Informatics Bulletin, Faculty of Computers and Artificial Intelligence, Helwan University
Published Online *Vol 7 Issue 2, July 2025*
(https://fcihib.journals.ekb.eg)

A Survey on Metaheuristic Algorithms Utilized for Feature Selection

Ahmed SabryAbolaban[1*], O. E. Emam[2], and Safaa. M. Azzam[2]
[1*]Information Systems and Digital Transformation administration, Shrakia governorate, Egypt.
[2] Department of Information Systems, Faculty of Computers and Artificial intelligence, Helwan University, P.O. Box 11795, Egypt.
Ahmed_Sabry_1053@fci.helwn.edu.eg *, Emam_o_e@yahoo.com , eng_safaa_azam@yahoo.com

***Abstract*— Feature selection (FS) has become an important step in data preprocessing, not only in data mining (DM) but also in machine learning (ML), owing to the ever-increasing amount of data. To tackle the challenge of selecting relevant features, many techniques have been proposed over time. In recent years, metaheuristic algorithms for feature selection, which are divided into swarm intelligence (SI), evolutionary algorithms (EA), and physics base algorithms (PA), have become increasingly popular and are now considered the most effective option compared to other methods. Our research aims to investigate the current challenges associated with feature selection using metaheuristic algorithms. We are particularly interested in exploring the outstanding performance of numerous metaheuristic algorithms for feature selection that have been observed in various areas over the past fifteen years. The study was segmented into several parts. At first, we presented the idea of feature selection. After that, we analyzed the scientific context that elaborated the issues related to feature selection and metaheuristic algorithms. Later on, we investigated the architecture of these algorithms and then proceeded towards the major metaheuristic algorithms that are commonly used in the domain of feature selection. Ultimately, we highlight the primary sources of datasets and some of the machine learning classifiers that are utilized in this field.**

***Index Terms*— Feature Selection (FS), Machine Learning (ML), Metaheuristic, Exploration, Exploitation.**

I. Introduction

Data's importance and growth are rapidly increasing in today's technology world, especially in the healthcare industry. This data provides valuable information that can help predict future outcomes. Healthcare data is generated through e-records and contains information about the health history of several patients. An AI-based system can help medical practitioners detect and classify diseases early with this healthcare data and classification neural networks. The health records of patients may contain numerous attributes, some of which could be unnecessary or not relevant to the particular ailment. Eliminating these features which are not relevant is recommended to enhance the accuracy of disease classification and prognosis [1].

Modern applications are generating a vast amount of data, with the number of instances and features increasing at an unprecedented rate. This sudden surge in data size, known as Big Data, presents a significant challenge for the latest machine learning algorithms [2] and [3]. Eliminating irrelevant, noisy, and redundant data has become a crucial technique for machine learning (ML) or data mining (DM) for enhancing classification accuracy, so selecting relevant features is very important for developing efficient machine learning and data mining algorithms [4].

FS methods using machine learning classifiers differ between supervised and unsupervised learning models. While supervised learning models select features based on an output label class, unsupervised learning models don't need an output label class for feature selection. Filter, wrapper, and embedded techniques are the three primary methods used for feature selection. The filter method is a technique that involves the application of statistical analysis for assessing the characteristics of the proposed approach, and in contrast for assessing the efficacy of various feature subsets, wrapper methods employ a classification model in conjunction with a search algorithm. The embedded method involves simultaneously discovering the relevant feature subset and conducting classification [5].

Discovering the relevant combination or subset of features is a challenging optimization problem that belongs to the NP-Hard class [6]. The term optimization is used to describe the act of identifying the most optimal or least optimal solution for a given problem that requires optimization. The process entails utilizing a variety of machine-learning techniques and various optimization algorithms found in scientific code libraries [7]. Selecting the right algorithm to solve a particular optimization problem can be challenging. In machine learning, Continuous function optimization is a widespread technique used to find the minimum or maximum value of numerical input parameters, such as floating-point values, and the function typically returns a parameter evaluation of the real world. This method is useful in identifying continuous optimization problems as opposed to combined optimization problems with discrete variables [8].

Metaheuristic algorithms are effective in dealing with combinatorial problems, and extensive research has shown that algorithms outperform exhaustive or greedy methods. [9]. latest metaheuristic algorithms are heavily inspired by nature and have become popular in the (FS) field [10].

Our study will be centered on modern metaheuristic algorithms that have come to light in the past twenty years and have been employed to optimize feature selection.

II. BACKGROUND

The upcoming section will shed light on the challenges associated with optimizing feature selection (FS), as well as the concept of metaheuristics.

A. Difficulties associated with the utilization of feature selection methods:

Algorithms based on the process of evolution challenges.

Evolutionary algorithms (EAs) are often utilized to tackle complex real-world problems, but one of the major disadvantages of their problem-solving abilities is their limited capacity for both exploring and exploiting potential solutions. Additionally, EAs face other limitations, including convergence issues, long computational times, many parameters, and difficulty in parameter tuning. An analysis was carried out by Tejas M. Vala on various Evolutionary Algorithms (EAs), including BA, ABC, GA, HAS, CS, FA, DE, and PSO [11]. The analysis was focused on evaluating three key metrics: reliability, solution quality, and efficiency. This was done using twelve different attributes.

Class imbalance

The term "class imbalance" refers to the situation in which a dataset has an unequal distribution of data, with one class containing a significantly larger number of samples than the other class. While working with datasets that have a high number of dimensions, the problem of class imbalance can arise. In cases where there is a significant imbalance between the numbers of samples in different classes, most classification techniques tend to give more importance to the most common class and tend to ignore the plurality class. In 2018, S. Maldonado and J. Lopez [13] tackled this issue by focusing on two key types: Support Vector Data Description and Cost Sensitive SVM. They found that resolving the class imbalance problem significantly improved the final predictive accuracy level.

Data complexity

Measuring the complexity of data can help speed up the FS process and reduce search space for evolutionary algorithms. This is done before classification with two objectives: reducing the feature space to decrease time complexity and using a static search space for convenience in the evolutionary algorithm. In 2020, S. Sarbazi-Azad introduced a new cultural algorithm that includes five data complexity measures and their significance [12].

Outliers' challenges.

Data outliers can greatly impact the accuracy of final predictions by deviating from the observed sample data. M. B. Naranjo et al. [11], in 2021, a novel approach based on SVM has been introduced that effectively handles outliers in high-dimensional datasets. Compared to other traditional classifiers, when encountering 5% of SVM outliers, the RL-FS-M approach delivers a superior accuracy rate of 99.73% ACC and 99.72% AUC. Similarly, Fisher-SVM and RFE-SVM techniques show improved outcomes in the presence of 5% of SVM outlier dataset problems.

Stability

When it comes to the neural network classification problem, stability refers to how an algorithm affects prediction behavior during the training phase. This is a crucial consideration, especially when dealing with medical datasets where accuracy is paramount for disease diagnosis. In 2017, B. Pes et al.[14] Emphasized the significance of incorporating stability requirements when designing classification. Additionally, the study compared the analytical accuracy rates of various classifiers using the same dataset.

B. Metaheuristic concepts

Metaheuristic optimization technique:

Metaheuristic optimization refers to the process of optimizing problems using metaheuristic techniques; this set of skills can have practical applications in many fields, including vacation planning, online travel, engineering, business, and other relevant areas [15]. These techniques are particularly useful in situations where time, resources, and money are scarce. Optimization problems encountered in real-world scenarios are often intricate and challenging to solve. However, optimization techniques have demonstrated remarkable effectiveness in addressing such complex problems. These methods are extensively utilized to navigate through a vast solution space and deliver optimal solutions within a reasonable time frame.

