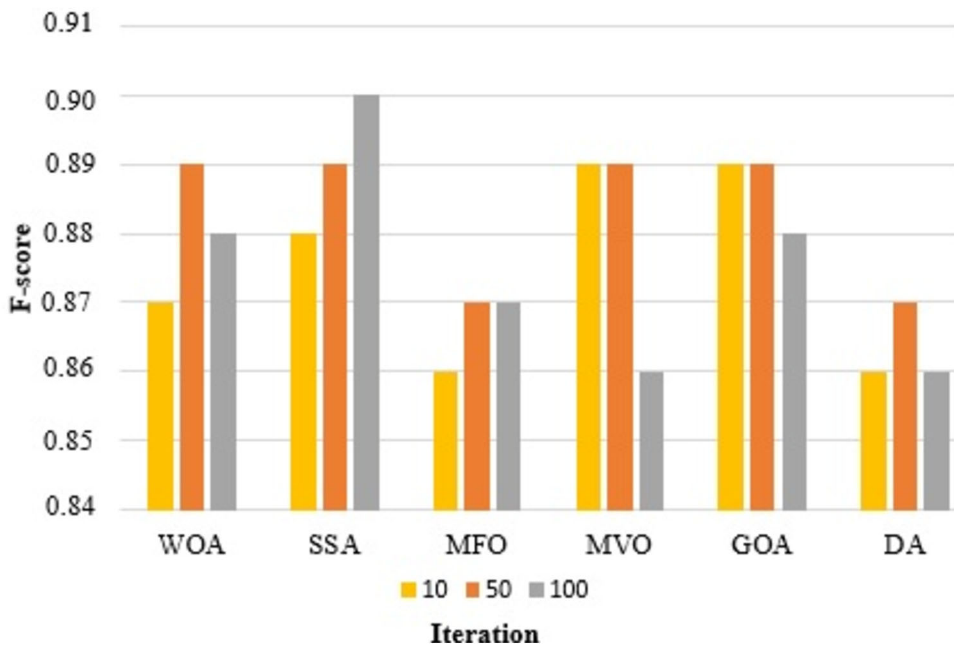

Fig. 3 F-score values for pistachio dataset

after feature selection. However, the DA algorithm showed no change in performance for population size 20 and 100 iterations, but in some cases, it was able to increase the success to 22% and in others to 25%. The highest performance achieved with DA was calculated as 78% (population size 10, iteration 50). While the CART algorithm experienced a decline in some feature sets, it was able to increase the performance obtained from the original dataset from 68 to 78%. As with other algorithms, an increase in performance from 32 to 88% was observed with MLP, despite a bumpy performance in success with the feature sets from DA. 88% was recorded as the highest performance value obtained for the Dry bean dataset. MLR experienced the highest increase and decrease in performance in the DA algorithm. MLR's 68% performance in the original dataset went up to 81%. It increased its F-score value in GNB from 72 to 88%.

F-score values of dry bean dataset for MLR with population size 10, are given in Fig. 4, for each feature selection algorithms.

The features selected for each of the meta-heuristic algorithms applied to the Date fruit dataset are given in Table 9.

Analysing Table 9 according to the minimum fitness value for population size 10 gives the lowest fitness value for WOA and GOA at 100, for SSA, MFO, MVO and DA at 50 iterations. According to best fitness value, attributes were selected from 50% of attributes (17 out of 34) with WOA and MFO, 41.18% (14 out 34) with SSA, 40.06% (16 out of 34) with MVO, 26.47% (9 out of 34) with GOA, and 17.65% (6 out of 34) with DA. Analyzing algorithms with regards to standard deviation for population size 10, it was determined that WOA and GOA at iteration 100, SSA, MFO and MVO at iteration 10, while DA at iteration 10 was more stable compared to others. At population size 20, the minimum fitness value was achieved with WOA and MFO at 100 iterations, while it was achieved with SSA and GOA at 50 iterations and with MVO and DA at 10 iterations. According to the lowest fitness value, attributes were selected from 59.82% of attributes (20 out of 34) with WOA and MFO, 41.18% (14 out of 34) with SSA and GOA, 44.12% (15 out of 34) with MVO and 14.71% (5 out of 34) with DA. Considering the smallest value of the standard deviation for 20 iterations, it can be said that WOA, SSA and DA are most stable than others at 50 iterations, MFO at 10, and GOA at 100 iterations. The F-score value obtained by assigning the selected features to the ML algorithms and the F-score value obtained from the original feature set is given in Table 10.

For the Dry bean dataset, the highest F-score value taken from the original feature set was 0.84, achieved through CART. Classification on the original dataset through kNN with an F-score value of 68% increased success up to 85% after feature selection. However, for the DA algorithm, the performance has decreased to 51%. The CART algorithm was able to classify with a maximum increase of 2% in performance. Also, this algorithm did not increase the classification performance in the features obtained with DA but decreased it. MLP was able to increase classification success from 43% to a maximum of 72%. MLR was able to increase performance from 55 to 81%. The highest performance value achieved for the Date fruit dataset with GNB was 88%. GNB was able to increase the F-score value from 58 to 88%.

