Evolutionary Intelligence (2022) 15:2631–2650
https://doi.org/10.1007/s12065-021-00634-6

RESEARCH PAPER

A comparative analysis of meta-heuristic optimization algorithms for feature selection and feature weighting in neural networks

P. M. Diaz[1] · M. Julie Emerald Jiju[2]

Received: 10 April 2021 / Revised: 11 June 2021 / Accepted: 23 June 2021 / Published online: 8 July 2021

Abstract
Feature selection and feature weighting are frequently used in machine learning for processing high dimensional data. It reduces the number of features in the dataset and makes the classification process easier. Meta-heuristic algorithms are widely adopted for feature selection and feature weighting due to their enhanced searching ability. This paper compares five different meta-heuristic optimization algorithms that are recently introduced for feature selection and feature weighting in artificial neural networks. This includes chimp optimization algorithm, tunicate swarm algorithm, bear smell search algorithm, antlion optimization algorithm and modified antlion optimization algorithm. Experimental evaluations are performed on five different datasets to illustrate the significant improvements observed during classification process of all the algorithms utilised in the comparative analysis. Both tunicate swarm algorithm and chimp optimization algorithm has gained better classification accuracy than other algorithms. However, all these algorithms are found to be more effective for feature selection and feature weighting processes.

Keywords Feature selection · Feature weighting · Machine learning · Meta-heuristic algorithms · Artificial neural networks

1 Introduction

During data processing, it is difficult to extract information or interpret the pattern from the data. Hence, machine learning models automated with training algorithms are widely employed to handle the big data. The exponential increase in database has increased the necessity of machine learning. Therefore, it is applied in several fields from military to medicine for extracting the necessary information. This further increased the need of machine learning to develop practical software for robot control, natural language processing, speech recognition, computer vision and other applications. Many artificial Intelligence (AI) system developers now acknowledge that, for many purposes, training a system with samples of desired input-output behaviour can be significantly easier than manual programming with assumptions on the required response for all potential inputs. Machine learning has widespread impact in computer science and variety of businesses that deal with data-intensive issues in consumer services, logistics chain control, and problem diagnostics in complex systems. To address the limitations faced in big data handling and the challenges encountered in machine-learning [1].

Classification is one of the extensively used techniques in the field of data mining and machine learning that requires set of features for learning process [2]. However, improving the learning ability of classification algorithms is more complex especially for dataset containing huge amount of features. Moreover, it makes the classification process tedious and thus a relatively longer time is required for learning every characteristic of the training data. This is due to the existence of redundant and irrelevant features in data which complicates the performance of learning algorithms thereby increasing the computation time [3]. It is necessary to eliminate these irrelevant features from the dataset to achieve effective learning process. Hence, optimal feature selection and feature weighting techniques are required for selecting the relevant features and improving the classification accuracy [4]. These processes reduce the dimensionality of data and make the learning process more efficient by reducing the time taken for learning task. Artificial Neural Networks

✉ P. M. Diaz
pauldiaz71@gmail.com

[1] Dedicated Juncture Researcher's Association, Kulasekharam, Kanyakumari, Tamilnadu, India

[2] Department of MCA, CSI Institute of Technology, Thovalai, Kanyakumari, Tamil Nadu, India

(ANNs) are adopted for classification in data mining field due to its higher performance [5]. It is a universal function approximation algorithm for modelling linear and non-linear data with a desired accuracy [6]. The conventional statistical methods have certain drawbacks like complications in the fusion of secondary data and so on. Thus, artificial neural networks are considered as the suitable alternative for these conventional statistical methods. ANNs possess many advantages such as easily adapting to different kinds of data, arbitrary decision boundary capabilities and their non-parametric nature. The learning of training data in neural networks typically takes place in an iterative way which considers all the patterns in the dataset for learning. Therefore, ANNs are called as the data dependent models [7]. During training, the weights of the ANN are adjusted until the actual output of the network and desired output of the network are as close as possible. Hence, ANNs can be effectively utilised for mapping an input to a desired output, for classifying data and for learning the patterns in the dataset provided.

Feature selection (FS) and feature weighting (FW) are the vital and broadly used data pre-processing methods in machine learning during classification [8]. FS is a combinatorial search problem that eliminates the redundant and irrelevant features and preserves the relevant features related to the dataset [9, 10]. Hence, the feature selection process minimises the number of features in the dataset and thereby speeds up the learning process by reducing computational complexity. This reduction in number of features makes the dataset easier to understand and manageable for further classification process [11]. FW is a continuous search problem in which the weights are allotted to features based on their relevance [12]. It approximates the optimal degree of influence of distinct features. Depending upon the individual feature values of the query and instance, the weights are assigned to the features dynamically [13]. FW approaches are suitable when the relevancy of features varies in data [14]. During classification process, each feature in the dataset will have different contribution. That is, some features will be more important than others while solving the classification problem. Hence, higher weights are allotted to the relevant features and lower weights are allotted to the less relevant and redundant features.

The techniques in FS and FW are classified under two methods namely, filter method and wrapper method. The filter method is used for filtering the insignificant features which contains lesser option during data analysis. It does not employ any learning algorithm for evaluating the features. The filter methods select a subset with large number of features or even select all the features in the dataset. Thus, a suitable threshold is necessary for choosing the subset. The selected features from the filter method are analysed based on data characteristics like information measures, correlation, consistency and distance in the feature space. The wrapper method uses predictive accuracy of a predetermined learning algorithm for determining the quality of the selected features. Generally, meta-heuristic algorithms are employed as learning algorithms in wrapper methods. The wrapper methods are mostly applied for feature weighting and feature selection process because filter methods have low classification accuracy than wrapper methods [15, 16].

Meta-heuristic algorithms are nature inspired algorithms that mimic the natural activities for solving optimization problem in several computations. The development of nature inspired algorithm lies in the point that it takes its sole inspiration from nature. These inspirations from nature have the ability to solve complex optimization problems. Nature serves as an abundant and massive resource of inspiration for solving complex as well as hard computing problem in the field of data science as it possesses extremely diverse, robust, complex, dynamic and fascinating substances. It always helps to determine optimal solutions for stochastic problems thereby maintaining a good balance between its key elements. This is the principle behind meta-heuristic optimization algorithms. An optimization algorithm always focuses on exploring and exploiting a search space to maintain a good balance between them [17]. In an algorithm, the exploration phase explores several best locations in the search space while exploitation phase searches the optimal solutions over the best locations [18]. There are many optimization algorithms based on exploration and exploitation process. Each optimization algorithm has its own nature and complexity making it efficient in specific optimization problem and they may be ineffective in other optimization problems. Thus, there is always a need for new optimization algorithms.

The optimization process for high dimensional spaces has become more complex under noisy environment because the data is not enough to create a complete mathematical model [19]. Hence, a powerful algorithm is needed for high-dimensional optimization issues. Moreover, no-free-lunch theorem had described that the existing optimization algorithms are not applicable for all optimization problems [20]. This motivates the analysts to develop new and more effective optimization algorithms for solving specific problems in various fields. Furthermore, each meta-heuristic algorithm has similar performance on every optimization problems. Thus, the unresolved problems in existing algorithms can be solved by introducing new meta-heuristic algorithms. In meta-heuristic algorithms, the search space exploration is well accomplished and global optimum exploitation is very consistent than other optimization algorithms. Also, it does not get stuck in the local optimum which makes it appropriate for solving problems in engineering sector. Different combinations of metaheuristic algorithms are also introduced in integration with different concepts. For instance, different versions of PSO invented for FS in high-dimensional data include fast hybrid PSO [21], bare-bones PSO [22], variable-size cooperative co-evolutionary PSO [23], and multi-objective PSO with fuzzy cost [24].

Numerous optimization algorithms are developed from the behaviour of some animals or insects in nature namely ant colonies, bees swarm and so on. This is because the biological activities of birds and animals are responsible for specific roles both individually and as a group, to achieve a specific task in their daily routine or lifetime. As a result, they have attracted the attention of data analysts to resolve numerous difficulties in science and engineering sector [25]. For example, Particle Swarm Optimization (PSO) is inspired from the biological behaviour of bird flocking and fish schooling [26], Lion Optimization Algorithm (LOA) simulates the activities of lions and their co-operation characteristics [27], Social Spider Optimization (SSO) algorithm is inspired from the nature of spiders [28], Whale Optimization Algorithm (WOA) imitates the actions of hump-back whales [29], Grey Wolf Optimizer (GWO) imitates the hunting skill and social leadership of grey wolves [30, 31], Artificial Bee Colony (ABC) algorithm mimics the cooperative behaviour of bee colonies [32], Ant Colony Optimization (ACO) simulates the food searching behaviour of ant colonies [33, 34] and so on. These algorithms are applied in different fields like data mining, machine learning and engineering design.

The main aim of this research is to present a comparative study on the impact of different metaheuristic based feature selection in classification accuracy. It also determines how the accuracy and computational time are enhanced based on optimization algorithms for better improvement in classification. This paper compares different meta-heuristic algorithms implemented in machine learning for feature selection and feature weighting practices. Five new meta-heuristic optimization techniques of state-of-the-art researches that have been presented recently which are derived merely on the behaviour of some insects or animals to mimic the biological nature in hunting and food searching processes are utilised in this study. It helps to realize the biological behaviour and response in diverse situations in nature to create solutions to complex issues by inspiring analogical reasoning and thinking of these insects or animals. The optimization algorithms considered for comparative analysis include Chimp Optimization Algorithm (ChOA), Bear Smell Search Algorithm (BSSA), Tunicate Swarm Algorithm (TSA), Ant Lion Optimization (ALO) and Modified Ant Lion Optimization (MALO). The features selected through these meta-heuristic algorithms are experimented to determine the accuracy with a classification framework.

This paper is formulated as follows. The background process of how this comparative study is accomplished is explained in Sect. 2. Different optimization based feature selection and feature weighting processes that are considered for analysis in this research are exemplified in Sect. 3. Section 4 describes the artificial neural network employed to estimate the accuracy of these optimization algorithms in classification. The details regarding the experimental setup and results are given in Sect. 5. Finally, Section 6 illustrates the conclusion and future works.