Metaheuristic algorithms are classified into five classes; the first class is an optimization algorithm used in Bio-Inspired algorithms. When faced with complex problems and searching space, imitating the natural behaviors of biological creatures can be an effective way to find solutions. Bio-inspired algorithms imitate the behavior of living organisms that search for food and mates. They use their logical reasoning and thinking abilities to explore alternative solutions for complex problems. Additionally, there are nature-inspired algorithms - a type of optimization algorithm that utilizes innovative techniques and approaches to achieve the best possible solution for optimization problems within specific search space constraints. These techniques are commonly known as Nature-Inspired algorithms; they imitate the natural actions of animals or birds as they hunt for food or seek a partner (matting); the third class is physics–based algorithms. This approach involves searching for feasible solutions that meet given constraints within a given solution set, both globally and locally. The approach takes cues from natural occurrences like the conduct of particles or atoms in particular environments and leverages them to tackle optimization problems; the fourth class is Evolutionary Algorithms. Metaheuristic approaches such as Evolutionary Algorithms possess the ability to tackle NP problems that are beyond the scope of polynomial time solutions. This method draws inspiration from biological evolution and natural selection, following a sequence of four key steps: initialization, selection, genetic operators, and termination. The fifth class is Swarm-based algorithms, a type of computing methodology that utilizes the cooperation of multiple natural and artificial individuals to solve problems through self-organization. These individuals can be bird flocks, animal groups, ant colonies, or fish schools [16].

Recent metaheuristics algorithms structures.

In this subsection, we provide details on the typical definitions and structures, representation of solutions, operators, selection methods, evaluation of fitness values, and many modern metaheuristics that make use of machine learning techniques, specifically classifiers.

Results simulation.

Metaheuristic optimization techniques usually consist of a group of possible solutions that are often presented as a series of numerical values. Binary sequences are often used to represent the preferred set of features in metaheuristic (FS) approaches [17]. As depicted in (Figure 1), we can see an example of a possible solution with its selected features, consisting of eight features in total. Five of these eight features are selected and represented as ones, while the remaining ones are not selected as zeros. This binary encoding allows for generating 2^n feature subsets.

0 1 0 1 1 0 1 1

Fig. 1 Wrapper-based feature selection algorithms often utilize a simulated solution that takes the form of binary encoding.

Achieving optimal performance, predictive accuracy, and faster convergence speed in metaheuristic approaches requires balancing between exploration and exploitation activities. Xu and Zhang's study sheds light on this key factor [18]. At present, answering this question is not straightforward. Nevertheless, by applying fitness landscape analysis and information landscape techniques, it may be possible to achieve a more optimal equilibrium between exploration and exploitation endeavors [19]. Metaheuristics can effectively enhance and optimize each stage of the algorithm, leading to superior results compared to other methods. To prevent getting stuck in local optima, it is crucial to venture into unexplored regions. Exploration is employed to eliminate local optima, while exploitation concentrates on discovering nearby alternatives to the current solution.

Fitness value evaluation.

Choosing a suitable fitness function for the optimization process is an indispensable element, as assessing the efficacy of the chosen features can be beneficial. The formula or function chosen can significantly impact the speed of optimization and the accuracy of predictions. Additionally, it is crucial to keep the selected features to a minimum. Metaheuristics frequently employ a commonly used fitness function for this purpose in (Eq.1) [20]:

$$\text{Fit}= \alpha ER + \beta \left(\frac{|S|}{|O|}\right) \qquad (1)$$

When calculating the classification error (ER), considering the selection of feature number or size that has been selected is very important and denoted by$S$; the original dataset's feature lengths need to be determined, represented by$|O|$.

Two values represent the significance of feature size, and classification error is denoted by $\alpha \in [0,1]$ and$\beta \in (1-\alpha)$. Some variations of these functions have a dual objective: enhancing classification accuracy and reducing the number of features that will be selected is the other objective. The multiobjective formulation for the two objectives denoted in (Eq.2):

$$\text{Minimize } \int \quad 1, \int \quad 2 \qquad (2)$$

Subject to

$$\int \quad 1 = |S|$$

$$\int \quad 2 = ER$$

For reducing the set of selected features, it is recommended to try out and employ different feature sets, as the ideal number of features $|S|$ to use is uncertain in the beginning. Even if we do know the optimal number of features beforehand, obtaining the optimal subset can pose a challenge as it involves exploring a vast number of potential options $\left(\frac{|S|}{|O|}\right)$ combinations. After selecting the features, the next step involves assessing their quality. Accuracy and F1-measure are the most commonly used measures for comparing classification quality [21]. For determining the accuracy, dividing instances that were classified correctly by the overall number of instances is necessary. On the other hand, the F1 measure offers a more equitable assessment of precision and recall, which gauge the correctness of actual predictions and the capacity to identify genuine occurrences, respectively. In cases where datasets contain disproportionate class distributions, the F1 measure is generally considered more suitable than accuracy for evaluation purposes.

Transfer function operators.

In the realm of optimization problems, metaheuristics were first developed to tackle continuous optimization problems. Later on, binary versions of these metaheuristics were introduced initially, metaheuristics were created to address optimization problems that involved continuous variables. However, subsequently, binary versions of these metaheuristics were also introduced. Transfer functions are employed to transform or modify continuous versions into binary ones. The literature often cites two transfer functions, the (S) and (V) shaped, as the most notable transferring methods. The transfer functions provide information on the likelihood or probability of choosing a feature or not. The transfer function of PSO is in (Eq. 3) [22]:

$$T\left(x_j^i(t)\right) = \frac{1}{1+exp^{-x_j^i(t)}} \qquad (3)$$

Where x_j^i represents the feature at index jth in the solution x, at the jth dimension and current iteration, is represented as t. If S-shaped transfer functions are used, updating an element can be accomplished by applying (Eq.4), the value of r is chosen randomly between 0 and 1.

$$x_j^i(t+1) = \{1 \;\;, \; r < T(x_j^i(t+1)) \; 0, \qquad otherwise \qquad (4)$$

Eq. 5 can be utilized for updating a feature based on the probability values provided in Eq. 6, where *r* represents a value ranging from 0 to 1 and is generated randomly.

To update a feature using V-shaped transfer functions, you can utilize (Eq.5) based on the probability values provided in (Eq.6)

$$x_j^i(t+1) = \{-x_j^i(t) \;\;, \; r < T(x_j^i(t+1)) \; x_j^i(t), \qquad otherwise \qquad (5)$$

$$T\left(x_j^i(t)\right) = \left|tanh\,(x_j^i(t))\right| \qquad (6)$$

Metaheuristic algorithm's main activities.

The flowchart diagram shown in (Figure 2) illustrates the primary activities performed by metaheuristic algorithms. The first step is to generate a group of individuals, which is then followed by assessing their fitness values. The process of generating new candidate solutions through exploration and exploitation operators begins after this point and continues until a termination condition is satisfied. It's vital to avert the duplication of candidate solutions during optimization to prevent wastage of computational resources. Faster variants of these algorithms, such as dynamic programming or parallel, can yield better outcomes by increasing the number of fitness evaluations within a shorter period.

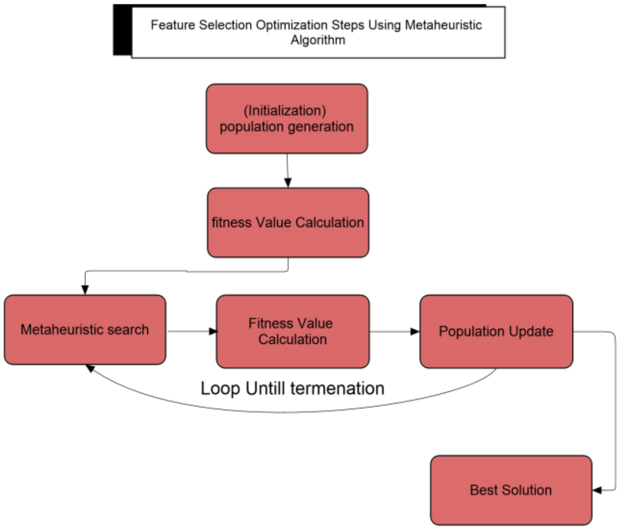
Feature Selection Optimization Steps Using Metaheuristic Algorithm
(Initialization) population generation
fitness Value Calculation
Metaheuristic search
Fitness Value Calculation
Population Update
Loop Untill termenation
Best Solution

Fig. 2 activities performed by metaheuristic algorithms.

Metaheuristic algorithms parameters.