Table 7 Results of meta-heuristic algorithms on dry bean dataset

Algorithm	Population size	Iteration	Min	Mean	Std	Selected features
WOA	10	10	0.0882	0.1048	0.0215	2, 3, 4, 8, 10, 11, 16
		50	0.0838	0.0935	0.0062	2, 3, 4, 5, 6, 8, 10, 11, 12, 13, 14, 15, 16
		100	0.0855	0.0979	0.0117	2, 3, 4, 5, 6, 8, 10, 12, 15, 16
	20	10	0.0877	0.1042	0.0216	2, 3, 4, 5, 8, 9, 11, 12, 16
		50	0.0855	0.0991	0.0115	2, 3, 4, 6, 8, 9, 11, 14, 15, 16
		100	0.0877	0.0968	0.0096	2, 3, 4, 5, 6, 8, 9, 10, 11, 14
SSA	10	10	0.0877	0.0996	0.0140	2, 3, 4, 8, 9, 10, 11, 15, 16
		50	0.0852	0.0922	0.0036	2, 3, 4, 6, 8, 9, 11, 12, 16
		100	0.0879	0.0939	0.0028	2, 3, 4, 6, 8, 10, 15
	20	10	0.0852	0.0954	0.0059	2, 3, 5, 8, 11, 12, 13, 14, 15
		50	0.0850	0.0925	0.0035	2, 3, 4, 5, 8, 9, 10, 11, 14
		100	0.0838	0.0940	0.0041	2, 3, 4, 8, 11, 13, 16
MFO	10	10	0.0855	0.0954	0.0062	2, 3, 8, 9, 10, 12, 14, 16
		50	0.0835	0.0932	0.0037	2, 3, 4, 6, 8, 9, 10, 11, 12, 15, 16
		100	0.0869	0.0948	0.0067	2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15
	20	10	0.0857	0.0928	0.0043	2, 3, 4, 5, 6, 8, 9, 16
		50	0.0882	0.0933	0.0029	2, 3, 4, 6, 8, 9, 15, 16
		100	0.0887	0.0940	0.0030	2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 14
MVO	10	10	0.0872	0.0962	0.0072	2, 3, 5, 6, 8, 9, 11, 15, 16
		50	0.0850	0.0944	0.0045	2, 3, 4, 5, 6, 8, 10, 13, 15, 16
		100	0.0869	0.0967	0.0072	2, 3, 4, 8, 14, 15, 16
	20	10	0.0877	0.0940	0.0037	2, 3, 8, 9, 10, 11, 12, 13, 14
		50	0.0860	0.0944	0.0041	2, 3, 4, 5, 8, 10, 13, 15
		100	0.0882	0.0938	0.0032	2, 3, 4, 6, 8, 11, 12
GOA	10	10	0.0872	0.1043	0.0161	2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14
		50	0.0869	0.1013	0.0175	2, 3, 4, 6, 8, 12
		100	0.0860	0.1041	0.0164	2, 3, 4, 5, 8, 12
	20	10	0.0882	0.0991	0.0115	2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16
		50	0.0877	0.0962	0.0068	2, 3, 4, 9, 10, 15
		100	0.0872	0.0972	0.0095	2, 3, 4, 5, 8, 10, 11, 12, 13, 14, 15, 16
DA	10	10	0.1107	0.1328	0.0298	4, 6, 7, 8, 9, 10
		50	0.1160	0.1299	0.0083	2, 3, 10
		100	0.1120	0.1303	0.0106	2, 3, 4, 7, 8, 9, 10
	20	10	0.1013	0.1241	0.0099	3, 4, 6, 8, 9
		50	0.1027	0.1236	0.0097	2, 3, 4, 6, 7, 8, 9, 10, 12
		100	0.1120	0.1238	0.0081	8, 10, 11, 12

F-score values of date fruit dataset for MLR with population size 10, are given in Fig. 5, for each feature selection algorithms.

4 Conclusions

This paper applies meta-heuristic algorithms to improve the ability of ML algorithms by eliminating non-informative features for classification problems. To see the applicability of feature selection algorithms to different problems with different datasets, different algorithms with different populations and iteration sizes are used. The metaheuristic algorithms WOA, SSA, MFO, MVO, GOA and DA are compared on four different datasets: pumpkin seeds dataset, pistachio dataset, dry bean dataset and date fruit dataset. Both the original datasets and the dimension-reduced datasets are classified with five classification algorithms (kNN, CART, MLP, MLR and GNB), and the effect of feature reduction on classification successes is

Table 8 F-score values for the dry bean dataset

Algorithm	Iteration	kNN		CART		MLP		MLR		GNB	
	Original	0.17		0.68		0.32		0.68		0.72	
		10	20	10	20	10	20	10	20	10	20
WOA	10	0.77	0.77	0.68	0.68	0.82	0.84	0.7	0.7	0.82	0.88
	50	0.78	0.77	0.78	0.68	0.86	0.84	0.7	0.7	0.87	0.88
	100	0.77	0.77	0.69	0.68	0.85	0.84	0.7	0.7	0.86	0.87
SSA	10	0.77	0.78	0.69	0.64	0.8	0.81	0.7	0.71	0.87	0.87
	50	0.77	0.77	0.67	0.67	0.85	0.85	0.7	0.7	0.88	0.87
	100	0.77	0.77	0.63	0.68	0.81	0.82	0.7	0.7	0.85	0.82
MFO	10	0.78	0.77	0.68	0.68	0.85	0.84	0.71	0.7	0.84	0.88
	50	0.77	0.77	0.69	0.66	0.84	0.86	0.7	0.7	0.88	0.86
	100	0.77	0.77	0.66	0.66	0.84	0.8	0.7	0.7	0.87	0.87
MVO	10	0.78	0.78	0.68	0.64	0.83	0.81	0.71	0.71	0.87	0.87
	50	0.77	0.77	0.68	0.64	0.82	0.88	0.7	0.7	0.86	0.86
	100	0.77	0.77	0.65	0.64	0.81	0.8	0.7	0.7	0.85	0.87
GOA	10	0.77	0.77	0.66	0.66	0.84	0.87	0.7	0.7	0.87	0.86
	50	0.77	0.78	0.58	0.77	0.78	0.82	0.7	0.69	0.84	0.86
	100	0.77	0.77	0.6	0.68	0.9	0.87	0.7	0.7	0.85	0.87
DA	10	0.22	0.65	0.57	0.55	0.32	0.8	0.77	0.61	0.73	0.85
	50	0.78	0.25	0.76	0.63	0.79	0.4	0.55	0.79	0.66	0.74
	100	0.25	0.17	0.62	0.61	0.49	0.76	0.81	0.77	0.74	0.88