2 Background overview

The data preparation process is essential to achieve high classification performance and make the data more relevant for classification. Hence, it is necessary to pre-process the data before using machine learning algorithms, in order to get good classification performance. In this research, the dataset initially undergoes pre-processing to manage the feature dominance and outlier problems concerning numerical range. Data normalization is a pre-processing technique in which the data is scaled or altered to ensure that each feature contributes equally. It involves transforming characteristics into a similar range so that larger numeric feature values do not dominate smaller numeric feature values. The quality of data used to create a generalised predictive model in classification problem is critical to the success of machine learning techniques. The major goal is to reduce the bias of those features whose numerical contribution to pattern classification is stronger. If the relative importances of characteristics are unknown, data features are rendered equally essential when predicting the output class of an unknown instance. Since all of the features in the data contribute equally to the learning process, it is highly beneficial for statistical learning approaches. The tanh normalization is the widely used method for data pre-processing because it increases the classification performance compared to well-known normalization methods like min-max and z-score [35]. This normalization controls both the dominant features and influence of outliers before classification process. The normalization is done by evaluating the statistical properties of each feature. To obtain the normalized feature y_i, the standard deviation and the mean of each transformed feature are calculated as represented in Eq. (1).

$$y_i = \frac{1}{2}\left\{ tanh\left(0.01\left(\frac{y_i - \mu_i^N}{\sigma_i^N}\right)\right) + 1 \right\} \tag{1}$$

where the standard deviation of ith feature is denoted by σ_i^N and the mean of ith feature is denoted by μ_i^N.

After normalization, the feature selection and feature weighting processes are done in the dataset using optimization algorithms for altering the feature space. The combination of feature selection and feature weighting processes expands the search space immensely while searching for an optimal solution. Thus, it causes the optimization algorithms to get trapped in the local optima. Hence, meta-heuristic algorithms are utilised for feature selection and feature weighting since it has effective exploration and exploitation abilities [36]. The significant features are selected by means of feature selection and feature weighting approaches. This eliminates the irrelevant features and selects only the relevant features for the next stage. Here, five evolutionary algorithms utilised for feature subset selection such as Chimp Optimization Algorithm (ChOA), Ant Lion Optimization (ALO) algorithm, Tunicate Swarm Algorithm (TSA), Bear Smell Search Algorithm (BSSA) and Modified Ant Lion Optimization (MALO) algorithm are taken for comparison. To estimate the accuracy through classification, a feed forward neural network is adopted since it offers high rate of classification. The feed forward neural network is trained with features selected from the evolutionary algorithms. After classification, the results are compared for five different evolutionary algorithms and the performance of each algorithm is evaluated. The flowchart of this comparative study is shown in Fig. 1.

3 Feature selection and feature weighting algorithms

3.1 Chimp optimization algorithm

Chimp optimization algorithm mimics the diverse intelligence and the sexual motivation of chimps during hunting process [37]. Generally, the society of chimp is a fission fusion society where the combination of society is time variant [38]. Each chimp in the society has an individual task based on its distinct ability and this task may vary with time. Consequently, every chimp in the group tries to find the search space individually with its distinct skill. For a successful hunting process, each chimp is given a certain task such as driving, blocking, chasing and attacking [39]. The driver chimp drives the prey towards the search space without trying to snatch them. The barrier chimp blocks the prey by creating a barrier around the escape route of the prey. Chaser chimp chases after the prey and tries to catch them. At last, the attacker chimp attacks the prey by predicting the escape route of the prey. The attackers should be smart enough to predict the next movements of the prey. Thus, the attackers have a major role in the hunting process. After a successful hunt, they are compensated with a larger piece of meat. The attackers are selected based on the physical ability, smartness and age of the chimps. Moreover, the chimps can change their tasks during certain hunt or can maintain their tasks throughout the whole process [40].

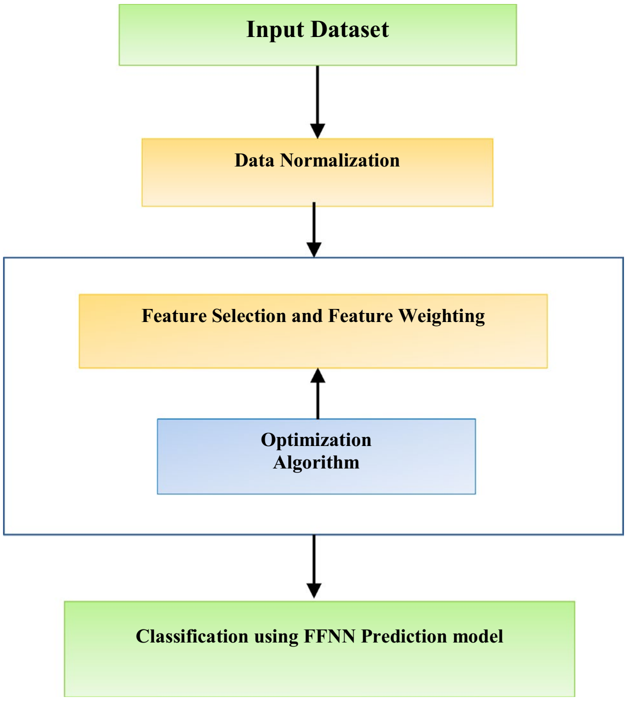
Input Dataset
Data Normalization
Feature Selection and Feature Weighting
Optimization Algorithm
Classification using FFNN Prediction model

Fig. 1 Schematic diagram showing the overview of metaheuristic-optimization based comparative analysis

In general, the meat obtained from the hunting process is deliberated as an exchange for social favours like grooming, sex or specified support [40]. Hence, by initiating the new domain of benefits, smartness will possibly have an implicit result on the hunt. These social inducements are normally used only by chimps and humans. As a consequence, it provides a great advantage in the hunting process of chimps. Further, the sexual motivation instigates the chimps to behave chaotically during the final stage. Thus, all chimps give up their own tasks and try to obtain the meat individually. The hunting behaviour of chimps can be categorized into two phases such as exploration and exploitation phase. In exploration phase, the chimp drives, blocks and chases the prey and in exploitation phase, the chimp attacks the prey. The hunting process of chimp is carried out by the exploration and exploitation phase as mathematically modelled in Eqs. (2) and (3).

$$f = \left| w.C_{prey}(t) - y.C_{chimp}(t) \right| \tag{2}$$

$$C_{chimp}(t+1) = C_{prey}(t) - z.f \tag{3}$$

where f implies the mathematical model for driving and chasing. C_{chimp} is denoted as the position vector of the chimp, C_{prey} is denoted as the position vector of the prey, and the current iteration number is denoted as t. The coefficient vectors are given as z,w and y estimated from Eqs. (4), (5) and (6).

$$z = 2.x.v_1 - x \tag{4}$$

$$w = 2.v_2 \tag{5}$$

$$y = \mathit{chaotic\ value} \tag{6}$$

where the random vectors are denoted as v_1 and v_2 in the range of [0, 1] and the chaotic vector is denoted as y which is computed by various chaotic maps. This signifies the influence of sexual motivation of chimps in hunting the prey. Here, x is reduced from 2.5 to 0 non-linearly by iteration process in both exploration and exploitation phase.

Exploration phase. The hunting process is usually carried out by the attackers but in rare cases, the drivers, barriers and chasers will also take part in the hunt. During the hunting process, the optimum location of the prey is not known at the first iteration. To overcome this limitation, the location of the attacker is assumed as the location of the prey. Subsequently, depending on the attacker's position, the location of driver, barrier and chaser will be updated. Hence, four of the best attained solutions are saved and the location of other chimps is updated according to the location of the best chimp. This process is demonstrated in Eqs. (7)–(9).

$$\begin{aligned} f_A &= \left|w_1 C_A - y_1 C\right|, f_B = \left|w_2 C_B - y_2 C\right|, \\ f_C &= \left|w_3 C_C - y_3 C\right|, f_D = \left|w_4 C_D - y_4 C\right| \end{aligned} \tag{7}$$

$$\begin{aligned} C_1 &= C_A - z_1(f_A), C_2 = C_B - z_2(f_B), C_3 \\ &= C_C - z_3(f_C), C_4 = C_D - z_4(f_D) \end{aligned} \tag{8}$$

$$C(t+1) = \frac{C_1 + C_2 + C_3 + C_4}{4} \tag{9}$$

where, f_A,f_B,f_C andf_D are the best solutions obtained for attacker, barrier, chaser and driver respectively. Also, C_A, C_B, C_C and C_D refers to the position vector of the attacker, barrier, chaser and driver respectively. The location of a chimp (search agent) in the search space is updated based on the location of other chimps. The final location of the chimp is randomly positioned in the circle based on the location of the attacker, barrier, driver and chaser.

Exploitation phase. In the chimp's society, the social incentives like grooming and sex depends upon the hunting process. The chimps try to snatch the prey for social favours by abandoning their own hunting task in the final stage. The chimps act chaotically for snatching the hunting meat and this chaotic behaviour is modelled by using chaotic maps. This improves the performance of ChOA algorithm. Moreover, this deterministic process also provides random behaviour. For all chaotic maps, the initial value is taken as 0.7 because of different behaviours [41]. The updating process in this method is given as in Eq. (10).

$$C_{chimp}(t+1) = \begin{cases} C_{prey} - z.f & \mathit{if}\,\mu < 0.5 \\ \mathit{Chaotic}_{value} & \mathit{if}\,\mu > 0.5 \end{cases}. \tag{10}$$

where μ is a random number in the range [0, 1]. The Pseudocode for chimp optimization algorithm is given in Algorithm 1.

Algorithm 1 Pseudocode for chimp optimization algorithm

```
Input: Neural network, Training data, total iterations
Initialize x,y,z and w.
Output: optimal feature weights and selection
//Start
Initialize chimp population $C_i$ at randomly.
Calculate the position of each chimp.
Assign chimps randomly into independent groups
Calculate fitness of ChOA using FFNN
Find the best search agent for Attacker ($C_A$), Barrier ($C_B$), Chaser ($C_C$) and Driver ($C_D$).
while current iteration, $t < Max.$ iteration, T do
    for i=1 to pop do
        Extract the chimps group to update x,y and z.
        Calculate w and n by using x,y and z.
    end for
    for each search chimp do
        if ($\mu < 0.5$) then
            if ($|w|$ <1) then
                Update the position of the current search agent.
            else
                Select random search agent
            end if
    end for
    Update x,y,z,w and $C_A$,$C_B$,$C_C$,$C_D$.
    t=t+1
end while
return  $C_A$
```

3.2 Tunicate swarm algorithm

Tunicate swarm algorithm is a meta-heuristic optimization algorithm which mimics the jet propulsion and swarm behaviours of tunicates. The tunicates are bright bio-luminescent, cylindrical shaped organisms that have one closed end and one open end [42]. It produces a pale blue-green light that can be viewed from a minimal distance. The tunicate size will be varied from few cm to 4 m [43]. Each tunicate contains certain gelatinous tunic useful in joining all the individuals. Also, each tunicate produces jet propulsion through its open end by inhaling water from the ocean and exhaling through atrial siphons. This jet propulsion helps the tunicate to migrate vertically in the ocean. It is the only animal in the ocean with such fluid jet like propulsion. Tunicates are typically located at 500-800 m depth in the ocean and moves towards the upper surface of water during night time. Generally, tunicates have the capability to locate the food source in the ocean. For this food searching process, the tunicate uses two behaviours such as jet propulsion and swarm intelligence. For jet propulsion process, the tunicates should satisfy three conditions such as avoiding conflict among the search agents, moving towards the location of the best search agent and converging towards the best search agent. Then, the swarm intelligence process updates the position of tunicates towards the best optimal solution.