Metaheuristic algorithm parameters are a crucial area of research, as they greatly affect the algorithm's performance [23]. Determining the suitable individual size and number of generations is a crucial factor in effectively utilizing population-based metaheuristics. Moreover, to enhance the quality of solutions and computation time, it is important to adjust critical parameters like the selection method, number of iterations, mutation ratio, and convergence ratio. Although there are some common parameters for all population-based metaheuristics, some recent algorithms have specific parameters that require fewer or more adjustments. Although it may seem like algorithms with fewer parameters are better, having a greater number of parameters can actually be beneficial in guiding and improving the optimization process [24].

Common classifiers utilized in FS algorithms.

FS algorithms that involve supervised machine learning techniques often utilize a division of the dataset into two categories: training dataset and testing or validation sets. During experiments that assess the accuracy of classifiers using cross-validation with selected features, K-Nearest Neighbors (KNN) is a commonly utilized algorithm in FS due to its easy implementation and lower computational costs compared to other classifiers. Given that various fitness evaluations are performed during these experiments, selecting a classifier heavily relies on the learning algorithm's speed. It is denoted that while SVM is known to deliver superior classification performance, it can be computationally expensive to use as a classifier. In contrast, deep learning has demonstrated exceptional performance in solving classification problems, making it a popular research field these days. FS can be performed using a variety of classifiers, including Optimum Path Forest (OPF), Random Forest (RF), Artificial Neural Networks(NN), Naive Bayes(NB), and Logistic Regression(LR) [25]. When conducting experiments, it is crucial to bear in mind that classifiers might exhibit varying performance levels across different domains. Consequently, it is recommended that researchers evaluate multiple classifiers instead of solely relying on one.

Evaluation metrics.

The assessment of algorithm efficacy depends on calculating fitness function. The evaluation of these scores considers not only the precision of predictions made but also the number of features selected in the subset. Furthermore, it is crucial to consider the time taken by the algorithms to execute. Prediction performance is often evaluated using Accuracy and F-measure as metrics, and there exist several other metrics such as Area under the Curve, Root Relative Squared Error, Kappa statistic [26], Correlation Coefficient [27], Root Mean Square Error, Precision, Recall, Relative Absolute Error, and Mean Absolute Error [28].

III. LITERATURE REVIEW

As a part of our study, we will discuss ten metaheuristic algorithms employed for FS (Butterfly Optimization Algorithm (BOA)[29], Grasshopper-Optimization Algorithm (GOA)[30], SalpSwam Algorithm (SSA)[31], Dragonfly-Algorithm (DA)[32], Crow Search Algorithm (CSA)[33], Whale Optimization Algorithm (WOA)[34], Sine Cosine Algorithm (SCA)[35], Ant Lion Optimization (ALO)[36], Grey Wolf Optimization (GWO)[37], and Bat-Algorithm (BA)) [38]. The algorithms will be presented in descending order, starting with the most recent and concluding with the oldest, where we will present some of the hybrid algorithms for the same purpose.

A- Metaheuristic Algorithms

Ant-Lion Optimization (ALO)

Mirjalili [36] introduced ALO in 2015; the algorithm depends on the inspiration from the hunting behavior of ant lions and mimics their actions towards their prey, specifically the way they trap ants that wander around in search of food. The movement of ants is modeled as follows in (Eq.7):

$$X(t) = \sum_{i=1}^{t} \quad 2r(t_i) - 1 \qquad (7)$$

The variable denotes the total steps taken during a random walk, while 'r' denotes a random value within the range of 0 to 1.

The ALO mechanism operates based on a set of guidelines. Through random walks in multiple dimensions, ants are able to navigate and explore their environment. In contrast, ant lions construct traps according to their fitness levels, with a higher level leading to a more effective trap. Ant lions that possess more effective traps are more likely to capture ants successfully. Once an ant falls into the trap, it becomes more restricted in its movement, and if it is weaker than the ant lion, it will be dragged under the sand. After catching their prey, ant-lions change their location and create new traps.

Emery et al. [14] introduced a wrapper-based feature selection to enhance classification based on ALO. These algorithms utilize a balancing act between exploration and exploitation operations through a single operator. The algorithms use binary and transfer functions as part of their implementation. The study evaluated the performance of the ALO algorithms by comparing PSO, GA, and binary BA. The researchers used 20 datasets from the UCI repository to carry out the evaluation; as per the findings of the study, the ALO algorithms were successful in identifying the best feature subset, irrespective of the approach adopted for producing the initial population or the application of other operators in the algorithms, the statement holds. In their work, Wang et al. [50] presented a technique that incorporates wavelet SVM and Levy flights to overcome the issue of local optima in reducing hyperspectral images called ALO. A new criterion was introduced to measure classification accuracy, which proved to be more efficient than other algorithms as it identified the optimal solution with fewer bands. Zawbaa et al. [51] proposed an Ant Lion Optimizer (ALO) for FS, where a parameter was introduced to control the trade-off between exploration and exploitation, resulting in chaotic behavior. The researchers made iterative adjustments to the parameter to regulate the ants' random walk and restrict their level of exploration for optimal results. They compared their findings using various quality metrics with PSO and GA algorithms.

Bat Algorithm (BA)

Yang [38] 2010 developed the BA metaheuristic. The concept of this method is based on the bat's echolocation technique, which is used to detect obstacles and prey and locate their nests. By bouncing sound waves off objects in their surroundings, bats are able to differentiate between them. This technique can be applied to optimize objective functions. The BA algorithm involves bats moving through a d-dimensional domain, with velocity (v_i) at position (x_i) and a frequency range between (f_{min}) and (f_{max}), Modifying the frequency of the emitted pulse can lead to an improvement in the algorithm's efficiency. The wavelength and frequency play a key role in determining the convergence of the BA, where shorter wavelengths and higher frequencies can cover shorter distances. Lower frequencies, on the other hand, have longer wavelengths and can reach farther distances. To optimize the algorithm's efficiency, the frequency can be adjusted during implementation. The positions (x_i) and movement and velocities (v_i) of bats in a d-dimensional space are updated continuously over time using specific equations for each timestamp (t).

$$f_i = f_{min} + (f_{max} - f_{min})\beta \qquad (8)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \qquad (9)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad (10)$$

The vector ' β ' is a randomly selected value within the range of 0 to 1, while 'x_*' represents the current best available solution. A random solution is generated during the process of local search using a random walk that follows a given formula:

$$x_{new} = x_{old} + \propto A^t \qquad (11)$$

$\propto$ It is a value that varies randomly and falls within the range of -1 to 1 and is multiplied by the average loudness of all the bats in the population at a specific timestamp (t), which is denoted as (A^t).

Rodrigues et al. [55] developed a binary wrapper named BA for FS. Another study by Jeyasingh et al. [56] developed a version of BA specifically for selecting features in breast cancer datasets.

Butterfly-Optimization-Algorithm (BOA)

Arora et al. [29] introduced BOA in 2019, which refers to a model that mimics the actions of butterflies as they hunt for nourishment and seek a partner for reproduction. Butterflies attract each other by emitting fragrance and move towards the butterfly with the strongest fragrance or move randomly. The fragrance emission is an important aspect of this simulation is generated using (Eq.12)

$$\int \quad = cI^a \qquad (12)$$

The sensory modality (c), intensity of the stimulus (I), and power exponent (a) that varies based on the modality all have an impact on the received fragrance amount (f).
Butterflies move around randomly to find food or a mate. They have two phases of searching: global and local. During each iteration, the butterfly can move towards the most optimal butterfly nearby, g*, using the following equation (Eq.13):

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \qquad (13)$$

The ith butterfly (or solution) represented by x_i^{t+1} a value ranges from 0 to 1 and is generated randomly, similar to how both global and local searches operate denoted in (Eq.14):

$$x_i^{t+1} = x_i^t + (r^2 \times x_i^t - x_k^t) \times f_i \qquad (14)$$

Within the context of the same group, it is possible to define a local random walk using Equation 9 for the values of j and k. During the search for food or potential mates, individuals can choose to explore their surroundings on both a local and global level by either focusing on a specific area or expanding their search to cover a wider region. By introducing a switch probability p, the BOA can alternate between the two search levels. The iteration process will persist until the stopping criterion is satisfied. To achieve optimal feature subset selection, Arora and Anand have suggested several BOA variations [39]. In order to navigate through the distinct possibilities of the problem, the algorithm utilizes a threshold function. The suggested algorithms have demonstrated their effectiveness in identifying a subset of features that is close to optimal. To address the challenge of FS, Sadeghian and their team have suggested employing an Information Gain binary BOA [40]. In order to confirm that the proposed method efficiently eliminates unnecessary and repetitive attributes while selecting the most appropriate set of features, datasets obtained from the UCI repository were used for carrying out experimental tests.