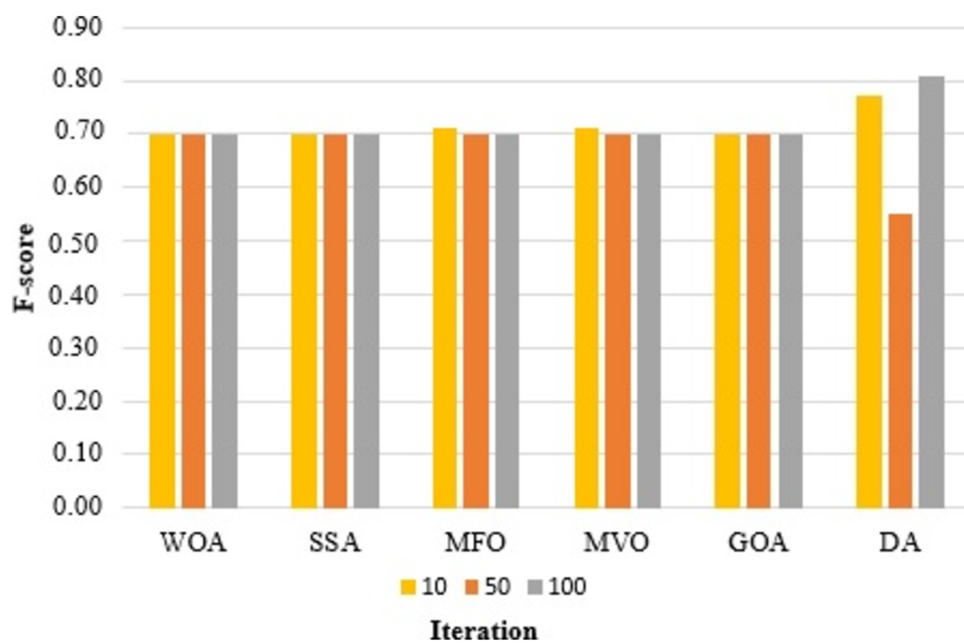

Fig. 4 F-score values for dry bean dataset

examined by using F-score. The experimental results show that the realized feature selection algorithms have great advantages for improving the classification accuracy of ML algorithms with reduced size dimensions. In other words, the realized feature selection algorithms have successfully selected the most important features that best represent the dataset.

Table 9 Results of meta-heuristic algorithms on date fruit dataset

Algorithm	Population size	Iteration	Min	Mean	Std	Selected features
WOA	10	10	0.1375	0.2603	0.0695	2, 6, 11, 17, 18, 21, 28
		50	0.1449	0.2214	0.0736	3, 4, 14, 15, 17, 21, 22, 25, 34
		100	0.1190	0.1917	0.0593	4, 6, 7, 10, 11, 12, 14, 15, 16, 17, 18, 21, 24, 25, 26, 32, 34
	20	10	0.1078	0.2216	0.0759	2, 4, 5, 6, 10, 11, 14, 15, 17, 18, 19, 20, 21, 22, 23, 25, 32, 33
		50	0.1264	0.1980	0.0533	4, 12, 14, 15, 16, 17, 19, 21, 23, 28
		100	0.1078	0.1864	0.0581	4, 11, 14, 15, 17, 19, 20, 21, 24, 25, 27, 28, 32, 34
SSA	10	10	0.1264	0.2803	0.0740	4, 9, 10, 11, 16, 17, 20, 21, 23, 24, 25, 27, 28, 32, 33
		50	0.1152	0.2017	0.0817	4, 5, 7, 10, 11, 16, 17, 18, 19, 22, 23, 24, 28, 34,
		100	0.1190	0.2024	0.0918	2, 3, 6, 7, 9, 12, 13, 14, 15, 16, 17, 18, 20, 21, 26, 27, 28, 32
	20	10	0.1375	0.2291	0.0836	4, 6, 9, 11, 14, 16, 17, 19, 20, 23, 25, 27, 32, 34
		50	0.1227	0.1519	0.0153	2, 4, 5, 6, 11, 14, 17, 19, 22, 23, 25, 26, 27, 28
		100	0.1227	0.1496	0.0376	4, 5, 10, 11, 13, 14, 15, 16, 17, 18, 19, 21, 22, 25, 27, 28, 32, 33, 34
MFO	10	10	0.1450	0.2617	0.0807	4, 6, 7, 9, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 32, 33, 34
		50	0.1040	0.2381	0.0939	4, 5, 7, 9, 10, 11, 15, 16, 17, 18, 19, 21, 23, 24, 28, 32, 33
		100	0.1227	0.2042	0.0881	4, 5, 6, 7, 12, 13, 14, 18, 20, 21, 24, 26, 27, 28, 32, 33, 34
	20	10	0.1338	0.2078	0.0764	3, 6, 7, 12, 13, 14, 17, 18, 19, 20, 24, 27, 28, 33
		50	0.1078	0.2001	0.0931	3, 4, 5, 11, 12, 13, 14, 17, 18, 19, 20, 23, 24, 26, 28, 32, 33
		100	0.1004	0.1817	0.0848	4, 5, 10, 12, 13, 14, 15, 17, 19, 20, 21, 22, 23, 24, 26, 28, 32, 33, 34
MVO	10	10	0.1413	0.2874	0.0773	3, 5, 6, 11, 15, 16, 17, 18, 19, 20, 21, 28, 33, 34
		50	0.1263	0.2864	0.0859	3, 6, 7, 12, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 27, 33
		100	0.1338	0.2766	0.0810	2, 6, 7, 9, 11, 12, 13, 14, 16, 17, 18, 22, 25, 27, 28, 32, 33, 34
	20	10	0.1115	0.2265	0.0878	4, 6, 7, 10, 11, 12, 14, 17, 19, 21, 22, 23, 25, 28, 33
		50	0.1338	0.2637	0.0881	4, 5, 9, 11, 12, 14, 17, 21, 25, 26, 32
		100	0.1152	0.2259	0.0948	6, 9, 10, 12, 15, 17, 18, 19, 21, 22, 23, 24, 32, 33
GOA	10	10	0.1413	0.2669	0.0746	2, 4, 7, 9, 13, 14, 16, 17, 18, 20, 22, 25, 26, 27, 33
		50	0.1338	0.2402	0.0755	3, 4, 15, 17, 19, 26, 27, 28, 32, 33, 34
		100	0.1338	0.1967	0.0587	3, 6, 9, 17, 19, 24, 25, 27, 32
	20	10	0.1375	0.2266	0.0703	4, 7, 10, 14, 16, 20, 23, 25, 32, 33
		50	0.1264	0.1776	0.0517	2, 3, 6, 7, 10, 11, 13, 15, 17, 18, 21, 25, 26, 32
		100	0.1301	0.1869	0.0495	2, 4, 9, 10, 12, 14, 15, 17, 18, 19, 23, 25, 26, 27, 33, 34
DA	10	10	0.1080	0.1326	0.0321	7, 8, 10, 12
		50	0.1053	0.1259	0.0106	2, 3, 4, 6, 7, 11
		100	0.1053	0.1344	0.0355	2, 3, 6, 7, 8
	20	10	0.1027	0.1253	0.0107	2, 3, 4, 7, 8
		50	0.1080	0.1264	0.0091	2, 3, 7, 8, 10, 11, 12
		100	0.1107	0.1256	0.0111	3, 4, 8, 11