Preventing the conflicts among search agents. To prevent the conflicts among other tunicates (search agents), the vector **L** is used to calculate the location of new search agent as shown in Eqs. (11–13).

$$\mathbf{L} = \frac{\mathbf{G}F}{\mathbf{S}F} \tag{11}$$

$$\mathbf{G}F = x_2 + x_3 - \mathbf{W_f} \tag{12}$$

$$\mathbf{W_f} = 2.x_1 \tag{13}$$

where, social forces among the search agents are represented by $\mathbf{S}F$, the flow of the water in deep ocean is denoted as $\mathbf{W_f}$ and $\mathbf{G}F$ denotes the gravitational force. The variables x_1, x_2 and x_3 are random numbers that lies in the range [0, 1]. The vector $\mathbf{S}F$ is calculated as in Eq. (14).

$$\mathbf{S}F = \lfloor S_{min} + x_1.S_{max} - S_{min} \rfloor \tag{14}$$

where, the initial speed for social interaction is denoted as S_{min} and the subordinate speed for social interaction is denoted as S_{max}.

Movement towards the direction of best neighbour. After preventing the conflict among other tunicates, the search agents shift towards the direction of the best search agent. The distance between food source and search agent is represented as **D** calculated as in Eq. (15).

$$\mathbf{D} = \mathbf{F} - r_{and}.\mathbf{P}(t) \tag{15}$$

where the location of the tunicate is represented as $\mathbf{P}(t)$, the location of food source is represented as **F** which is optimum and r_{and} denotes the random number in the range [0, 1].

Converge towards the best search agent. The search agent can maintain its location near the best search agent. The position of tunicate is given in Eq. (16).

$$\mathbf{P}(t) = \begin{cases} \mathbf{F} + \mathbf{L}.\mathbf{D}, & if r_{and} \geq 0.5 \\ \mathbf{F} - \mathbf{L}.\mathbf{D}, & if r_{and} < 0.5 \end{cases} \tag{16}$$

where $\mathbf{P}(t)$ is denoted as the updated location of tunicate with respect to the location of food source **F**.

Swarm behaviour. For mathematically simulating the swarm behaviour of tunicate, save the first two optimal best solutions and update the position of other search agents depending on the position of the best search agents. The swarm behaviour of tunicate is displayed in Eq. (17).

$$P(\mathbf{t}+1) = \frac{\mathbf{P}(t) + P(\mathbf{t}+1)}{2 + x_1} \tag{17}$$

where $P(\mathbf{t}+1)$ defines the swarm behaviour of tunicate.

The Pseudocode for tunicate swarm algorithm is given in Algorithm 2.

Algorithm 2 Pseudocode for tunicate swarm algorithm

```
Input: Number of population, training data, neural network, parameters $\mathbf{L},\mathbf{GF},\mathbf{W_f}and\mathbf{SF}$
Output: Optimal fitness, feature weights and feature selection
//Start
Initialize the tunicate population and parameters.
Calculate fitness of TSA using FFNN
Choose initial parameters and $Max_{iter}$.
Assign $P_{min} = 1, P_{max} = 4andSwarm = 0$;
while $(t < Max_{iter})$ do
    for i=1: size of position do
        Calculate fitness value of each search agent $(Ts)$
        // Jet propulsion behaviour
        $x_1, x_2, x_3 and r_{and} R_{and}()$
        Avoid conflicts between search agents.
        Find best position of search agent.
        // Swarm behaviour
        if $((r_{and} < 0.5))$ then
            Update the positions of other search agent
        else
            Select random search agent
        end if
    end for
    Adjust the updated search agent.
    Swarm → P(t)
    Compute the updated search agent fitness value using FFNN
    Update the parameters
    t=t+1
end while
return $T_S$
```

3.3 Bear smell search algorithm

Bear smell search algorithm is inspired from the dynamic behaviour of bears like smell sensing mechanism and the movement of bears in search of food for longer distance [44]. To predict the quality of an odour, a set of odorant components are considered. These odorant components mingle with each other which make the prediction process difficult. However, the prediction process can be easily done with the bear's sense of smell mechanism. The bears have excellent smelling sense since it has the largest olfactory bulb than other organisms. Thus, the olfactory bulb has a major part in this process [45]. During sense of smell mechanism, the odour is initially received by the olfactory bulb that transfers the information to the brain using olfactory tract. It has the simple structure among all the senses and thus considered for the algorithm. The main parts of this sense are glomerular, granular and the dissimilarity assessment parts.

Mathematical formulation of BSSA. In bear smell search algorithm, initially the bear's nose absorbs different odours from the environment. Each odour shows different locations for moving since everything in the environment has a special smell. Thus, these different odours are taken as the local solution and the odour of the preferred food is taken as the global solution. Consider $S_i = [sc_i^1 sc_i^2 \cdots sc_i^j \cdots sc_i^k]$ as the ith received odour with k molecules or components. The initial solution is taken as matrix, $SM = [S_i]_{n\times k} = [sc_i^j]_{n\times k}$ as the bear receives n odours while breathing. According to the breathing function and glomerular layer process in a sniff cycle, BS_i^j represents the jth odour component in the ith odour formulated as shown in Eq. (18).

$$BS_i^j = \begin{cases} G_i(t - t_{inhale}) + BS_i^{t_{inhale}}, & t_{inhale} \leq t \leq t_{exhale} \\ BS_i^{t_{exhale}} exp\frac{t_{exhale}-t}{\mu_{exhale}}, & t_{exhale} \leq t \end{cases} \tag{18}$$

where $_{exhale}$ denotes the constant value of exhalation time, t_{inhale} denotes the inhalation time and t_{exhale} denotes the exhalation time. The length of ith odour is denoted as k which is similar to the overall time cycle of breathing. Odour components are divided into two groups. Here, $G = G_1, G_2, \ldots, G_i, \ldots, G_n$ have the receptor sensitivities which identify and absorb the odour and have an input for ith mitral. The non-negative set G is represented in Eq. (19).

$$G_i(S_i) = \frac{1}{k}\sum_{j=1}^{k} f(sc_i^j), f(sc_i^j) = \begin{cases} 1, & T_1 \leq sc_i^j \\ 0, & T_2 > sc_i^j \end{cases} \tag{19}$$

where k is represented as the length of ith odour. T_1 and T_2 are represented as the threshold variables dependent on the average value of odour information. The information is transferred to the mitral and granular layers by employing Erdi and Li-Hopfield formulations [46]. It imitates the neural dynamics ascending from mitral and granular layers. To improve the exploitation during optimization process, the guided mechanism depending on the global solution is used.

Moreover, after transmitting all the information from neural activity to brain, the separating process is initiated with dissimilarity assessment which simulates the Pearson correlation. It assists the bear to choose the best path for the next location. The probability smell component (PSC) and probability smell fitness (PSF) is defined in Eqs. (20) and (21).

$$PSC_i = \frac{S_i}{max(S_i)} \tag{20}$$

$$PSF_i = \frac{SF_i}{max(SF_i)} \tag{21}$$

where the smell fitness is represented by SF. The dissimilarity among two odours is evaluated by distance smell components (DSC) and expected smell fitness (ESF) functions as represented in Eqs. (22) and (23).

$$DSC_i = 1 - \frac{\sum_{i=1}^{k} PSC_j^1 - PSC_j^2}{\sqrt{\sum_{j=1}^{k} \left(PSC_j^1 - PSC_j^2\right)^2}} \tag{22}$$

$$ESF_i = \left|PSF_i - PSF^g\right| \tag{23}$$

where g denotes the global solution. It gives the possible path of food for bear and also gives the relation between odours reached in the desired location. It demonstrates that the output of the brain decides the suitable path for the next location. Here, the distance between all odours is calculated using two thresholds $\vartheta_1 and\, \vartheta_2$. Thus, the next odours can be calculated from Eq. (24).

$$S_{k+1} = \begin{cases} Co_{1,i} \times S_k - rand \times Co_{2,i} \times (S_k - S_{best}), & DSC_i \leq \vartheta, ESF_i \leq \vartheta_2 \\ Co_{3,i} \times S_k - rand \times Co_{4,i} \times (S_k - S_{best}), & Otherwise \end{cases} \tag{24}$$

$$\begin{aligned} Co_{1,i} &= -\,ESF_i \frac{2 - DSC_i}{\vartheta_1}, Co_{2,i} = -ESF_i \frac{2 - DSC_i}{\vartheta_2}, \\ Co_{3,i} &= ESF_i \frac{2 - DSC_i}{\vartheta_1}, Co_{4,i} = ESF_i \frac{2 - DSC_i}{\vartheta_2} \end{aligned} \tag{25}$$

The Pseudocode for bear smell search algorithm is given in Algorithm 3.

Algorithm 3 Pseudocode for bear smell search algorithm

```
Input: Number of population, number of variables, training data, neural network, two
thresholds.
Output: Optimal fitness
//Start
Generate random population SM=p.d
Calculate fitness of BSSA (SM) using FFNN
Set as best smell as current position, $S_g$
while ($t < Max_{iter}$) do
    for i=size of bear smell do
        Calculate the max. Fitness value
        if ($min.ofbearfitness < Score$ then
            calculate best smell(i);
        end if
    end for
    for i=1← size of bear smell do
        for j=1← size of bear smell do
            Evaluate probability of smell component (PSC) and fitness (PSF)
            Define global solution ← min. of PSF.
            Calculate the (ESF) and (DSC) // Expected Smell Fitness
            // Distance smell Component
            // $(Co)_1 to (Co)_4$
            Calculate 4 co- efficient
        end for
    end for
    for i= 1:N // Number of population do
        if (($DSC < thr1, ESF < thr2$) then
            Generate best smell of $Co_1$
        else
            Generate best smell of $Co_3$
        end if
    end for
    t=t+1
end while
return  SM
```

3.4 Ant lion optimization algorithm

Antlion optimization algorithm is a meta-heuristic algorithm which mimics the hunting behaviour of antlions in nature [47]. The antlions create cone-shaped pit in the ground for trapping their prey (especially ants). After making a hole, the antlion waits under the bottom of the trap until the prey gets stuck in the trap. When the prey arrives into the trap, the antlion attempts to grab the prey by shooting the sand towards the edge of the pit which makes the prey to slide towards the bottom of the hole. After trapping the prey successfully, the antlion pulls the prey under the soil and consumes it. Then, it throws the leftovers of the prey outside the hole and waits for the next prey. The size of the pit is designed based on the hunger level of the antlion. If the hunger level of the antlion is high, then the pit size will be larger and vice-versa. This hunting behaviour of the antlion is mathematically modelled for solving optimization problems. Further, this algorithm utilises Roulette Wheel (RW) strategy for choosing the antlions depending on their fitness during optimization. It offers more possibilities for the elite antlions while hunting the prey.