Crow Search Algorithm (CSA)

Askarzadeh et al. [33] 2016 introduced a technique called CSA, which is an algorithm that takes inspiration from the behavior of crows. Crows are known for their high intelligence and impressive capabilities, such as facial recognition, communication skills, and the ability to remember where they hide their food. They live in groups and work together to locate sources of food. The CSA algorithm emulates the way crows hide and retrieve their food in secret places. The effectiveness of this algorithm has been demonstrated in solving engineering problems that possess constraints. In CSA, Crows tend to have their unique hiding places for food, which they consider as the most effective solution they have discovered until now, denoted as $m^{i,iter}$ for crow I at a given (iter). During the process, crows persisted in searching for fresh food sources. At a certain point, one crow (let's call it crow j) might expose the location of its stash to another crow (crow i), who could then choose to steal from it. However, in case Crow J senses the presence of Crow I, it might relocate to another area with the intention of confusing the one chasing it. This would lead to a change in Crow's position, as illustrated below in (Eq.15):

$$x^{i,iter+1} = \{x^{i,iter} + r_i \times fi^{i,iter} \times (m^{j,iter} - x^{i,iter}),\ \ r_j \geq AP^{j,iter}\ a\ random\ position, \qquad otherwise \quad (15)$$

Two values range from 0 to 1 and are generated randomly. They are denoted by r_i andr_j. The flight distance of Crow I is fii,er, and Crow's awareness probability is represented by $AP^{j,iter}$. Small values of fi indicate local search, while large values indicate global search.

Ouadfel et al. [46] came up with a proposal that tackles the challenge of premature convergence in the FS process carried out by the CSA algorithm. By incorporating a novel global search method and awareness, probability, exploration, and exploitation balancing were greatly improved. According to their experiments with UCI datasets, the convergence speed was increased.

Dragonfly Algorithm (DA)

Mirjalili et al. [32] 2016 introduced a new optimization process called DA, which takes inspiration from the collective behaviors of dragonflies. Throughout the process, the DA's primary tasks include avoiding collisions, matching velocity, and maintaining cohesion within the swarm. Moreover, the swarm entities attempt to approach food sources while steering clear of any potential threats. Separation is simulated using computational modeling, which results in individuals moving in multiple directions.

$$S_i = -\sum_{j=1}^{N} \quad X - X_j \qquad (16)$$

Where the dragonfly's current position, denoted by X, is surrounded by its *jth* neighbor, represented by X_j. The number of all neighbors is indicated by *N*. The alignment is determined in the following equation (Eq.17):

$$A_i = \frac{\sum_{i=1}^{N} \quad V_j}{N} \qquad (17)$$

In j^th, V_j is the velocity of individual Cohesion is modeled by the following equation (Eq.18)

$$C_i = \frac{\sum_{i=1}^{N} \quad X_j}{N} - X \qquad (18)$$

To make a move towards the food, you may utilize the formula provided below in (Eq.19):

$$F_i = X^+ + X \qquad (19)$$

Where X^+ denotes the food location. To move away from the opponent, follow the provided equation here (Eq.20):

$$E_i = X^- + X \qquad (20)$$

The enemy's position is denoted by. The step (Δ) and position (X) vectors serve as a means to monitor the dragonflies' movements. The former stores the direction, while the latter stores the current location. By utilizing these vectors, the dragonflies' positions can be updated with ease.
The step vector is calculated using various weights and the iteration number. The position vector is then calculated using the step vector as follows in (Eq.21):

$$\Delta X_{t+1} = sS_i + aA_i + cC_i + fF_i + eE_i + w\Delta SX_t \qquad (21)$$

Denoted that the position vector can be calculated using the variables: separation weight (s), alignment weight (a), cohesion weight (c), food factor (f), enemy factor (e), inertia weight (w), and iteration (t). According to the following equation (Eq.22):

$$X_{t+1} = X_t + \Delta X_{t+1} \qquad (22)$$

UCI datasets were used to test the binary DA feature selection method, as introduced by Mafarja et al. [44]. A comparison was made between DA from the side and PSO and GA on the other side in terms of the number of selected features and their accuracy. According to the results, the binary DA proves to be a very efficient method for FS. Mafarja et al. [45] devised a wrapper DA that tested eight distinct transfer functions during experiments. The S-shape variant of the DA algorithm has been found to be more efficient compared to its conventional counterpart.

Grasshopper-Optimization-Algorithm (GOA)

Mirjalili et al. [30], in 2018, introduced a solving Optimization problems technique using the GOA where the behavior of grasshopper swarms serves as the basis. Millions of grasshopper's form swarms that can travel vast distances. To navigate their environment, the grasshoppers make sudden movements for exploitation purposes. In the context of GOA, the optimization problem being addressed can be potentially solved by considering the location of each grasshopper; mathematically, the location of a grasshopper is modeled in the following (Eq.23):

$$X_i = S_i + G_i + A_i \qquad (23)$$

In the context where it is being used, the formula for a random walk with respect to the location (X_i) of the ith grasshopper is given by $X_i = r_1 S_i + r_{21} G_i + r_{13}$, ($S_i$) represents the surface stress due to surface waves, (Gi) represents the gravitational force, and (A_i) represents the wind's influence on the movement of a fluid.

(r_1), (r_2), and (r_3) are two values that range from 0 to 1 and are generated randomly.

According to (Eq.24), social interaction is the initial component of the three primary constituents of GOA.

$$S_i = \sum_{J=1,j\neq i}^{N} \quad s(d_{ij})\widehat{d_{\imath\jmath}} \tag{24}$$

For calculating the distance between two individuals d_{ij} (Xi and Xj), two factors play an important role: they are the size of the population (N) represented by the unit vector $\widehat{d_{ij}}$ and the strength function of social forces given as below in (Eq.25):

$$s(d) = fe^{\frac{-d}{1}} - e^{-d} \tag{25}$$

The intensity of attraction is denoted by (f), while the attractive length scale is represented by (l).

The gravity force can be denoted by (Eq.26):

$$G_i = -g\hat{e}_g \tag{26}$$

Where the gravitational constant is denoted by (g) and the vector to the center of the earth is denoted by $(\hat{e}_g)$

Finally, the formula for calculating wind advection is the following equation:

$$A_i = u\hat{e}_w \tag{27}$$

The vector in the wind direction is denoted as $(\hat{e}_w)$, and the constant drift is represented by (u).

Mafarjaet al. [41] have implemented an FS technique called GOA using creative selection operators and population dynamics. The proposed algorithms were assessed for their efficacy using diverse datasets present in the UCI datasets repository. The findings indicated that these algorithms outperformed other existing methods in terms of performance.

Grey Wolf Optimization (GWO)

Mirjalili et al.[37] introduced GWO in 2014, and another version for multiobjectives was proposed [52] in 2016. Grey Wolf lives in a pack with a social structure. In the pack, there are 5-12 wolves, and each wolf is assigned a designation such as alpha, beta, omega, or subordinate based on their dominance factor. The wolf that holds the alpha position within the pack is responsible for making most of the decisions and is considered the most dominant member. In a wolf pack, the beta wolf holds the second-highest position of dominance. Its role involves assisting the alpha wolf in making decisions and communicating the alpha's orders to the rest of the pack. Within the wolf pack, Omega is the least dominant member and is submissive to all other wolves. The rest of the wolves are known as subordinate or delta. Grey wolves hunt in groups, and their typical hunting strategy involves tracking and chasing their prey before surrounding and attacking it in a coordinated manner. The GWO was founded with the aim of replicating the social structure and hunting strategies of grey wolves. The method entails classifying the three most prominent solutions into three categories - the alpha category, the beta category, and the delta wolf's category; in a wolf pack, the alpha wolves hold the responsibility of deciding the outcome of the next iteration. On the other hand, the remaining wolves in the pack, often referred to as omega wolves, are not involved in this decision-making process.