Table 10 F-score values for date fruit dataset

Algorithm	Iteration	kNN		CART		MLP		MLR		GNB	
	Original	0.68		0.84		0.43		0.55		0.58	
		10	20	10	20	10	20	10	20	10	20
WOA	10	0.84	0.84	0.83	0.84	0.53	0.58	0.77	0.8	0.86	0.88
	50	0.83	0.85	0.84	0.83	0.68	0.68	0.78	0.77	0.88	0.86
	100	0.83	0.85	0.85	0.85	0.72	0.69	0.78	0.8	0.88	0.86
SSA	10	0.83	0.83	0.84	0.84	0.67	0.68	0.78	0.8	0.86	0.87
	50	0.84	0.82	0.84	0.83	0.71	0.59	0.79	0.78	0.84	0.88
	100	0.52	0.85	0.83	0.84	0.19	0.65	0.76	0.81	0.82	0.84
MFO	10	0.84	0.84	0.85	0.84	0.72	0.68	0.8	0.79	0.86	0.84
	50	0.85	0.83	0.84	0.81	0.62	0.64	0.8	0.78	0.83	0.84
	100	0.82	0.85	0.84	0.83	0.7	0.71	0.8	0.81	0.87	0.79
MVO	10	0.84	0.84	0.82	0.84	0.65	0.71	0.78	0.81	0.85	0.89
	50	0.84	0.83	0.81	0.83	0.73	0.65	0.8	0.71	0.83	0.88
	100	0.82	0.84	0.83	0.81	0.48	0.62	0.76	0.79	0.87	0.8
GOA	10	0.82	0.81	0.84	0.84	0.63	0.6	0.76	0.73	0.87	0.86
	50	0.83	0.83	0.83	0.86	0.71	0.55	0.76	0.8	0.85	0.88
	100	0.82	0.83	0.83	0.84	0.65	0.59	0.75	0.75	0.83	0.85
DA	10	0.51	0.52	0.7	0.71	0.17	0.22	0.22	0.69	0.53	0.68
	50	0.73	0.52	0.71	0.73	0.36	0.18	0.6	0.65	0.75	0.53
	100	0.51	0.52	0.67	0.69	0.18	0.22	0.66	0.7	0.63	0.69

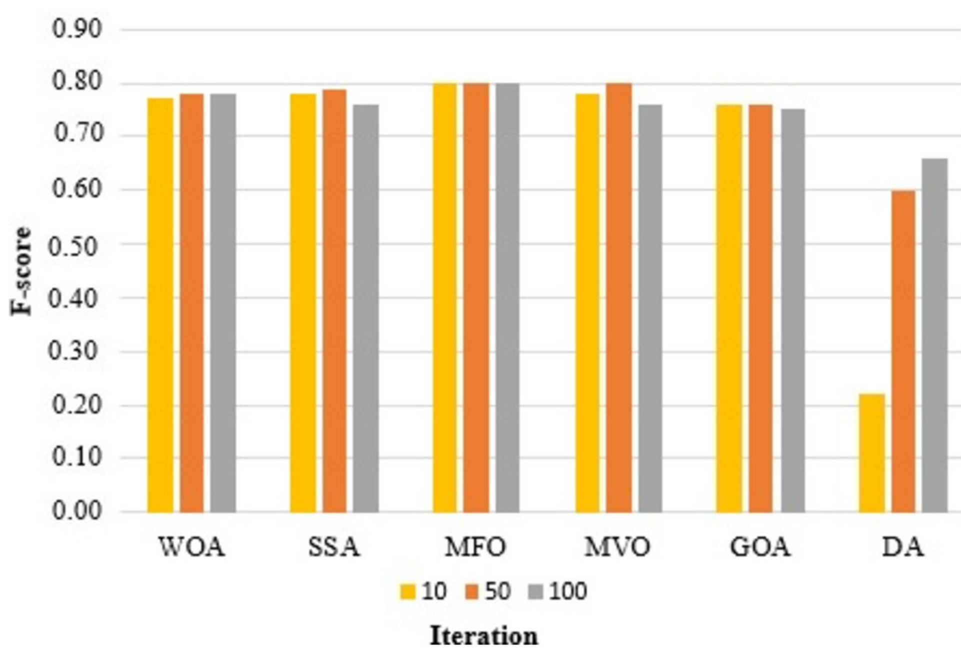

Fig. 5 F-score values for date fruit dataset

Author contributions ZG: Supervision, Conceptualization, Methodology, Writing & Reviewing & Editing. EE: Supervision, Conceptualization, Methodology, Writing, Experiments. MEÇ: Methodology, Writing, Experiments.

Funding No funding.

Data availability The datasets used in the study are public.

Declarations

Competing interests The authors declare no competing interests.

Ethical approval Not applicable.