Mathematical modelling of ALO. The ants move randomly during food searching process. The random walk of ants is denoted as Y^n and it is illustrated in Eq. (25).

$$\begin{aligned} Y^n =& [0, cumsum(2f(n_1) - 1), cumsum(2f(n_2) - 1)], \\ & \ldots, cumsum(2f(n_m) - 1)] \end{aligned} \tag{26}$$

where the cumulative sum is represented as cumsum, the step of random walk is denoted as n and the maximum number of iterations is given as m. The stochastic function is represented by f(n) and it is defined in Eq. (26).

$$f(n) = \begin{cases} 1 \ \ if \ \ rand > 0.5 \\ o \ \ if \ \ rand \le 0.5 \end{cases} \tag{27}$$

where rand is denoted as the random number formed by uniform distribution in the range of [0, 1]. In order to retain the random walk of ants within the search space, the location of each ant is normalized by min-max normalisation as given in Eq. (27).

$$Y_i^n = \frac{(Y_i^n - a_i)(b_i - c_i^n)}{d_i^n - a_i} + c_i^n \tag{28}$$

where d_i^n is denoted as the maximum of ith variable at iteration n, c_i^n is denoted as the minimum of ith variable at iteration n, b_i is denoted as the maximum random walk of ith variable and a_i is denoted as the minimum random walk of ith variable. The antlions can build traps proportional to their fitness and the ants should move stochastically. When the antlions realize that the ant is in the pit, they throw the sand outward from centre of the pit. This causes the ant to slide down the pit which is trying to escape. For mathematically modelling this process, the random walk radius of the ants is decreased as mentioned in Eqs. (28)–(30).

$$c^n = \frac{c^n}{I} \tag{29}$$

where I is denoted as the ratio and the minimum of all variables at nth iteration is denoted as c^n.

$$d^n = \frac{d^n}{I} \tag{30}$$

where the maximum of all variables at nth iteration is denoted as d^n and the ratio I is described in Eq. (30).

$$I = 10^{\omega} \frac{n}{N} \tag{31}$$

where the maximum number of iterations is represented as N and ω is represented as the constant which depends on the current iteration. ($\omega = 2$ when $n > 0.1N, \omega = 3$ when $n > 0.5N, \omega = 4$ when $n > 0.75N, \omega = 5$ when $n > 0.9N, and\omega = 6$ when $n > 0.95N$). Also, the accuracy level of exploitation can be adjusted using the constant ω.

3.5 Modified ant lion optimization

Antlion optimization algorithm is the novel meta-heuristic global search algorithm inspired from the hunting behavior of antlions [47]. It does not trap in the local optimum easily since its performance is not dependent on any parameters. It utilises roulette wheel and random walk strategy for exploration which generates diverse solutions. Likewise, the adaptive shrinking boundaries of antlion pits ensure the exploitation of search space. Thus, ALO algorithm is successfully used in many engineering applications due to its good exploration and exploitation abilities. For effective feature selection process, the modified ALO is used which is based on ALO and Lévy Flight (LF) distribution. The LF distribution effectively searches the optimal ants because of its strong exploration ability [48].

LF distribution with random walk. Lévy Flight distribution is the random walk strategy used for optimizing the searching efficiency of an algorithm. It is a specific division of generalised random walk wherein the step lengths during the walk are demonstrated by a heavy tailed probability distribution [49]. LF distribution is extensively used for solving complex optimization problems in the field of evolutionary computation due to its dynamic random walk properties [50, 51]. Assume Y_i as the ant's position transferred to the new position by LF distribution as stated in Eq. (31).

$$LY_i = Y_i + \alpha \oplus levy(\lambda) \tag{32}$$

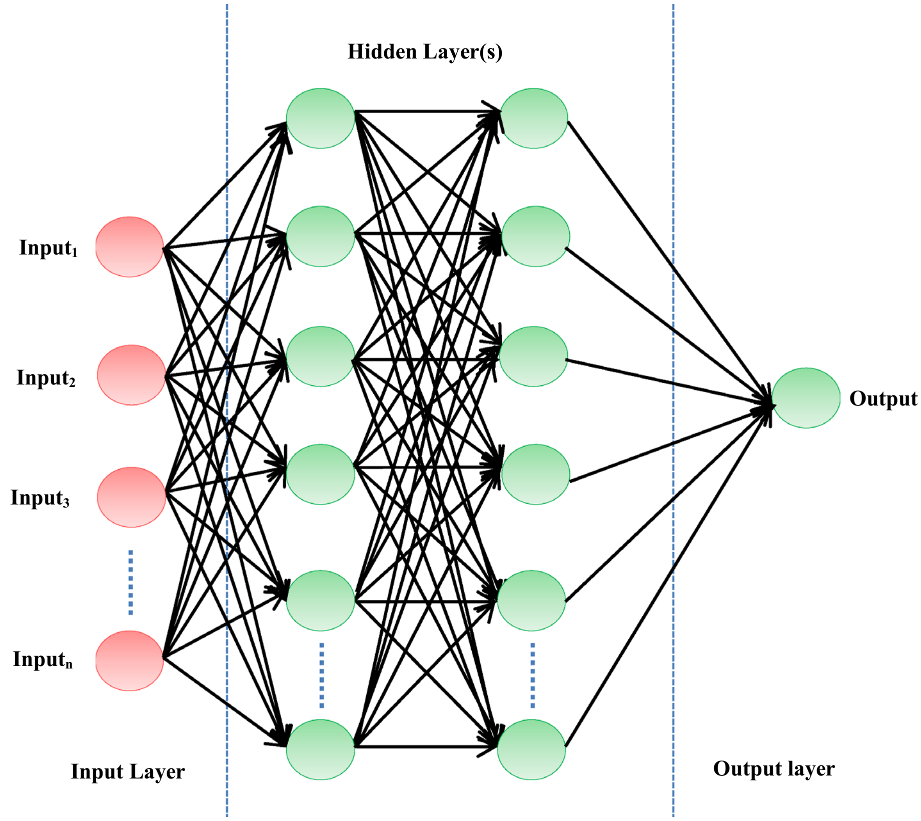

Fig. 2 Feed forward neural network architecture

where λ is denoted as the step size and LY_i represents the new position of the ant.

Elitism with crossover operation. In meta-heuristic algorithms, elitism is an important feature because it helps to sustain the optimal solutions acquired at any phase of the optimization process. It is not adapted to the binary coding form and depends on the addition operation. Therefore, crossover operation is used to obtain more than one parent solution and an offspring solution from the entire population. It is a process among two binary solutions acquired from random walk [52]. During all iterations, the fittest antlion is considered as the elite which affect the movement of all ants throughout the iterations. Assume that the ants walk nearby the elite antlion based on the roulette wheel strategy as shown in Eq. (33).

$$(Ant)_i^n = Crossover\left(RW_A^n, RW_E^n\right) \tag{33}$$

Equation (33) gives the elitism with crossover operation occurring in the search space. Here, RW_E^n is denoted as the random walk near the elite at iteration n and RW_A^n is denoted as the random walk near the antlion chosen by the roulette wheel. Hence, the modified ALO algorithm uses LF distribution and crossover operation for solving feature selection problem and obtaining optimal solution for classification process. Algorithm 4 gives the pseudocode for modified antlion optimization algorithm.

Table 1 Dataset description

Dataset	No. of instances	No. of attributes
Breast cancer wisconsin (diagnostic)	569	32
Pima Indians diabetes	846	8
Heart disease	303	75
Liver disorders	345	7
Parkinson's disease	197	23

Table 2 Parameter settings of ANN

Parameters	Values
Size of input layers	Size of data
Learning rate	0.01, 0.1, 0.5
Type of ANN	Feed forward
Number of training algorithm	9
Number of neurons in each middle layer	20–100
K-fold cross validation	10

Algorithm 4 Pseudocode for modified antlion optimization algorithm

```
Input: training data, neural network, number of populations and iteration number, ($n = 1$)
Output: optimal feature weighting and selection, fitness value ($Ma$)
//Start
Initialize MALO population P,
Compute fitness value of ant lion using FFNN
while ($t < Max_{iter}$) do
    for for i=1 to P do
        Select two ant lion's pits of c and d
        // minimum and maximum $i^{th}$ variables.
        Create normalization random walk around two selected ant lions
        Decide the new position of $i^{th}$ antlion.
        Change the population to the binary code.
    end for
    Calculate the fitness of ants using FFNN
    if (ant best < antlion then
        Replace the position of antlion
    else
        Update its position to the latest position of ant
    end if
    t=t+1
end while
return  Ma
```

Table 3 Parameter settings of evolutionary algorithms

Algorithm	Parameters	Values
ChOA	Maximum number of iterations	1000
	Search agent	Number of records in dataset
	m	Chaotic
	Number of chimp position	4
BSSA	Maximum number of iterations	1000
	Number of search agents	Number of records in dataset
	Number of population	100
	Threshold1	1
	Threshold2	50
	Dimensions	Number of instances in dataset
	Number of variables	10
	Co-efficient	1-4
TSA	Maximum number of iterations	1000
	Search agent	Number of records in dataset
	Number of variables	4
	P_{min} & P_{max}	1 & 4
	Dimensions	Number of instances in dataset
MALO	Maximum number of iterations	1000
	Search agent	Number of records in dataset
	Number of antlions and ants	20
	Number of Population	10
	Dimensions	Number of instances in dataset
ALO	Feature weights	[0,1]
	Maximum number of iterations	1000
	Search agent	Number of records in dataset
	Populations	20
	Dimensions	Number of instances in dataset

4 Artificial neural network

Artificial Neural Networks (ANNs) are widely used machine learning algorithms that are applied in several approaches due to their high classification performance. It consists of neurons capable of extracting information from the dataset even in noisy data [53]. They are universal function approximation algorithms for modelling both linear and non-linear data with required accuracy. Various kinds of neural networks are modelled using different interconnection approaches in the data mining field. The commonly used neural network model is the feed forward neural network, also known as multilayer perceptron. FFNN consists of three types of units such as input unit, hidden unit and output unit. These units have processing nodes fully connected with one another and they do not have any interconnections among the nodes within the same layer. In this paper, FFNN is adopted for classification and to test the accuracy of features selected through above optimization techniques. The architecture of feed forward neural network is given in Fig. 2.