The following method is used to simulate the encircling activity of the wolves in (Eq.28 & Eq.29):

$$\vec{E} = \left|\vec{B}.\overrightarrow{X_p}(t) - \vec{G}(t)\right| \tag{28}$$

$$\vec{G}(t+1) = \vec{G}_p(t) - \vec{A}.\left(\vec{E}\right) \tag{29}$$

The $(\vec{A})$ vector is used for exploration, while the $(\vec{C})$ vector is used for exploitation. The grey wolf's prey and its corresponding position vector are represented as (G_p) and $(\vec{G})$, respectively, while the variable " t " represents the iteration. Additionally, the multiplication of the elements is symbolized by the dot (.) notation.

$(\vec{A})$ and $(\vec{C})$ are two coefficient vectors that can be calculated with the following equations:

$$\vec{A} = 2.\vec{a}.\vec{r_1} - \vec{a} \tag{30}$$

$$\vec{C} = 2.\vec{r_2} \tag{31}$$

A range of random values from 0 to 1 is used to fill vectors r_1 andr_2, Vector $\vec{a}$ is also created with identical elements, the values of its elements decrease gradually from 2 to 0 as time passes. Additionally, vectors $\vec{A}$and $\vec{C}$ have elements that fall within the ranges of [-a, a] and [0, 2], respectively.

In order to compute the desired outcome, equations 32 and 33 rely on the positions of the three categories of wolves: alpha, beta, and delta. Hence, their positions are revised using the following:

$$\vec{E}_\alpha = \left|\vec{C}_1.\vec{G}_\propto - \vec{G}\right|$$

$$\vec{E}_\beta = \left|\vec{C}_2.\vec{G}_\beta - \vec{G}\right| \tag{32}$$

$$\vec{E}_\delta = \left|\vec{C}_1.\vec{G}_\delta - \vec{G}\right|$$

$$\vec{G}_1 = \vec{G}_\propto - \vec{A}_1.\left(\vec{G}_\propto\right)$$

$$\vec{G}_2 = \vec{G}_\beta - \vec{A}_2.\left(\vec{E}_\beta\right) \tag{33}$$

$$\vec{G}_3 = \vec{G}_\delta - \vec{A}_1.\left(\vec{E}_\delta\right)$$

$$\vec{G}(t+1) = \frac{\vec{G}_1+\vec{G}_2+\vec{G}_3}{3} \qquad (34)$$

Al-Tashi et al. [53] A binary version for FS was proposed by integrating GWO and PSO. Hu et al. [54] discovered that the Grey Wolf Optimizer algorithm was dependable when applied to real-world optimization problems. Additionally, they introduced a binary version of the GWO that maps transfer functions to binary representations. This binary variant was successful in carrying out FS on UCI datasets with minimal deviations.

Salpswam Algorithm (SSA)

Mirjalili et al. introduced SSA [31] in 2017. This method was influenced by the behavior of rays (slaps) as they forage for food in the deep sea. A series of slaps are connected, with one leading and the rest following, to develop their movement and foraging capabilities.

The swarm of slaps is directed forward to the food source (F) by the leader, and all members follow this direction. In a specific dimension (j) of the search space, the problem leader's position is utilized to determine their course of action; the algorithm cares about the positions of the slaps and the location of the food source as follows in (Eq.35):

$$X_j^1 = \{F_j + a_1\left(\left(ug_j - lg_j\right) * c_2 + lg_j\right),\ a_3 \geq 0\ F_j - a_1\left(\left(ug_j - lg_j\right) * c_2 + lg_j\right),\ a_3 < 0 \qquad (35)$$

The upper bound is denoted by $(ug_j\ )$, while the lower bound is denoted by $(lg_j\ )$. From the equation, it is evident that the leader's position is updated depending on the food location. The coefficient denoted by a_1 ensures exploration and exploitation and balancing between them, as illustrated below in (Eq.36):

$$a_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \qquad (36)$$

While iterating, l represents the current iteration, and L represents the maximum iteration count. Additionally, step size and movement direction are indicated by a_2 anda_3, respectively; they are values ranging from 0 to 1 and are generated randomly.

The follower's position is determined in (Eq.37):

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right), \qquad i \geq 2 \qquad (37)$$

The jth dimension contains the representation of the position of the follower indexed asx_j^i.

In their study, Hegazy et al. [42] enhanced the feature selection SSA algorithm by introducing an inertia weight that helped establish the most optimal solution. The algorithm's performance was evaluated against traditional SSA and contemporary swarm techniques; it exhibited better precision in forecasting and had advanced capabilities for selecting features.

To improve FS. Tubishat et al. [43] made an advanced version of SSA where a new local search technique was introduced, and Opposition Learning was used for population initialization to improve the exploitation; results determined that it was effective in case comparison with other techniques.

Sine Cosine Algorithm (SCA)

Mirjalili [35] 2016 introduced the Sine Cosine Algorithm (SCA). The math model of the algorithm uses sine and cosine functions to achieve balancing between exploration and exploitation by incorporating both random and adaptive variables harmoniously. As a result, the global optimum can be efficiently reached through its convergence. The update mechanism in SCA is structured in a manner that facilitates this process as follows (Eq.38):

$$x_i^{t+1} = \{X_i^t + r_1 \times sin\ sin\ (r_2) \times |r_3 P_i^t - X_i^t|,\ r_{4<0.5}\ X_i^t + r_1 \times cos\ cos\ (r_2) \times |r_3 P_i^t - X_i^t|,\ otherwise \qquad (39)$$

The solution at the t^th iteration is represented by (x_i^{t+1}) i, where i denotes the ith dimension. In the i^th dimension, P_i is used to denote the target. During the iterations, r_1 gradually decreases from a constant value (such as 2) to 0. Additionally, *r2, r3*, and *r4* are random values. The functions sine, cosine, and absolute value are denoted by $(sin\ (.)\ )$, $(cos\ (.))$, and $(|.|)$ respectively. The ranges for r_2, r_3, and r_4 are from 0 to 2p, 0 to 2, and 0 to 1, respectively.

In their research, Hafez et al. [48] presented an FS model called SCA; they aimed to maximize accuracy while minimizing the feature size by combining both fitness functions. Sindu et al. [49], a new SCA model has been created using an elitism approach and an updated technique to identify the best features for classification precisely; according to the results of the experiment, the effectiveness of this new algorithm is considerably high.

Whale Optimization Algorithm (WOA)

Mirjalili et al. [34] 2016 introduced the WOA algorithm, and the hunting behavior of humpback whales served as inspiration. The algorithm emulates the collaborative hunting behavior of these social creatures, where they use bubble nets to trap their prey. WOA is a mathematical model of this behavior and can be used to solve complex optimization problems by employing multiple search agents. To start the process, a solution is randomly selected, and then the most efficient search agent is chosen to update the positions of the others; these agents then move toward the selected agent to search to obtain the optimal solution. The following equations (Eq.40 & (Eq.41) show the mathematical model:

$$\vec{D} = \left|\vec{G}.\vec{X}^{*}(t) - \vec{X}(t)\right| \qquad (40)$$

$$\vec{X}(t+1) = \vec{X}^{*}(t) - \vec{B}.\vec{D} \qquad (41)$$

In the given equation, $\vec{B}$ and $\vec{G}$ represent vectors consisting of coefficients, t denotes the iteration number, $\vec{X}^{*}$ is the target whale, $\vec{X}$ represents the position vector, and the symbol (.) denotes the multiplication of the elements.
The vectors $\vec{B}$ and $\vec{G}$ can be calculated as follows in (Eq.42 &Eq.43) :

$$\vec{B} = 2\,\vec{b}.\vec{r} - \vec{b} \qquad (42)$$

$$\vec{G} = 2\,.\vec{r} \qquad (43)$$

Two vectors, filled with random values, are denoted by ($\vec{b}$) and ($\vec{r}$). The first vector ranges from 0 to 2, while the second one ranges from -2 to 2, respectively.
Equation 27 outlines that to establish an encircling behavior, the impact of $\vec{b}$ gradually diminishes with each iteration.
The whale's position update follows a spiral pattern as a result, which is defined as follows (Eq.44):

$$\vec{X}(t+1) = D'\,.\,e^{bl}.cos\,cos\,(2\pi l)\ + \vec{X}^{*}(t) \qquad (44)$$