References

1. Gupta, S.: Artificial intelligence in real life. In: Sundari, S., Chong, S.T., Prabu, M. (eds.) Outcomes of best practices in classroom research, 1st edn., pp. 305–308. L Ordine Nuovo Publication, Madurai (2021)
2. Farokhzadeh, S., Fakheri, B.A., Zinati, Z., Tahmasebi, S.: New selection strategies for determining the traits contributing to increased grain yield in wheat (*Triticum aestivum* L.) under aluminum stress. Genet. Resour. Crop Evol. **68**(5), 2061–2073 (2021). https://doi.org/10.1007/s10722-021-01117-4
3. Dönmez, E.: Enhancing classification capacity of CNN models with deep feature selection and fusion: a case study on maize seed classification. Data Knowl. Eng. **141**, 102075 (2022). https://doi.org/10.1016/j.datak.2022.102075
4. Pradhan, A.K., Swain, S., Kumar Rout, J.: Role of machine learning and cloud-driven platform in IoT-based smart farming. In: Satyanarayana, C., Gao, X.Z., Ting, C.Y. (eds.) Machine learning and internet of things for societal issues, pp. 43–54. Springer, Singapore (2022)
5. Raj, N., Perumal, S., Singla, S., Sharma, G.K., Qamar, S., Chakkaravarthy, A.P.: Computer aided agriculture development for crop disease detection by segmentation and classification using deep learning architectures. Comput. Electr. Eng. **103**, 108357 (2022). https://doi.org/10.1016/j.compeleceng.2022.108357
6. Gulzar, Y., Hamid, Y., Soomro, A.B., Alwan, A.A., Journaux, L.: A convolution neural network-based seed classification system. Symmetry **12**(12), 2018 (2020). https://doi.org/10.3390/sym12122018
7. Bhole, V., Kumar, A.: Analysis of convolutional neural network using pre-trained squeezenet model for classification of thermal fruit images. In: Mishra, D.K., Dey, N., Deora, B.S., Joshi, A. (eds.) ICT for competitive strategies, 1st edn., pp. 759–768. CRC Press, Boca Raton (2020)
8. Mahmood, A., Tiwari, A.K., Singh, S.K., Udmale, S.S.: Contemporary machine learning applications in agriculture: Quo Vadis? Concurr. Comput.: Pract. Exp. **34**(15), e6940 (2022)
9. Benos, L., Tagarakis, A.C., Dolias, G., Berruto, R., Kateris, D., Bochtis, D.: Machine learning in agriculture: a comprehensive updated review. Sensors **21**(11), 3758 (2021)
10. Mahmood, A., Singh, S.K., Tiwari, A.K.: Pre-trained deep learning-based classification of jujube fruits according to their maturity level. Neural Comput. Appl. **34**(16), 13925–13935 (2022)
11. Gupta, A.K., Mazumdar, B.D.: Multidimensional schema for agricultural data warehouse. Int. J. Res. Eng. Technol. **2**(3), 245–253 (2013)
12. Kapila, G., Vandana, B., Khaitan, A., Francis Avinash, A., Ajay Kumar, C.H.: Apple fruit classification and damage detection using pre-trained deep neural network as feature extractor. In: Saini, H.S., Singh, R.K., Tariq Beg, M., Mulaveesala, R., Mahmood, M.R. (eds.) Innovations in electronics and communication engineering, pp. 235–243. Springer, Singapore (2022)
13. Bhargava, A., Bansal, A.: Classification and grading of multiple varieties of apple fruit. Food Anal. Methods **14**(7), 1359–1368 (2021). https://doi.org/10.1007/s12161-021-01970-0
14. Xia, C., Yang, S., Huang, M., Zhu, Q., Guo, Y., Qin, J.: Maize seed classification using hyperspectral image coupled with multilinear discriminant analysis. Infrared Phys. Technol. (2019). https://doi.org/10.1016/j.infrared.2019.103077
15. Ali, A., Qadri, S., Mashwani, W.K., Belhaouari, S.B., Naeem, S., Rafique, S., Jamal, F., Chesneau, C., Anam, S.: Machine learning approach for the classification of corn seed using hybrid features. Int. J. Food Prop. **23**(1), 1110–1124 (2020). https://doi.org/10.1080/10942912.2020.1778724
16. Khojastehnazhand, M., Roostaei, M.: Classification of seven iranian wheat varieties using texture features. Expert Syst. Appl. (2022). https://doi.org/10.1016/j.eswa.2022.117014
17. Madhavan, J., Salim, M., Durairaj, U., Kotteeswaran, R.: Wheat seed classification using neural network pattern recognizer. Mater. Today: Proc. (2021). https://doi.org/10.1016/j.matpr.2021.03.226
18. Li, X., Fan, X., Zhao, L., Huang, S., He, Y., Suo, X.: Discrimination of pepper seed varieties by multispectral imaging combined with machine learning. Appl. Eng. Agric. **36**(5), 743–749 (2020). https://doi.org/10.13031/aea.13794
19. Sabanci, K., Aslan, M.F., Ropelewska, E., Unlersen, M.F.: A convolutional neural network-based comparative study for pepper seed classification: analysis of selected deep features with support vector machine. J. Food Process Eng. (2022). https://doi.org/10.1111/jfpe.13955
20. Bantan, R.A.R., Ali, A., Naeem, S., Jamal, F., Elgarhy, M., Chesneau, C.: Discrimination of sunflower seeds using multispectral and texture dataset in combination with region selection and supervised classification methods. Chaos **30**, 113142 (2020). https://doi.org/10.1063/5.0024017
21. Onmankhong, J., Ma, T., Inagaki, T., Sirisomboon, P., Tsuchikawa, S.: Cognitive spectroscopy for the classification of rice varieties: a comparison of machine learning and deep learning approaches in analyzing long-wave near-infrared hyperspectral images of brown and milled samples. Infrared Phys. Technol. **123**, 104100 (2022). https://doi.org/10.1016/j.infrared.2022.104100
22. Yang, X., Zhang, R., Zhai, Z., Pang, Y., Jin, Z.: Machine learning for cultivar classification of apricots (Prunus armeniaca L.) based on shape features. Sci. Hort. 256108524 (2019). doi: 10.1016/j.scienta.2019.05.051.
23. Sari, C.A., et al.: Papaya fruit type classification using LBP features extraction and naive bayes classifier. 2020 international seminar on application for technology of information and communication (iSemantic), pp. 28–33. IEEE, New York, (2020). https://doi.org/10.1109/iSemantic50169.2020.9234240
24. Iqbal, S.M., Gopal, A., Sankaranarayanan, P.E., Nair, A.B.: Classification of selected citrus fruits based on color using machine vision system. Int. J. Food Prop. **19**(2), 272–288 (2016). https://doi.org/10.1080/10942912.2015.1020439
25. Oliveira, A.N., Bolognini, S.R.F., Navarro, L.C., Delafiori, J., Sales, G.M., et al.: Tomato classification using mass spectrometry-machine learning technique: a food safety-enhancing platform. Food Chem. **398**, 133870 (2023). https://doi.org/10.1016/j.foodchem.2022.133870
26. Peres, A.M., Baptista, P., Malheiro, R., Dias, L.G., Bento, A., Pereira, J.A.: Chemometric classification of several olive cultivars from Trás-os-montes region (northeast of Portugal) using artificial neural networks. Chemometr. Intell. Lab. Syst. **105**(1), 65–73 (2011). https://doi.org/10.1016/j.chemolab.2010.11.001
27. Beyaz, A., Özkaya, M.T., İçen, D.: Identification of some spanish olive cultivars using image processing techniques. Sci. Hort. **225**, 286–292 (2017). https://doi.org/10.1016/j.scienta.2017.06.041
28. Sabzi, A., Abbaspour-Gilandeh, Y., García-Mateos, G.: A new approach for visual identification of orange varieties using neural networks and metaheuristic algorithms. Inform. Process. Agric. **5**(1), 62–172 (2018). https://doi.org/10.1016/j.inpa.2017.09.002
29. Fermo, I.R., Cavali, T.S., Bonfim-Rocha, L., Srutkoske, C.L., Flores, F.C., Andrade, C.M.G.: Development of a low-cost digital image processing system for oranges selection using hopfield networks. Food Bioprod. Process **125**, 181–192 (2021). https://doi.org/10.1016/j.fbp.2020.11.012