The network is represented by directed graphs, the units are represented by nodes and the connections among them are represented by arcs. Each arc has a value which is the connection weight among a pair of units [54]. In FFNN model, all connections from the input unit of the network are directed towards the hidden unit and finally to the output unit. Each unit i carries out a function as expressed in Eq. (32).

$$u_i = f_i\left(\sum_{j=1}^{n} s_{ij}v_j - \vartheta_i\right) \tag{34}$$

where $_i$ is denoted as the threshold of unit i, v represents the jth input of the unit, s_{ij} is represented as the connection weight among units i and j, u_i represents the output of unit i and f_i is denoted as the activation function of unit i. The fitness function is the classification accuracy, and it is given in Eq. (33).

Table 5 Results for different normalization techniques

Normalization techniques	Dataset				
	Breast	Pima	Heart	Liver	Parkinson
Accuracy					
Min–max	92.20	79.30	85.14	71.01	67.51
z-score	75.80	81.25	86.15	75.07	76.65
tanh	93.81	91.96	94.72	96.10	91.56
Sensitivity					
Min–max	0.7813	0.8800	0.8214	0.5068	0.7207
z-score	0.7396	0.8940	0.8567	0.6233	0.6977
tanh	0.9158	0.9152	0.9162	0.9518	0.9044
Specificity					
Min–max	0.8459	0.8800	0.8347	0.8593	0.6163
z-score	0.8275	0.6604	0.8571	0.8442	0.8198
tanh	0.9657	0.9321	0.9853	0.9865	0.9500

$$fitness\ function = \frac{Correctly\ classified\ instances}{Total\ instances} \tag{35}$$

5 Experimental evaluation

5.1 Experimental setup

5.1.1 Dataset description

The presented algorithms are used for optimizing the classifier weights in the dataset having large number of features. The experimental process is performed and compared on five benchmark datasets such as Heart Disease, Pima Indians Diabetes, Breast Cancer Wisconsin (Diagnostic), Liver

Table 4 Ten-fold cross validated Pima Indian diabetes dataset

Dataset	Data files	Number of records	Task	Class-1	Class-2
Pima Indians diabetes	10-1train.dat	700	Training	560	140
	10-1test.dat	146	Testing	96	50
	10-2train.dat	700	Training	560	140
	10-2test.dat	146	Testing	96	50
	10-3train.dat	700	Training	560	140
	10-3test.dat	146	Testing	96	50
	.	.	.	.	.
	.	.	.	.	.
	.	.	.	.	.
	10-10 train.dat	700	Training	560	140
	10-10 test.dat	146	Testing	96	50

Table 6 Comparison of feature selection and feature weighting algorithms

Datasets	Parameter	ChOA	BBSA	TSA	MALO	ALO
BREAST	Best	− 0.26656	− 0.96782	0.012857	0.12761	0.35025
	Worst	0.47357	0.53945	4.4278	14.4164	15.3721
	Mean	0.053976	0.091341	0.03455	0.1354	0.79372
	Std. Dev	317.3516	127.517	1.0079	3.2107	3.925
PIMA	Best	− 0.0036805	− 0.046096	0.0093316	0.039009	− 0.053572
	Worst	0.012862	0.12713	0.098557	8.1947	2.7348
	Mean	0.00051892	0.042731	0.012069	0.64479	0.14449
	Std. Dev	51.774	0.081863	0.062409	27.7464	0.24612
HEART	Best	− 10.5364	− 10.156	−10.5364	− 5.1756	− 0.94599
	Worst	4.0007	4.0348	4.0007	7.9995	1.0306
	Mean	4.0001	3.9988	4.0001	7.9995	0.76013
	Std. Dev	4.0813	0.036118	0.0006316	1.746e-05	0.10468
LIVER	Best	− 0.51279	− 0.43107	− 0.50596	− 0.31598	− 0.21926
	Worst	0.09067	2.3041	0.1419	2.8357	3.5578
	Mean	− 0.39732	0.045401	− 0.41792	0.009813	0.020746
	Std. Dev	0.054382	6113.1743	0.069686	0.2505	28233.1247
Parkinson	Best	3.3569e-07	0.006236	5.9043e-17	0.040729	0.66698
	Worst	2.4033e–06	0.12523	2.723e–15	0.51223	80.03
	Mean	1.0353e–05	0.47733	2.5583e–14	4.0341	10
	Std. Dev	147.8541	0.03889	1.0418e–15	155.0050	5.2083

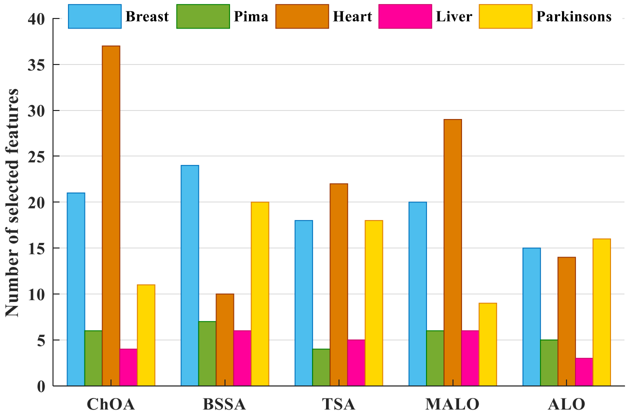

Fig. 3 Number of selected features in each meta-heuristic algorithm

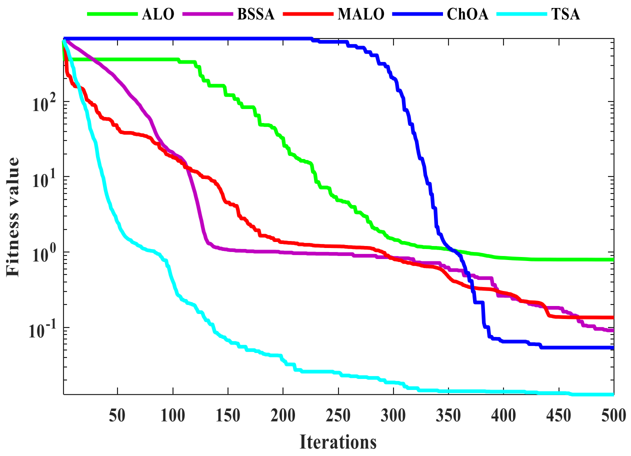

Fig. 4 Convergence curve for breast cancer dataset

Disorders and Parkinson's disease to demonstrate the performance of the metaheuristic algorithms. These datasets are taken from the UCI machine learning repository [55]. The performance analysis is carried out among five evolutionary algorithms such as Chimp Optimization Algorithm (ChOA), Ant Lion Optimization (ALO) algorithm, Modified Ant Lion Optimization (MALO) algorithm, Tunicate Swarm Algorithm (TSA) and Bear Smell Search Algorithm (BSSA) for feature selection and feature weighting. The proposed technique is implemented on Matlab R2018a software running on a windows operating system. The dataset description is displayed in Table 1.

5.1.2 Parameter settings

The parameter settings for feed forward neural network are illustrated in Table 2. For validating the datasets, ten-fold cross validation is conducted. Here, the dataset is divided into 10 folds in which 9 folds are used for training and 1 fold is used for testing.

The parameter settings of five evolutionary algorithms such as ChOA, BSSA, ALO, MALO and TSA for feature selection and feature weighting during experimentation are