X denotes the whale position, Y, while the position of the prey is represented by $(X^{*}, Y^{*}\ )$. The measurement that represents the space separating (distance) the whale from its target is indicated by t, and$D' = \left|\vec{X}^{*}(t) - \vec{X}(t)\right|$. Additionally, b is the logarithmic spiral, and l is a value ranging from -1 to 1. The process of selecting a target depends on the value of $when\ \left|\vec{A}\right|>$ 1; the target is selected randomly from a group of whales, while the best whale is specifically chosen as the target if $\left|\vec{A}\right|<$ 1.

The WOA algorithm for FS, which was proposed by Sharawi et al. [48], is designed to identify the most critical set of features to achieve optimal performance with a minimal feature subset. Mafarja and Mirjalili [47] created two versions of the WOA algorithm that use binary values for FS. They also evaluated the effectiveness of various selection methods. The mutation and crossover operators were improved, and a comparison between the proposed algorithm and other optimizers was made.

B- Categorization of the reviewed metaheuristic algorithms.

In our study, we presented ten metaheuristic algorithms that gave better solutions for optimizing the problems of FS. Now, in (Table 1) we categorized these algorithms according to age, nature, and inspiration.

TABLE 1

THE REVIEWED ALGORITHMS' CATEGORIZATION.

Algori
Butte Optimi Algor
Grassh Optimi Algor
Salp Sv Algor
Drago Algor
Crow S Algor
Wha Optimi Algor
Sine C Algor
Ant L Optimi
Grey V Optimi
Bat Alg

C- Hybrid metaheuristic algorithms

In the domain of FS, hybrid algorithms have been devised that amalgamate existing metaheuristics or classical algorithms to capitalize on the advantages of each and elevate overall performance. Typically, hybrid metaheuristic algorithms surpass their single metaheuristic counterparts, resulting in the creation of more effective and adaptable algorithms [57]. This section provides a brief overview of some of these hybrid algorithms.

Zhang et al. [58] proposed an innovative HHO method that employs SSA to boost the optimizer's search capability. HHO is a new optimization technique that was introduced by Chen et al. [59]; it integrates DE, chaos, and topological multi-population strategies for better optimization results. The algorithm has been compared with other studies and demonstrated its effectiveness in solving complex optimization problems.

Mafarja and Mirjalili [60] utilized a strategy that utilized hill-climbing and binary Ant Lion Optimizer (ALO) to generate a cluster of ants. These ants were then consolidated by incorporating the most efficient characteristics with filter FS models. Their algorithm outperformed recent approaches on UCI datasets. In their research paper, Sarhani et al. [61] proposed a new approach for FS, which utilized a combination of PSO and GSA algorithms. In order to encourage variation among the population, they incorporated a mutation operator into their approach. The results of their study indicated that their method outperformed several other popular FS algorithms and metaheuristic techniques. In their study, Pandey et al. [64] demonstrated a precise feature selection approach that combined compressed sensing with principal component analysis and fast independent component analysis. Du et al. [62] in his study he presented a hybrid HHO method in distinct research to inform people about dangerous air pollutants and tackle the issue of air pollution levels. Abdel et al. [63] introduced a new feature selection optimization approach by integrating the HHO algorithm with SA in their research. In their study, Ibrahim et al. [64] introduced a combined approach that utilizes both Particle Swarm Optimization (PSO) and Salp

Swarm Algorithm (SSA) to enhance the exploration and exploitation processes effectively. According to recent studies, a new search algorithm named the Stochastic Search Algorithm (SSA) has been introduced by Neggaz et al. [65]. This algorithm utilizes Simulated Annealing (SA) and a Disrupt Operator. This approach is designed to handle stagnation issues effectively. Arora et al. [66] introduced a hybrid optimization algorithm that integrates the Grey Wolf Optimizer (GWO) and the Cuckoo Search Algorithm (CSA) to attain the global optimum. This approach shows great promise. Statistically, it has been shown to outperform other algorithms.

Deb et al. [67] introduced an enhanced Chicken Swarm Optimization algorithm by incorporating a novel constraint-handling technique and an updated method for roosters to address stagnation issues and achieve better convergence and optimization performance through a hybrid algorithm. The algorithm that was proposed demonstrated that, in the literature, a level of competitiveness comparable to other optimization algorithms was documented.

Kihel et al. [68] proposed a study in which A new feature selection theory has been introduced which draws inspiration from AIS-based clonal selection. In the study, FA and clonal selection algorithms were utilized to pinpoint the most significant features from a given dataset. Two FS algorithms based on the Immune Firefly Algorithm were created, which produced significantly improved results in comparison to various other algorithms. The results conducted on UCI datasets provided evidence of the effectiveness of these hybrid algorithms.

IV. CLASSIFIERS AND DATA SETS USED FOR FEATURE SELECTION OPTIMIZATION

As part of our investigation, we are providing details about the datasets utilized in benchmarking for machine learning. FS algorithms utilize these datasets during experiments. To determine the effectiveness of FS algorithms, it is crucial to conduct experiments on established datasets that have well-defined benchmark features. These datasets should be easily accessible and have been used in previous research, so that they cater to a wide audience of readers and researchers. Moreover, they should offer a level playing field for experimentation. The crucial aspect to consider when performing experiments is the dataset size, which includes the number of records and attributes or features. Evaluating the efficiency of metaheuristics on extensive datasets consisting of numerous instances and features is of utmost significance.

The characteristics of the top 10 renowned datasets are shown in this study. The data in (Table 2) is sourced from the Machine Learning Repository at the University of California, Irvine (UCI) [69]; it is a well-known source of datasets for machine learning studies. According to the results of an analysis of 82 studies which is conducted by D.Tansel et al. in [70], the frequently used datasets are presented in (Figure 3) On the other hand, the most used classifiers used for FS, according to D.Tansel study are shown in (Figure 4).

There are several widely used biomedical datasets available for classification purposes, including ColonTumor, DLBCL-Harvard, and Nervous-System. The instances in these datasets have a high number of dimensions, ranging from 2,000 to 12,600, and there are a number of well-known cancer datasets available as well. In a recent study, the Gene Expression Model Selector (GEMS) system was proposed along with the development of five new datasets that have been designed specifically to assist in the treatment of cancer patients [71].

Kaggle is a website that has been active since 2010 and is popular among data scientists and machine learning enthusiasts. Users are given the ability to access a vast selection of datasets and share their datasets using this platform. Data science enthusiasts, including scientists and engineers, have the opportunity to work together and also challenge each other through Kaggle's platform. The website provides various real-life datasets in different formats, along with easy access to algorithms and code. Currently, Kaggle offers over 100,000 public datasets as of November 2021 [72].