30. Al-Saif, A.M., Abdel-Sattar, M., Aboukarima, A.M., Eshra, D.H.: Identification of indian jujube varieties cultivated in Saudi Arabia using an artificial neural network. Saudi J. Biol. Sci. **28**(10), 5765–5772 (2021). https://doi.org/10.1016/j.sjbs.2021.06.019
31. Koklu, M., Sarigil, S., Ozbek, O.: The use of machine learning methods in classification of pumpkin seeds (*Cucurbita pepo* L). Genet. Resour. Crop Evol. **68**, 2713–2726 (2021). https://doi.org/10.1007/s10722-021-01226-0
32. Liu, Y., Wu, T., Yang, J., Tan, K., Wang, S.: Hyperspectral band selection for soybean classification based on information measure in FRS theory. Biosyst. Eng. **178**, 219–232 (2019). https://doi.org/10.1016/j.biosystemseng.2018.12.002
33. Koklu, M., Ozkan, I.A.: Multiclass classification of dry beans using computer vision and machine learning techniques. Comput. Electron. Agric. 174 (2020). doi: 10.1016/j.compag.2020.105507.
34. Esteki, M., Heydari, E., Simal-Gandara, J., Shahsavari, Z., Mohammadlou, M.: Discrimination of pistachio cultivars based on multi-elemental fingerprinting by pattern recognition methods. Food Control **124**, 107889 (2021). https://doi.org/10.1016/j.foodcont.2021.107889
35. Ozkan, I.A., Koklu, M., Saraçoğlu, R.: Classification of pistachio species using improved K-NN classifier. Prog. Nutr. (2021). https://doi.org/10.23751/pn.v23i2.9686
36. Koklu, M., Kursun, R., Taspinar, Y.S., Cinar, I.: Classification of date fruits into genetic varieties using image analysis. Math. Probl. Eng. (2021). https://doi.org/10.1155/2021/4793293
37. Dogan, M., Taspinar, Y.S., Cinar, I., Kursun, R., Ozkan, I.A., Koklu, M.: Dry bean cultivars classification using deep cnn features and salp swarm algorithm based extreme learning machine. Comput. Electron. Agric. **204**, 107575 (2023). https://doi.org/10.1016/j.compag.2022.107575
38. Taspinar, Y.S., Dogan, M., Cinar, I., Kursun, R., Ozkan, I.A., Koklu, M.: Computer vision classification of dry beans (*Phaseolus vulgaris* L.) based on deep transfer learning techniques. Eur. Food Res. Technol. **248**(11), 2707–2725 (2022). https://doi.org/10.1007/s00217-022-04080-1
39. Khan, M.S., Nath, T.D., Hossain, M.M., Mukherjee, A., Hasnath, H.B., Meem, T.M., Khan, U.: Comparison of multiclass classification techniques using dry bean dataset. Int. J. Cognit. Comput. Eng. **4**, 6–20 (2023). https://doi.org/10.1016/j.ijcce.2023.01.002
40. Macuácua, J.C., Centeno, J.A.S., Amisse, C.: Data mining approach for dry bean seeds classification. Smart Agric. Technol. **5**, 100240 (2023). https://doi.org/10.1016/j.atech.2023.100240
41. Dokeroglu, T., Deniz, A., Kiziloz, H.E.: A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing **494**, 269–296 (2022). https://doi.org/10.1016/j.neucom.2022.04.083
42. Alzaqebah, M., Briki, K., Alrefai, N., Brini, S., Jawarneh, S., Alsmadi, M.K., Mohammad, R.M.A., Almarashdeh, I., Alghamdi, F.A., Aldhafferi, N., Alqahtani, A.: Memory based cuckoo search algorithm for feature selection of gene expression dataset. Inf. Med. Unlocked (2021). https://doi.org/10.1016/j.imu.2021.100572
43. Xie, W., Wang, L., Yu, K., Shi, T., Li, W.: Improved multi-layer binary firefly algorithm for optimizing feature selection and classification of microarray data. Biomed. Signal Process Control **79**(1), 104080 (2023). https://doi.org/10.1016/j.bspc.2022.104080
44. Mirjalili, S.: Handbook of moth-flame optimization algorithm: variants, hybrids, improvements, and applications, 1st edn. CRC Press, Boca Raton (2022)
45. Ewees, A.A., El Aziz, M.A., Hassanien, A.E.: Chaotic multiverse optimizer-based feature selection. Neural Comput. Appl. **31**, 991–1006 (2019). https://doi.org/10.1007/s00521-017-3131-4