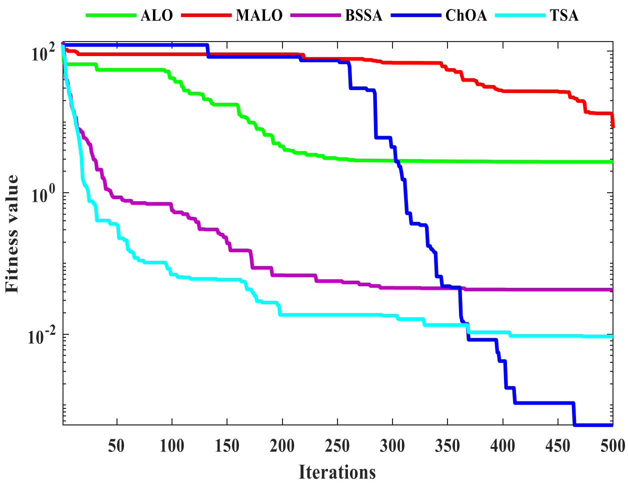

Fig. 5 Convergence curve for Pima diabetes dataset

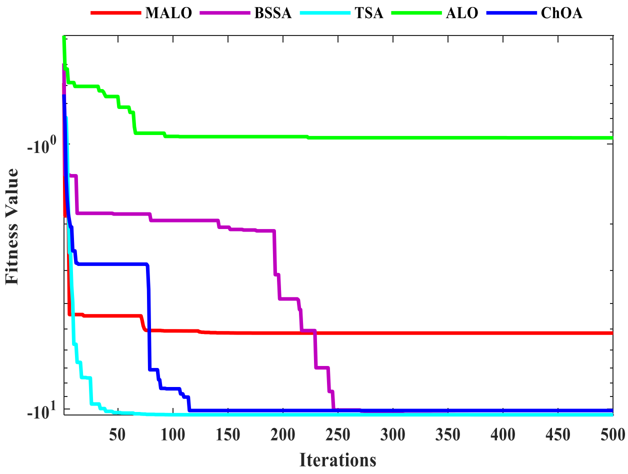

Fig. 6 Convergence curve for heart disease dataset

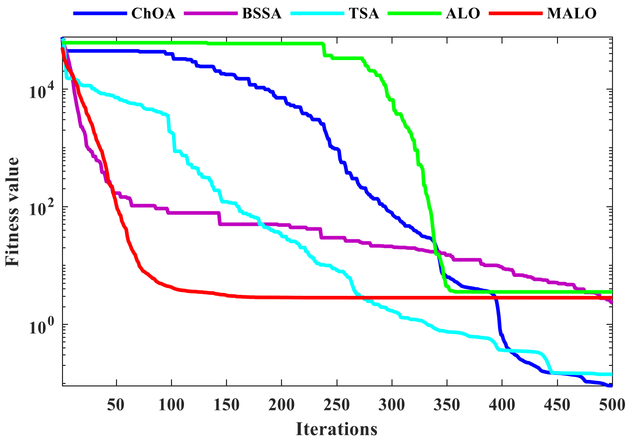

Fig. 7 Convergence curve for liver disorders dataset

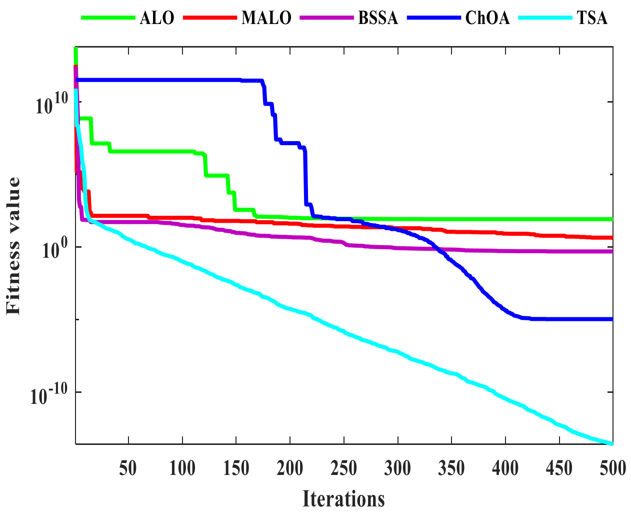

Fig. 8 Convergence curve for Parkinson dataset

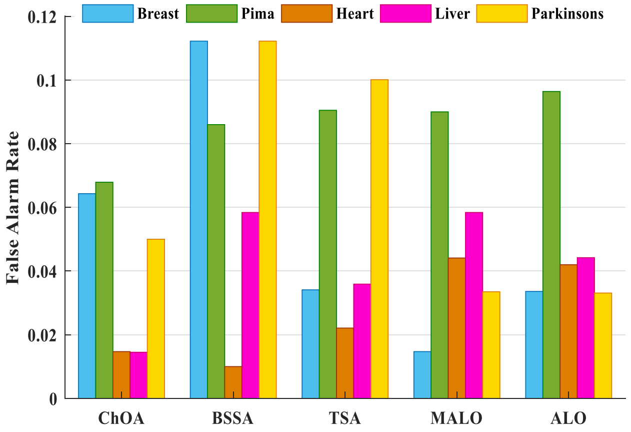

Fig. 9 False Alarm rate in each meta-heuristic algorithm

Table 7 Classification without feature selection and feature weighting

Parameters	Dataset				
	Breast	Pima	Heart	Liver	Parkinson
Accuracy	62.21	78.26	85.47	75.94	71.57
Sensitivity	0.6644	0.9400	0.2174	0.6138	0.8198
Specificity	0.5766	0.4888	0.9720	0.8650	0.5814
Training time	22.1852	8.19	146.8249	12.6562	12.3091
Predicted time	0.00033	0.000119	0.08642	0.000374	0.0003471

illustrated in Table 3. For searching through the large solution space, larger iterations are required to avoid stagnation. Thus, the maximum number of iterations is set as 1000 for all meta-heuristic algorithms.

Table 8 Classification with feature selection and feature weighting

Algorithm	Parameters	Dataset				
		Breast	Pima	Heart	Liver	Parkinson
ChOA	Accuracy	93.81	91.96	94.72	96.10	91.56
	Sensitivity	0.9158	0.9152	0.9162	0.9518	0.9044
	Specificity	0.9657	0.9321	0.9853	0.9865	0.9500
BSSA	Accuracy	87.82	92.91	95.27	91.30	90.79
	Sensitivity	0.8687	0.9344	0.9584	0.8942	0.9283
	Specificity	0.8878	0.9140	0.9367	0.9416	0.9178
TSA	Accuracy	94.80	90.66	94.39	95.98	97.89
	Sensitivity	0.9422	0.9056	0.9162	0.9583	0.9763
	Specificity	0.9641	0.9095	0.9779	0.9641	0.9842
MALO	Accuracy	95.05	89.36	93.40	89.28	93.03
	Sensitivity	0.9222	0.8880	0.9162	0.8606	0.9181
	Specificity	0.9853	0.9095	0.9559	0.9416	0.9641
ALO	Accuracy	92.55	93.62	93.06	86.38	91.61
	Sensitivity	0.9117	0.9327	0.9281	0.8125	0.8989
	Specificity	0.9641	0.9416	0.9559	0.9416	0.9641

5.2 Experimental results and discussion

Most of the research works in meta-heuristic optimization algorithm address the problem of improving long execution time and classification accuracy. A lot of meta-heuristic optimization approaches have been proposed to improve the classification accuracy using feature selection methods. The cross validation concept is used for computing the generalised unbiased performance of the proposed method [56]. The evaluation process is performed on five different datasets where tenfold cross validation is employed for evaluation. Ten individual runs are carried out for every single fold. It compares the learning algorithms by splitting the dataset into training and testing set where the training set is used for training the model and the testing dataset is used for evaluating the model. The Pima diabetes dataset is cross validated by utilising ten-fold cross validation and the other datasets are also prepared in similar way. For simplicity, only the ten-fold cross validated Pima diabetes dataset has been shown in Table 4.

Table 5 gives the data normalization values for different normalization techniques. The normalization techniques utilised are min-max normalization, z-score normalization and tanh normalization. The table shows that the tanh normalization method has gained better accuracy, sensitivity and specificity for all optimization algorithms in the given datasets.

Different evolutionary optimization algorithms are presented in this research for feature selection and feature weighting during classification. The main aim of this research is employing feature selection and feature weighting with five recently developed meta-heuristic algorithms to get precise classification for the input datasets. It is necessary

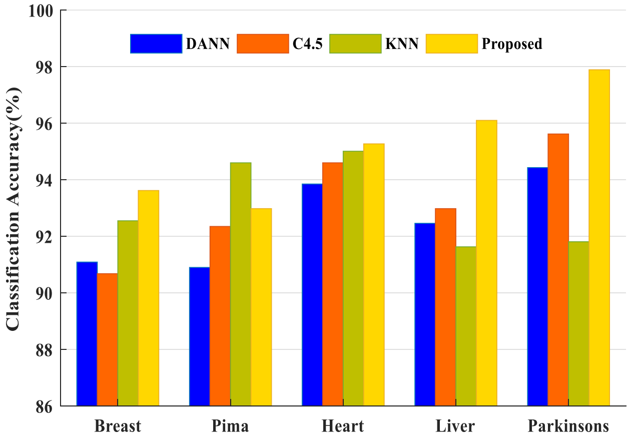

Fig. 10 Classification accuracy for each dataset

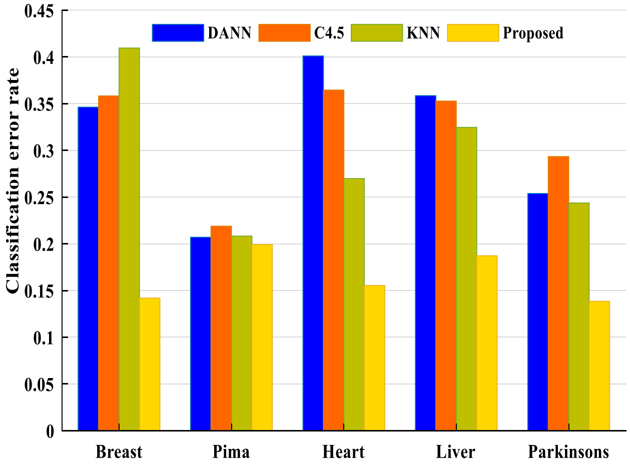

Fig. 11 Classification error rate for each dataset

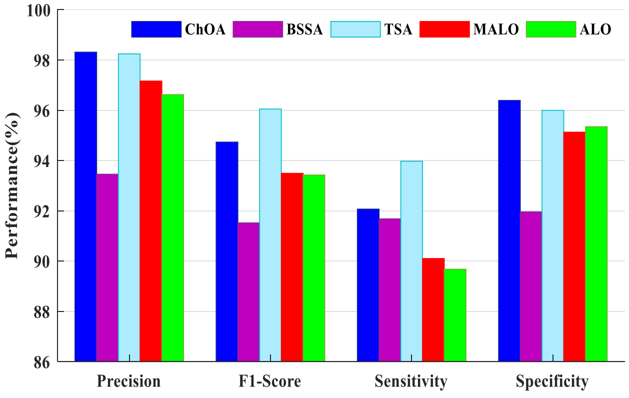

Fig. 12 Performance of different metaheuristic algorithms

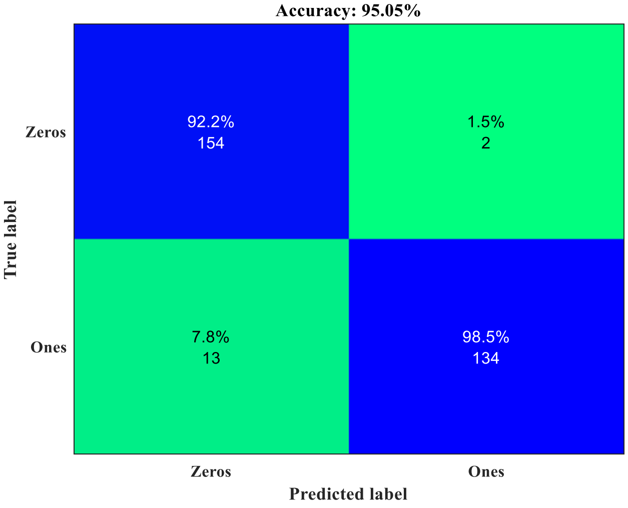

Fig. 13 Confusion matrix for breast cancer dataset

to distinguish the performance of each algorithm to find its merits as well as demerits. The optimal solutions obtained for the five optimization algorithms are evaluated with different metrics like standard deviation, mean, best solution and worst solution. Table 6 gives the comparison among the given feature selection and feature weighting algorithms for five different datasets.

In case of all features, only minimum and maximum values of best training algorithms are presented over the 10 independent runs. Minimum value is presented as best value and maximum as the worst. Also the average value obtained for 10 runs is presented as mean value. Table 6 shows that the ChOA has gained high standard deviation value for all datasets except liver dataset. For liver dataset ALO algorithm has obtained the highest standard deviation value.

The number of selected features for different dataset by the meta-heuristic algorithms is given in Fig. 3. It can be observed that ALO has selected a least number of features for breast cancer and liver datasets. Also, TSA has selected a least number of features for Pima dataset than other algorithms. For heart dataset, BSSA has selected a least number of features. For Parkinson dataset, MALO has selected a least number of features than other algorithms. Hence, it shows that all the evolutionary algorithms have better feature selection ability for all the datasets.