Table 2

The characteristics of the top 10 renowned datasets

Da
Breast C
Congres
Ionosphe
Connect
Bench
Statlog (
Wind
Zoo
Wavefor
Spect He
Lynphog

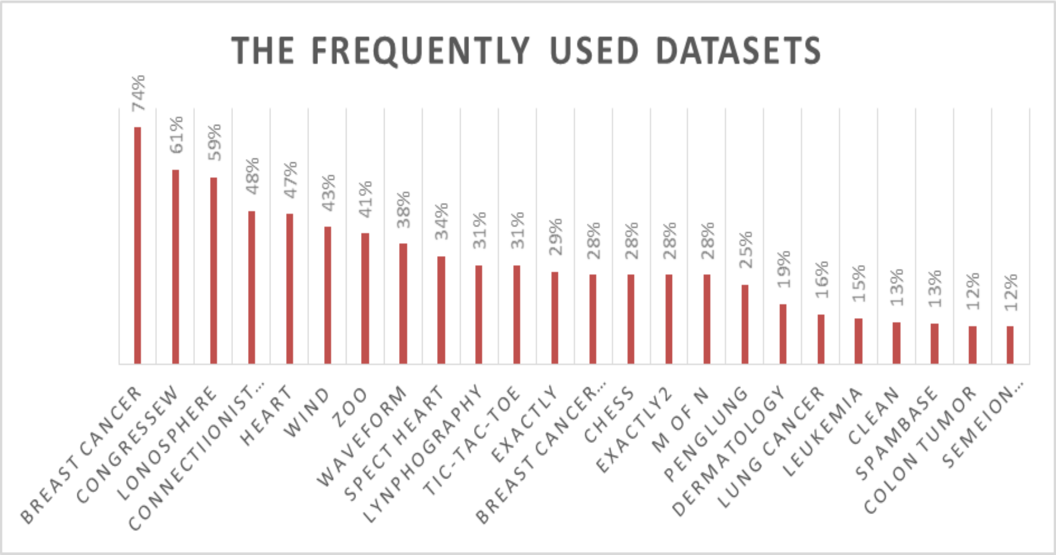

Fig. 3 frequently used datasets

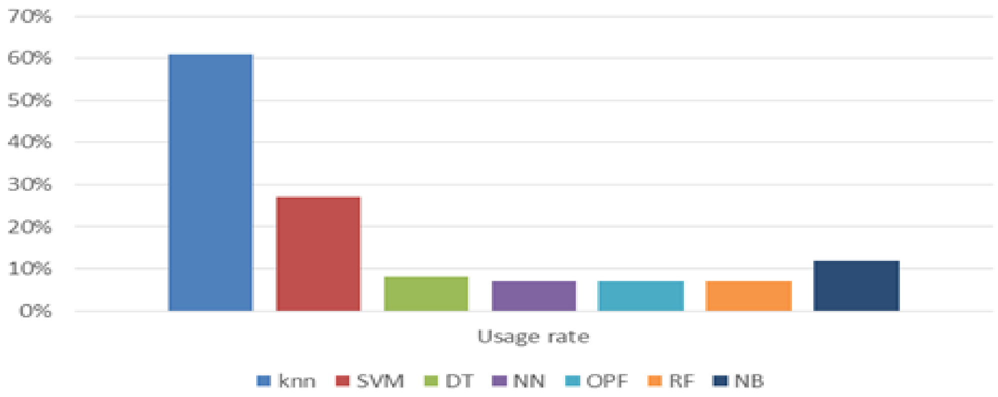

Fig. 4 classifiers utilized for feature selection

V. Conclusion

This paper delves into the topic of feature selection and the various metaheuristic algorithms and their variants that have been utilized to solve this problem. It offers an in-depth explanation and mathematical model of these optimization algorithms to aid researchers in comprehending the issue at hand. Additionally, it presents techniques for tackling feature selection problems through metaheuristic algorithms and outlines the basic definition, significance, and classification of these algorithms. The study encompasses a range of metaheuristic algorithm categories, including evolution-based, swarm-based, physics-based, and human-related algorithms. Specifically, it focuses on ten of them (BOA, GOA, SSA, DA, CSA, WOA, ALO, GWO, and BA) and their variants for feature selection. The study also highlights the challenges faced by feature selection methods, such as outliers, algorithms based on the process of evolution, data complexity, class imbalance, and stability. Finally, the study concludes that the choice of classifier has a significant impact on the quality of the obtained solution, with the KNN classifier being the most used in obtaining the best subset with well-known datasets of the UCI repository, followed by the SVM classifier. Other classifiers, such as RF, NN, DT, and NB, are less commonly used in classification. This presents another gap in using different classifiers to solve classification problems and comparing them with the most used ones. Ultimately, this study will benefit researchers as it provides all the critical factors needed to solve the feature selection problem using metaheuristic algorithms.

REFERENCES

[1] R. A. Kumar, J. V. Franklin, and N. Koppula, “A comprehensive survey on metaheuristic algorithm for feature selection techniques,” *Mater. Today Proc.*, vol. 64, pp. 435–441, 2022.

[2] Z. Obermeyer and E. J. Emanuel, “Predicting the future—big data, machine learning, and clinical medicine,” *N. Engl. J. Med.*, vol. 375, no. 13, p. 1216, 2016.

[3] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, “A survey of machine learning for big data processing,” *EURASIP J. Adv. Signal Process.*, vol. 2016, pp. 1–16, 2016.

[4] L. Zhou, S. Pan, J. Wang, and A. V Vasilakos, “Machine learning on big data: Opportunities and challenges,” *Neurocomputing*, vol. 237, pp. 350–361, 2017.

[5] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[6] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, 2005.

[7] S. S. M. Ghoneim, T. A. Farrag, A. A. Rashed, E.-S. M. El-Kenawy, and A. Ibrahim, “Adaptive dynamic meta-heuristics for feature selection and classification in diagnostic accuracy of transformer faults,” *Ieee Access*, vol. 9, pp. 78324–78340, 2021.

[8] A. Ibrahim *et al.*, “Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm,” *IEEE Access*, vol. 9, pp. 125787–125804, 2021.

[9] A. Deniz, H. E. Kiziloz, T. Dokeroglu, and A. Cosar, “Robust multiobjective evolutionary feature subset selection algorithm for binary classification using machine learning techniques,” *Neurocomputing*, vol. 241, pp. 128–146, 2017.

[10] Y. Zhang, X. Song, and D. Gong, “A return-cost-based binary firefly algorithm for feature selection,” *Inf. Sci. (Ny).*, vol. 418, pp. 561–574, 2017.

[11] A. H. Gandomi and A. H. Alavi, “Krill herd: a new bio-inspired optimization algorithm,” *Commun. nonlinear Sci. Numer. Simul.*, vol. 17, no. 12, pp. 4831–4845, 2012.

[12] S. Sarbazi-Azad, M. S. Abadeh, and M. E. Mowlaei, “Using data complexity measures and an evolutionary cultural algorithm for gene selection in microarray data,” *Soft Comput. Lett.*, vol. 3, p. 100007, 2021.

[13] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *2009 World congress on nature & biologically inspired computing (NaBIC)*, Ieee, 2009, pp. 210–214.

[14] B. Pes, N. Dessì, and M. Angioni, “Exploiting the ensemble paradigm for stable feature selection: A case study on high-dimensional genomic data,” *Inf. Fusion*, vol. 35, pp. 132–147, 2017, doi: 10.1016/j.inffus.2016.10.001.

[15] S. M. J. Jalali *et al.*, “Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting,” *Eng. Comput.*, pp. 1–25, 2021.

[16] W. K. Wong and C. I. Ming, “A review on metaheuristic algorithms: recent trends, benchmarking and applications,” in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, IEEE, 2019, pp. 1–5.

[17] J.-S. Pan, P. Hu, V. Snášel, and S.-C. Chu, “A survey on binary metaheuristic algorithms and their engineering applications,” *Artif. Intell. Rev.*, vol. 56, no. 7, pp. 6101–6167, 2023.

[18] J. Xu and J. Zhang, “Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis,” in *Proceedings of the 33rd Chinese control conference*, IEEE, 2014, pp. 8633–8638.

[19] Y. Borenstein and R. Poli, “Information landscapes,” in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 2005, pp. 1515–1522.

[20] J. Too and S. Mirjalili, “A hyper learning binary dragonfly algorithm for feature selection: A COVID-19 case study,” *Knowledge-Based Syst.*, vol. 212, p. 106553, 2021.

[21] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary grey wolf optimization approaches for feature selection,” *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[22] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, IEEE, 1997, pp. 4104–4108.

[23] Á. E. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, 1999.

[24] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, “Deep learning-based classification of hyperspectral data,” *IEEE J. Sel. Top. Appl. earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, 2014.