46. Nadimi-Shahraki, M.H., Zamani, H., Mirjalili, S.: Enhanced whale optimization algorithm for medical feature selection: a COVID-19 case study. Comput. Biol. Med. (2022). https://doi.org/10.1016/j.compbiomed.2022.105858
47. Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M.A., Omar, M.S.: Dynamic salp swarm algorithm for feature selection. Expert Syst. Appl. **164**, 113873 (2021). https://doi.org/10.1016/j.eswa.2020.113873
48. Wang, D., Chen, H., Li, T., Wan, J., Huang, Y.: A novel quantum grasshopper optimization algorithm for feature selection. Int. J. Approx. Reason. **127**, 33–53 (2020). https://doi.org/10.1016/j.ijar.2020.08.010
49. Zadsafar, F., Tabrizchi, H., Parvizpour, S., Razmara, J., Lotfi, S.: A model for mesothelioma cancer diagnosis based on feature selection using Harris Hawk optimization algorithm. Comput. Methods Programs Biomed. Update (2022). https://doi.org/10.1016/j.cmpbup.2022.100078
50. Varzaneh, Z.A., Hossein, S., Mood, S.E., Javidi, M.M.: A new hybrid feature selection based on improved equilibrium optimization. Chemometr. Intell. Lab. Syst. **228**, 104618 (2022). https://doi.org/10.1016/j.chemolab.2022.104618
51. Yousri, D., Abd Elaziz, M., Oliva, D., Abraham, A., Alotaibi, M.A., Hossain, M.A.: Fractional-order comprehensive learning marine predators algorithm for global optimization and feature selection. Knowl. Based Syst. (2022). https://doi.org/10.1016/j.knosys.2021.107603
52. Sreejith, S., Nehemiah, H.K., Kannan, A.: A clinical decision support system for polycystic ovarian syndrome using red deer algorithm and random forest classifier. Healthc. Anal. (2022). https://doi.org/10.1016/j.health.2022.100102
53. Qaraad, M., Amjad, S., Hussein, N.K., Elhosseini, M.A.: Large scale salp-based grey wolf optimization for feature selection and global optimization. Neural Comput. Appl. **34**(11), 8989–9014 (2022). https://doi.org/10.1007/s00521-022-06921-2
54. Mactina, F., Neduncheliyan, S.: Multi-classification of kidney abnormalities in sonography using the LOA-MFO and long-term recurrent convolutional network. Multimed. Tools Appl. (2023). https://doi.org/10.1007/s11042-023-16013-5
55. Acharya, S., Ganesan, S., Kumar, D.V., Subramanian, S.: A multi-objective multi-verse optimization algorithm for dynamic load dispatch problems. Knowl. Based Syst. **231**, 107411 (2021). https://doi.org/10.1016/j.knosys.2021.107411
56. Chhikara, S., Kumar, R.: MI-LFGOA: multi-island levy-flight based grasshopper optimization for spatial image steganalysis. Multimed. Tools Appl. **79**(39–40), 29723–29750 (2020). https://doi.org/10.1007/s11042-020-09328-0
57. Faramarzi, A., Gandomi, H.M.M.S.: Marine predator algorithm, a nature-inspired metaheuristic. Int. J. Expert Syst. Appl. **52**, 113377 (2020). https://doi.org/10.1016/j.eswa.2020.11337
58. Chen, X., Qi, X., Wang, Z., Cui, C., Wu, B., Yang, Y.: Fault diagnosis of rolling bearing using marine predators algorithm-based support vector machine and topology learning and out-of-sample embedding. Measurement (2021). https://doi.org/10.1016/j.measurement.2021.109116
59. Yousri, D., Hasanien, H.M., Fathy, A.: Parameters identification of solid oxide fuel cell for static and dynamic simulation using comprehensive learning dynamic multi-swarm marine predators algorithm. Energy. Conv. Manag. **228**, 113692 (2021). https://doi.org/10.1016/j.enconman.2020.113692
60. Abdel-Basset, M., El-Shahat, D., Chakrabortty, R.K., Ryan, M.: Parameter estimation of photovoltaic models using an improved marine predators algorithm. Energy. Conv. Manag. **227**, 113491 (2021). https://doi.org/10.1016/j.enconman.2020.113491
61. Gharehchopogh, F.S., Gholizadeh, H.: A comprehensive survey: whale optimization algorithm and its applications. Swarm Evol. Comput. **48**, 1–24 (2019). https://doi.org/10.1016/j.swevo.2019.03.004

62. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl. Based Syst. **89**, 228–249 (2015). https://doi.org/10.1016/j.knosys.2015.07.006
63. Wang, M., Liang, Y., Hu, Z., Chen, S., Shi, B., Heidari, A.A., Zhang, Q., Chen, H., Chen, X.: Lupus nephritis diagnosis using enhanced moth flame algorithm with support vector machines. Comput. Biol. Med. (2022). https://doi.org/10.1016/j.compbiomed.2022.105435
64. Sapre, S., Mini, S.: Emulous mechanism based multi-objective moth-flame optimization algorithm. J. Parallel Distrib. Comput. **150**, 15–33 (2021). https://doi.org/10.1016/j.jpdc.2020.12.010
65. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. **114**, 163–191 (2017). https://doi.org/10.1016/j.advengsoft.2017.07.002
66. Mirjalili, S., Mirjalili, S.M., Hatamlou, A.: Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Appl. **27**, 495–513 (2016). https://doi.org/10.1007/s00521-015-1870-7
67. Fathy, A., Rezk, H.: Multi-verse optimizer for identifying the optimal parameters of PEMFC model. Energy **143**, 634–644 (2018). https://doi.org/10.1016/j.energy.2017.11.014
68. Suresh, T., Brijet, Z., Sheeba, T.B.: CMVHHO-DKMLC: a Chaotic Multi Verse Harris Hawks optimization (CMV-HHO) algorithm based deep kernel optimized machine learning classifier for medical diagnosis. Biomed. Signal Process Control (2021). https://doi.org/10.1016/j.bspc.2021.103034
69. Abasi, A.K., Khader, A.T., Al-Betar, M.A., Naim, S., Makhadmeh, S.N., Alyasseri, Z.A.A.: Link-based multi-verse optimizer for text documents clustering. Appl. Soft Comput. **87**, 106002 (2020). https://doi.org/10.1016/j.asoc.2019.106002
70. Ali, T.A., Xiao, Z., Mirjalili, S., Havyarimana, V.: Efficient design of wideband digital fractional order differentiators and integrators using multi-verse optimizer. Appl. Soft Comput. **93**, 106340 (2020). https://doi.org/10.1016/j.asoc.2020.106340
71. Ewees, A.A., Abd Elaziz, M.: Performance analysis of Chaotic Multi-Verse Harris Hawks optimization: a case study on solving engineering problems. Eng. Appl. Artif. Intell. **88**, 103370 (2020). https://doi.org/10.1016/j.engappai.2019.103370
72. Saremi, S., Mirjalili, S., Lewis, A.: Grasshopper optimisation algorithm: theory and application,advances. Eng. Softw. **105**, 30–47 (2017). https://doi.org/10.1016/j.advengsoft.2017.01.004
73. Liu, Y., Cheng, Y., Zhang, Z., et al.: Multi-information fusion fault diagnosis based on KNN and improved evidence theory. J. Vib. Eng. Technol. **10**, 841–852 (2022). https://doi.org/10.1007/s42417-021-00413-8
74. Chen, J., Li, Z., Wang, X., et al.: A hybrid monotone decision tree model for interval-valued attributes. Adv. Comp. Int. **2**, 12 (2022). https://doi.org/10.1007/s43674-021-00016-6
75. Shamrat, J.M., Ranjan, F.M., Hasib, R., Yadav, K.M., Siddique, A.H.: Performance evaluation among ID3, C4.5, and CART decision Tree algorithm. In: Ranganathan, G., Bestak, R., Palanisamy, R., Rocha, Ã. (eds.) Pervasive computing and social networking, vol. 317. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-5640-8_11
76. Taud, H., Mas, J.F.: Multilayer perceptron (MLP). In: Camacho Olmedo, M.T., Paegelow, M., Mas, J.F., Escobar, F. (eds.) Geomatic approaches for modeling land change scenarios, pp. 451–455. Springer, Verlag (2018)
77. Altay, O., Varol, A.E.: A novel hybrid multilayer perceptron neural network with improved grey wolf optimizer. Neural. Comput. Appl. (2022). https://doi.org/10.1007/s00521-022-07775-4
78. Bhattacharjee, P., Dey, V., Mandal, U.K., Paul, S.: Quantitative risk assessment of submersible pump components using interval number-based multinomial logistic regression (MLR) model. Reliab. Eng. Syst. Saf **226**, 108703 (2022). https://doi.org/10.1016/j.ress.2022.108703
79. Wang, Y.: A multinomial logistic regression modeling approach for anomaly intrusion detection. Comput. Secur. **24**(8), 662–674 (2005). https://doi.org/10.1016/j.cose.2005.05.003
80. Cataldi, L., Tiberi, L., Costa, G.: Estimation of MCS intensity for Italy from high quality accelerometric data, using GMICEs and Gaussian Naive Bayes classifiers. Bull. Earthq. Eng. **19**, 2325–2342 (2021). https://doi.org/10.1007/s10518-021-01064-6
81. Lou, W., Wang, X., Chen, F., Chen, Y., Jiang, B., et al.: Sequence based prediction of DNA-binding proteins based on hybrid feature selection using Random Forest and Gaussian Naive Bayes. PLoS ONE (2014). https://doi.org/10.1371/journal.pone.0086703
82. Hu, G., Zhong, J., Wang, X., Wei, G.: Multi-strategy assisted chaotic coot-inspired optimization algorithm for medical feature selection: a cervical cancer behavior risk study. Comput. Biol. Med. **151**, 106239 (2022). https://doi.org/10.1016/j.compbiomed.2022.106239
83. Kunhare, N., Tiwari, R., Dhar, J.: Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm. Comput. Electr. Eng. (2022). https://doi.org/10.1016/j.compeleceng.2022.108383
84. Cinar, I., Koklu, M.: Determination of effective and specific physical features of rice varieties by computer vision in exterior quality inspection. Selcuk J. Agric. Food Sci. **35**(3), 229–243 (2021)

Zeynep Garip received Ph.D. degree from the Department of Mechatronics Engineering, Sakarya University, Turkey, in 2018. She is currently an Assoc. Professor at the Sakarya University of Applied Sciences, Turkey. Her current research interests include metaheuristics algorithms and image processing.

Ekin Ekinci received Ph.D. degree from the Department of Computer Engineering, Kocaeli University, Turkey, in 2019. She is currently an Assist. Professor at the Sakarya University of Applied Sciences, Turkey. Her current research interests include artificial intelligence, machine learning and nature language processing.

Murat Erhan Çimen received Ph.D. degree from the Department of Electrical and Electronics Engineering, Sakarya University of Applied Sciences, Turkey, in 2022. He is currently an Assist. Professor at the Sakarya University of Applied Sciences, Turkey. He performs research in the areas of control systems, engineering education, circuits, chaos, model predictive control and meta-heuristic algorithm.