The convergence curve comparison of different techniques for all the input datasets is shown in Figs. 4, 5, 6, 7 and 8. It can be seen that, TSA and ChOA has obtained good results for almost all datasets. On the other hand, BBSA, ALO and MALO have also achieved best outcomes but the accuracy is relatively lower than other algorithms. From the convergence analysis, it is illustrated that the TSA has gained the best convergence for datasets such as breast cancer and Parkinson dataset. For heart dataset, both TSA and ChOA have gained the best convergence. Likewise, for the other two datasets like Pima diabetes and liver disorders dataset, ChOA has obtained the best convergence. The results show that the TSA and ChOA have gained better performance than other algorithms.

Figure 9 gives the False Alarm Rate (FAR) for the meta-heuristic algorithms with five different datasets. Lesser the false alarm rate, higher the efficiency of the algorithm. It is observed that the ChOA has obtained a lesser FAR for two datasets namely Pima and liver dataset. For breast cancer dataset, the MALO has obtained a lesser FAR value. For heart dataset, the BSSA has obtained a lesser FAR value and for Parkinson dataset, ALO has obtained the lesser value. It shows that the ChOA has relatively less FAR rate compared to other evolutionary optimization algorithms.

The classification results without feature selection and feature weighting are given in Table 7. The evaluation process is carried out on five different datasets with parameters

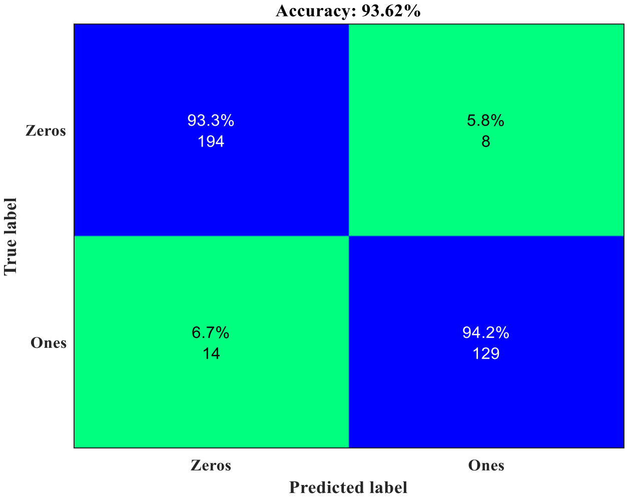

Fig. 14 Confusion matrix for Pima diabetes dataset

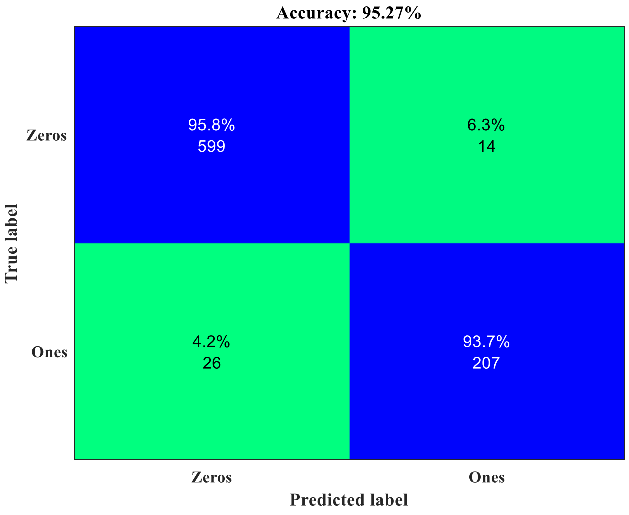

Fig. 15 Confusion matrix for heart disease dataset

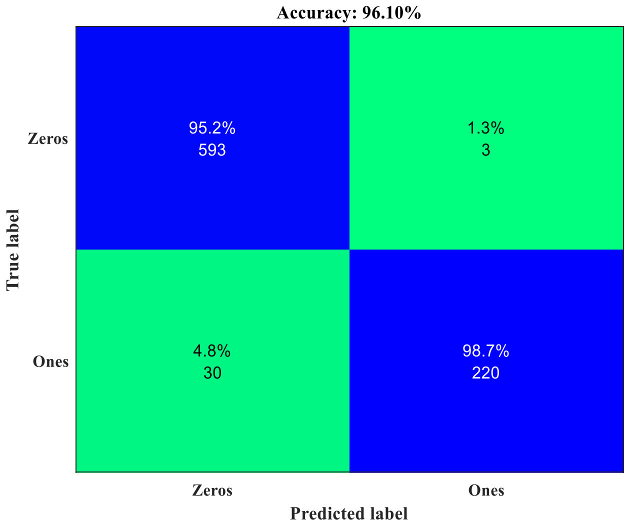

Fig. 16 Confusion matrix for liver dataset

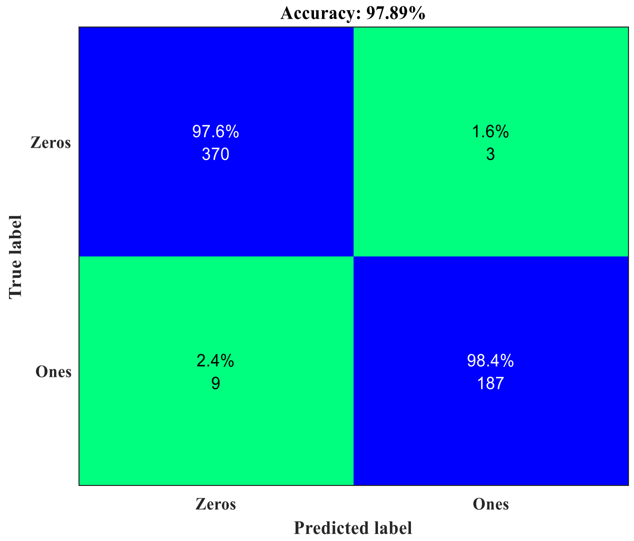

Fig. 17 Confusion matrix for Parkinson dataset

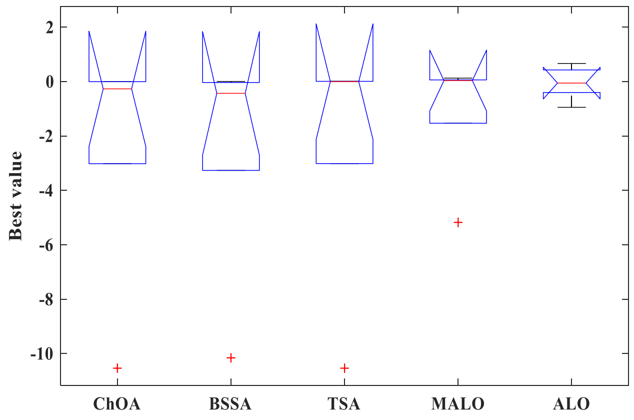

Fig. 18 ANOVA test results for the metaheuristic algorithms

Table 9 ANOVA test results

Source	SS	df	MS	F	ProbF
Columns	20.237	4	5.0593	0.37	0.8266
Error	272.915	20	13.6458		
Total	293.152		24		

like specificity, accuracy, sensitivity and classification time. The heart dataset has obtained higher accuracy and specificity but it require huge time for training than the other datasets while the Pima dataset has gained high specificity and minimum training and prediction time than the other datasets.

The classification results with feature selection and feature weighting using meta-heuristic algorithms are given in Table 8. It is observed that TSA has obtained highest classification accuracy and sensitivity for feature selection and weighting when compared to other evolutionary algorithms. The accuracy of classification with feature selection and weighting is higher for all datasets when compared to the accuracy of classification without feature selection and weighting. This shows that the feature selection and weighting increases the classification accuracy of neural networks. From the experimental results, it is illustrated that the TSA and ChOA have shown comparatively better results than other evolutionary algorithms.

For evaluating the performance of FFNN with feature selection and weighting, it is compared with existing models like discriminant adaptive nearest neighbour (DANN) [57], C4.5 [58] and K-nearest algorithm [59]. For each dataset, the meta-heuristic algorithm with highest classification accuracy

Table 10 Training and prediction time for feature selection and feature weighting algorithms

Algorithm	Parameters	Dataset				
		Breast	Pima	Heart	Liver	Parkinson
ChOA	Training time	21.001	5.501	129.321	13.564	20.147
	Predicted time	0.0012	0.00014	0.03741	0.00301	0.0011
BSSA	Training time	5.647	10.253	21.128	8.3621	12.2001
	Predicted time	0.0024	0.00032	0.0009	0.000214	0.00134
TSA	Training time	54.235	14.95870	116.254	11.154	20.3654
	Predicted time	0.0068	0.00321	0.00023	0.0316	0.0012
MALO	Training time	113.9541	21.9581	163.4711	13.149	15.487
	Predicted time	0.000142	0.0101	0.02781	0.00121	0.00254
ALO	Training time	96.859	9.2511	234.1425	26.9514	52.142
	Predicted time	0.0021	0.00023	0.01012	0.00029	0.0032

is considered as the accuracy of FFNN. For breast dataset, MALO algorithm has gained the highest accuracy. For Pima dataset, ALO algorithm has gained the highest classification accuracy. For heart dataset, BSSA has obtained the highest classification accuracy. For liver dataset, ChOA has gained the highest classification accuracy and for Parkinson dataset, TSA has gained the highest classification accuracy. Figure 10 gives the classification accuracy of FFNN and other existing classification algorithms for five different datasets. It illustrates that the FFNN has obtained higher classification accuracy for all datasets except Pima diabetes dataset. For Pima dataset, KNN has gained highest classification accuracy.

Figure 11 gives the classification error rate for FFNN and existing algorithms for five different datasets. The feed forward neural network has obtained lesser error rate for all the datasets than existing classification models. The lesser error rate implies that the FFNN is more effective for classification than other comparative algorithms. On the other hand, the DANN algorithm has obtained maximum error rate for heart and liver datasets which implies that it is less efficient for classification.

Figure 12 gives the performance of algorithms for feature selection and feature weighting. The performance of algorithms is evaluated based on precision, F-measure, sensitivity and specificity. These parametric values are taken as the mean of all datasets for each algorithm. The figure shows that ChOA has obtained high precision and specificity than other algorithms. Also, TSA has obtained high F1-Score and sensitivity than the other metaheuristic algorithms. on the other hand, BSSA has obtained least value for all performance measures expect sensitivity. Hence, It can be observed that ChOA and TSA have shown better performance than other existing algorithms.

The performance of feed forward neural network for the selected features is estimated on the basis of classification accuracy for the given datasets. The confusion matrix is a resultant from the testing process used for analysing the performance of FFNN with feature selection and feature weighting using five evolutionary algorithms. For each dataset, the algorithms with highest classification accuracy are displayed for the confusion matrix. For breast dataset, MALO algorithm has gained the highest accuracy. For Pima dataset, ALO algorithm has gained the highest classification accuracy. For heart dataset, BSSA has obtained the highest classification accuracy. For liver dataset, ChOA has gained the highest classification accuracy and for Parkinson dataset, TSA has gained the highest classification accuracy. The confusion matrix in Figs. 13, 14, 15, 16 and 17 shows the classification performance of feed forward neural network for five different datasets. From the experimental results, it is concluded that the performance of FFNN with feature selection and feature weighting using five evolutionary algorithms is appropriate in all datasets.