[25] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, “Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019),” *Ieee Access*, vol. 9, pp. 26766–26791, 2021.

[26] A. J. Viera and J. M. Garrett, “Understanding interobserver agreement: the kappa statistic,” *Fam med*, vol. 37, no. 5, pp. 360–363, 2005.

[27] R. Taylor, “Interpretation of the correlation coefficient: a basic review,” *J. diagnostic Med. Sonogr.*, vol. 6, no. 1, pp. 35–39, 1990.

[28] C. J. Willmott and K. Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance,” *Clim. Res.*, vol. 30, no. 1, pp. 79–82, 2005.

[29] S. Arora and S. Singh, “Butterfly optimization algorithm: a novel approach for global optimization,” *Soft Comput.*, vol. 23, pp. 715–734, 2019.

[30] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper Optimisation Algorithm: Theory and application,” *Adv. Eng. Softw.*, vol. 105, pp. 30–47, 2017, doi: 10.1016/j.advengsoft.2017.01.004.

[31] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems,” *Adv. Eng. Softw.*, vol. 114, pp. 163–191, 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[32] S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” *Neural Comput. Appl.*, vol. 27, pp. 1053–1073, 2016.

[33] A. Askarzadeh, “A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm,” *Comput. Struct.*, vol. 169, pp. 1–12, 2016.

[34] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[35] S. Mirjalili, “SCA: a sine cosine algorithm for solving optimization problems,” *Knowledge-based Syst.*, vol. 96, pp. 120–133, 2016.

[36] S. Mirjalili, “The ant lion optimizer,” *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[38] X.-S. Yang, “A new metaheuristic bat-inspired algorithm,” in *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, 2010, pp. 65–74.

[39] S. Arora and P. Anand, “Binary butterfly optimization approaches for feature selection,” *Expert Syst. Appl.*, vol. 116, pp. 147–160, 2019.

[40] Z. Sadeghian, E. Akbari, and H. Nematzadeh, “A hybrid feature selection method based on information theory and binary butterfly optimization algorithm,” *Eng. Appl. Artif. Intell.*, vol. 97, p. 104079, 2021.

[41] M. Mafarja *et al.*, “Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems,” *Knowledge-Based Syst.*, vol. 145, pp. 25–45, 2018, doi: 10.1016/j.knosys.2017.12.037.

[42] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, “Improved salp swarm algorithm for feature selection,” *J. King Saud Univ. Inf. Sci.*, vol. 32, no. 3, pp. 335–344, 2020.

[43] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, “Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection,” *Expert Syst. Appl.*, vol. 145, p. 113122, 2020.

[44] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, “Binary dragonfly algorithm for feature selection,” in *2017 International conference on new trends in computing sciences (ICTCS)*, IEEE, 2017, pp. 12–17.

[45] M. Mafarja *et al.*, “Binary dragonfly optimization for feature selection using time-varying transfer functions,” *Knowledge-Based Syst.*, vol. 161, pp. 185–204, 2018.

[46] S. Ouadfel and M. Abd Elaziz, “Enhanced crow search algorithm for feature selection,” *Expert Syst. Appl.*, vol. 159, p. 113572, 2020.

[47] M. Mafarja and S. Mirjalili, “Whale optimization approaches for wrapper feature selection,” *Appl. Soft Comput.*, vol. 62, pp. 441–

453, 2018.

[48] A. I. Hafez, H. M. Zawbaa, E. Emary, and A. E. Hassanien, “Sine cosine optimization algorithm for feature selection,” in *2016 international symposium on innovations in intelligent systems and applications (INISTA)*, IEEE, 2016, pp. 1–5.

[49] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, and M. Hariharan, “Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism,” *Neural Comput. Appl.*, vol. 28, pp. 2947–2958, 2017.

[50] M. Wang, C. Wu, L. Wang, D. Xiang, and X. Huang, “A feature selection approach for hyperspectral image based on modified ant lion optimizer,” *Knowledge-Based Syst.*, vol. 168, pp. 39–48, 2019.

[51] H. M. Zawbaa, E. Emary, and C. Grosan, “Feature selection via chaotic antlion optimization,” *PLoS One*, vol. 11, no. 3, p. e0150652, 2016.

[52] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. dos S. Coelho, “Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization,” *Expert Syst. Appl.*, vol. 47, pp. 106–119, 2016.

[53] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, “Binary optimization using hybrid grey wolf optimization for feature selection,” *Ieee Access*, vol. 7, pp. 39496–39508, 2019.

[54] P. Hu, J.-S. Pan, and S.-C. Chu, “Improved binary grey wolf optimizer and its application for feature selection,” *Knowledge-Based Syst.*, vol. 195, p. 105746, 2020.

[55] D. Rodrigues *et al.*, “A wrapper approach for feature selection based on bat algorithm and optimum-path forest,” *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2250–2258, 2014.

[56] S. Jeyasingh and M. Veluchamy, “Modified bat algorithm for feature selection with the wisconsin diagnosis breast cancer (WDBC) dataset,” *Asian Pacific J. cancer Prev. APJCP*, vol. 18, no. 5, p. 1257, 2017.

[57] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, “Hybrid metaheuristics in combinatorial optimization: A survey,” *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, 2011.

[58] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, “Boosted binary Harris hawks optimizer and feature selection,” *Eng. Comput.*, vol. 37, pp. 3741–3770, 2021.

[59] H. Chen, A. A. Heidari, H. Chen, M. Wang, Z. Pan, and A. H. Gandomi, “Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies,” *Futur. Gener. Comput. Syst.*, vol. 111, pp. 175–198, 2020.

[60] M. M. Mafarja and S. Mirjalili, “Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection,” *Soft Comput.*, vol. 23, no. 15, pp. 6249–6265, 2019.

[61] M. Sarhani, A. El Afia, and R. Faizi, “Facing the feature selection problem with a binary PSO-GSA approach,” *Recent Dev. metaheuristics*, pp. 447–462, 2018.

[62] P. Du, J. Wang, Y. Hao, T. Niu, and W. Yang, “A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2. 5 and PM10 forecasting,” *Appl. Soft Comput.*, vol. 96, p. 106620, 2020.

[63] M. Abdel-Basset, W. Ding, and D. El-Shahat, “A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection,” *Artif. Intell. Rev.*, vol. 54, pp. 593–637, 2021.

[64] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. Abd Elaziz, and S. Lu, “Improved salp swarm algorithm based on particle swarm optimization for feature selection,” *J. Ambient Intell. Humaniz. Comput.*, vol. 10, pp. 3155–3169, 2019.

[65] N. Neggaz, A. A. Ewees, M. Abd Elaziz, and M. Mafarja, “Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection,” *Expert Syst. Appl.*, vol. 145, p. 113103, 2020.

[66] S. Arora, H. Singh, M. Sharma, S. Sharma, and P. Anand, “A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection,” *Ieee Access*, vol. 7, pp. 26343–26361, 2019.

[67] S. Deb, X.-Z. Gao, K. Tammi, K. Kalita, and P. Mahanta, “A new teaching–learning-based chicken swarm optimization algorithm,” *Soft Comput.*, vol. 24, pp. 5313–5331, 2020.

[68] B. K. Kihel and S. Chouraqui, “Firefly Optimization Using Artificial Immune System for Feature Subset Selection.,” *Int. J. Intell. Eng. Syst.*, vol. 12, no. 4, 2019.

[69] A. Frank, “UCI machine learning repository,” *http//archive. ics. uci. edu/ml*, 2010.

[70] T. Dokeroglu, A. Deniz, and H. E. Kiziloz, “A comprehensive survey on recent metaheuristics for feature selection,” *Neurocomputing*, vol. 494, pp. 269–296, 2022.

[71] A. Statnikov, I. Tsamardinos, Y. Dosbayev, and C. F. Aliferis, “GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data,” *Int. J. Med. Inform.*, vol. 74, no. 7–8, pp. 491–503, 2005.

[72] S. Ben Taieb and R. J. Hyndman, “A gradient boosting approach to the Kaggle load forecasting competition,” *Int. J. Forecast.*, vol. 30, no. 2, pp. 382–394, 2014.