5.3 Statistical analysis

The statistical analysis is carried out to check whether the new algorithms performed better than the current algorithms. It examines issues like classification error rates, classification accuracy, and so on. For, statistical analysis, a variety of tests including the Post Hoc test, Dunnett test, Tukey test, Friedman test, and ANOVA test are conducted. A one-way analysis of variance (ANOVA) test is used to determine whether there is a statistical difference between the suggested algorithm and other comparable algorithms. The statistical correctness of the metaheuristic algorithms is compared to the statistical correctness of other existing algorithms. It consists of mean and variance, which are used to calculate the test statistics. The test statistic is then used to determine if the data in the group is the same or different. The box plots for ANOVA test is illustrated in Fig. 18. The ANOVA test is carried out for five metaheuristic algorithms.

To examine the significant difference between algorithms, a parametric statistical test known as the analysis of variance (ANOVA) is performed using a 5% significance level. The selected algorithms served as the control group in the trials. They were compared to the other metaheuristic algorithms in terms of the mean value of performance metrics. Table 9 gives the ANOVA test results.

5.4 Computational complexity analysis

The most appropriate way to assess the complexity of an algorithm is to look at how long it takes to run. A complexity metric can also be based on space, i.e. memory, though time is usually the more meaningful metric. Table 10 gives the training and prediction time for feature selection and feature weighting algorithms. It shows that BSSA has consumed lesser time for training process than other meta-heuristic algorithms for all datasets except Pima. Also, the heart dataset has consumed more time for training process than other datasets for all algorithms. On the other hand, Liver dataset has consumed lesser time for training process than other datasets with and without feature selection and feature weighting for all evolutionary algorithms except ChOA. In ChOA, Pima diabetes dataset has consumed lesser training time with and without feature selection and feature weighting. As for testing time, the MALO has consumed minimum prediction time than other evolutionary algorithms.

6 Conclusion

In this paper, five meta-heuristic algorithms are compared for feature selection and feature weighting process. The algorithms employed are ChOA, TSA, BSSA, ALO and MALO. Five UCI datasets are used for performance analysis of the meta-heuristic optimization algorithms and the comparative results are obtained. From the experimental results, it has been observed that the accuracy of FFNN is greatly improved and the computation time is comparatively reduced after feature selection and feature weighting using meta-heuristic algorithms. For evaluating the classification performance, the classification results are compared with other existing classifiers like DANN, C4.5 and KNN classifier. The experimental results display that the FFNN with feature selection and feature weighting has obtained higher classification accuracy than existing algorithms on different datasets. Thus, the meta-heuristic optimization algorithms used for comparison are found to be effective for feature selection and feature weighting processes. Moreover, neural network classification is stated as a promising model with better classification rates. The algorithms used in the comparative analysis are stable enough to find relevant feature required for hard classification problems. It would be noteworthy to consider hybrid optimization algorithm for feature selection and feature weighting in data mining classification tasks in the future research.

References

1. Dey A (2016) Machine learning algorithms: a review. Int J Comput Sci Inf Technol 7(3):1174–1179
2. Xue B, Cervante L, Shang L, Browne WN, Zhang M (2013) Multi-objective evolutionary algorithms for filter based feature selection in classification. Int J Artif Intell Tools 22(04):1350024
3. Hughes G (1968) On the mean accuracy of statistical pattern recognizers. IEEE Trans Inf Theory 14(1):55–63
4. Tahir MA, Bouridane A, Kurugollu F (2007) Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier. Pattern Recogn Lett 28(4):438–446
5. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
6. Galeshchuk S (2016) Neural networks performance in exchange rate prediction. Neurocomputing 172:446–452
7. Kavzoglu T, Mather PM (2000) The use of feature selection techniques in the context of artificial neural networks. In: Proceedings of the 26th annual conference of the remote sensing society
8. Perez-Rodriguez J, Arroyo-Pena AG, Garcia-Pedrajas N (2015) Simultaneous instance and feature selection and weighting using evolutionary computation: Proposal and study. Appl Soft Comput 37:416–443
9. Zhou Y, Cheng G, Jiang S, Dai M (2020) Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput Networks 174:107247
10. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2000) Feature selection for SVMs. Adv Neural Inf Process Syst 13:668–674
11. Peng HC, Ding C, Long FH (2005) Minimum redundancy-maximum relevance feature selection 70–71
12. Kelly JD Jr, Davis L (1991) A hybrid genetic algorithm for classification. IJCAI 91:645–650
13. Dialameh M, Jahromi MZ (2017) A general feature-weighting function for classification problems. Expert Syst Appl 72:177–188
14. Wettschereck D, Aha DW, Mohri T (1997) A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artif Intell Rev 11(1–5):273–314
15. Raymer ML, Punch WF, Goodman ED, Kuhn LA, Jain AK (2000) Dimensionality reduction using genetic algorithms. IEEE Trans Evol Comput 4(2):164–171
16. Singh D, Singh B (2019) Hybridization of feature selection and feature weighting for high dimensional data. Appl Intell 49(4):1580–1596
17. Alba E, Dorronsoro B (2005) The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans Evol Comput 9(2):126–142
18. Lozano M, García-Martínez C (2010) Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report. Comput Oper Res 37(3):481–497
19. Shayeghi H, Ghasemi A, Moradzadeh M, Nooshyar M (2017) Day-ahead electricity price forecasting using WPT, GMI and modified LSSVM-based S-OLABC algorithm. Soft Comput 21(2):525–541
20. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
21. Song XF, Zhang Y, Gong DW, Gao XZ (2021) A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. IEEE Trans Cybern 99:1–14
22. Song XF, Zhang Y, Gong DW, Sun XY (2021) Feature selection using bare-bones particle swarm optimization with mutual information. Pattern Recogn 112:107804

23. Song XF, Zhang Y, Guo YN, Sun XY, Wang YL (2020) Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. IEEE Trans Evol Comput 24(5):882–895
24. Hu Y, Zhang Y, Gong D (2020) Multiobjective particle swarm optimization for feature selection with fuzzy cost. IEEE Trans Cybern 51(2):874–888
25. Darwish A (2018) Bio-inspired computing: algorithms review, deep analysis, and the scope of applications. Future Comput Inf J 3(2):231–246
26. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks IEEE, vol 4. pp 1942–1948
27. Yazdani M, Jolai F (2016) Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J Comput Des Eng 3(1):24–36
28. Cuevas E, Cienfuegos M, Rojas R, Padilla A (2015) A computational intelligence optimization algorithm based on the behavior of the social-spider. In: Computational intelligence applications in modeling and control. Springer, pp. 123–146
29. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
30. Song X, Tang L, Zhao S, Zhang X, Li L, Huang J, Cai W (2015) Grey wolf optimizer for parameter estimation in surface waves. Soil Dyn Earthq Eng 75:147–157
31. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381
32. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization, Technical Report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department, 200, pp 1–10
33. Parpinelli RS, Lopes HS, Freitas AA (2002) Data mining with an ant colony optimization algorithm. IEEE Trans Evol Comput 6(4):321–332
34. Michelakos I, Mallios N, Papageorgiou E, Vassilakopoulos M (2011) Ant colony optimization and data mining. In: Next Generation Data Technologies for Collective Computational Intelligence. Springer, pp 31–60
35. Singh D, Singh B (2020) Investigating the impact of data normalization on classification performance. Appl Soft Comput 97:105524
36. Tharwat A, Hassanien AE (2018) Chaotic antlion algorithm for parameter optimization of support vector machine. Appl Intell 48(3):670–686
37. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. Expert Syst Appl 149:113338
38. Couzin ID, Laidre ME (2009) Fission-fusion populations. Curr Biol 19(15):R633–R635
39. Boesch C (2002) Cooperative hunting roles among Tai chimpanzees. Hum Nat 13(1):27–46
40. Stanford CB, Wallis J, Mpongo E, Goodall J (1994) Hunting decisions in wild chimpanzees. Behaviour 131(1–2):1–18
41. Mosavi MR, Khishe M, Akbarisani M (2017) Neural network trained by biogeography-based optimizer with chaos for sonar data set classification. Wirel Pers Commun 95(4):4623–4642
42. Berrill JJ (1950) The Tuniccafa. The Royal Society, London
43. Davenport J, Balazs GH (1991) Fiery bodies—are pyrosomas an important component of the diet of leatherback turtles? Biology 37:33–38
44. Ghasemi-Marzbali A (2020) A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm. Soft Comput 24(2):1–33
45. Jordan R, Fukunaga I, Kollo M, Schaefer AT (2018) Active sampling state dynamically enhances olfactory bulb odor representation. Neuron 98(6):1214–1228
46. Li Z, Hopfield JJ (1989) Modeling the olfactory bulb and its neural oscillatory processings. Biol Cybern 61(5):379–392
47. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
48. Wang M, Wu C, Wang L, Xiang D, Huang X (2019) A feature selection approach for hyperspectral image based on modified ant lion optimizer. Knowl-Based Syst 168:39–48
49. Barthelemy P, Bertolotti J, Wiersma DS (2008) A Lévy flight for light. Nature 453(7194):495–498
50. Haklı H, Uğuz H (2014) A novel particle swarm optimization algorithm with Levy flight. Appl Soft Comput 23:333–345
51. Heidari AA, Pahlavani P (2017) An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. Appl Soft Comput 60:115–134
52. Heinzel A, Barragan VM (1999) A review of the state-of-the-art of the methanol crossover in direct methanol fuel cells. J Power Sources 84(1):70–74
53. Ledesma S, Cerda G, Avina G, Hernández D, Torres M (2008) Feature selection using artificial neural networks. In: Mexican international conference on artificial intelligence, pp 351–359
54. Mavrovouniotis M, Yang S (2015) Training neural networks with ant colony optimization algorithms for pattern classification. Soft Comput 19(6):1511–1522
55. Bache K, Lichman M (2013) UCI machine learning repository. University of California, School of Information and Computer Science
56. Larson SC (1931) The shrinkage of the coefficient of multiple correlation. J Educ Psychol 22(1):45
57. Hastie T, Tibshirani R (1996) Discriminant adaptive nearest neighbor classification. IEEE Trans Pattern Anal Mach Intell 18(6):607–616
58. Quinlan JR (2014) C4. 5: programs for machine learning. Elsevier, London
59. Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27