Expert Systems With Applications 213 (2023) 119015

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Boosted sooty tern optimization algorithm for global optimization and feature selection

Essam H. Houssein [a,*], Diego Oliva [b], Emre Çelik [c], Marwa M. Emam [a], Rania M. Ghoniem [d,e]

[a] Faculty of Computers and Information, Minia University, Minia, Egypt
[b] Depto. Innovación Basada en la Información y el Conocimiento, Universidad de Guadalajara, CUCEI, Guadalajara, Jal, Mexico
[c] Department of Electrical and Electronics Engineering, Faculty of Engineering, Düzce University, Düzce, Turkey
[d] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[e] Department of Computer, Mansoura University, Mansoura, Egypt

ARTICLE INFO

Keywords:
Sooty Tern Optimization Algorithm (STOA)
Feature selection
Dimensionality reduction
Metaheuristics
Optimization algorithm

ABSTRACT

Feature selection (FS) represents an optimization problem that aims to simplify and improve the quality of highly dimensional datasets through selecting prominent features and eliminating redundant and irrelevant data to classify results better. The goals of FS comprise dimensionality reduction and enhancing the classification accuracy in general, accompanied by great significance in different fields like data mining applications, pattern classification, and data analysis. Using powerful optimization algorithms is crucial to obtaining the best subsets of information in FS. Different metaheuristics, such as the Sooty Tern Optimization Algorithm (STOA), help to optimize the FS problem. However, such kind of techniques tends to converge in sub-optimal solutions. To overcome this problem in the STOA, an improved version called mSTOA is introduced. It employs the balancing exploration/exploitation strategy, self-adaptive of the control parameters strategy, and population reduction strategy. The proposed approach is proposed for solving the FS problem, but also it has been validated over benchmark optimization problems from the CEC 2020. To assess the performance of the mSTOA, it has also been tested with different algorithms. The experiments in terms of FS provide qualitative and quantitative evidence of the capabilities of the mSTOA for extracting the optimal subset of features. Besides, statistical analyses and no-parametric tests were also conducted to validate the result obtained by the mSTOA in optimization.

1. Introduction

Optimization can be indicated as a process in which a solution that minimizes or maximize a problem is found (Heidari et al., 2019). Different problems require optimization tools to search for the best solution. In real life, optimization is present to search for optimal routes, the best flight according to a specific budget, or to save time (Houssein, Emam et al., 2022). Metaheuristics are important tools that provide strategies to create efficient optimization algorithms (Houssein, Helmy et al., 2022). They can be efficiently carried out to solve challenging real-world problems by searching for the optimal solution under certain circumstances (Houssein, Emam et al., 2021; Zamani et al., 2022). It means that the exact solution is not guaranteed, but the algorithm can provide one of the best solutions (Hashim et al., 2022; Nadimi-Shahraki & Zamani, 2022).

Nowadays, one of the most recent development technologies in artificial intelligence is machine learning (ML) (Houssein, Abohashima et al., 2022). The success of ML depends on features extracted from the data. Herein, feature sets can comprise not only features of relevance but also noisy and irrelevant information. In feature selection (FS), the datasets are massive, and the instances are highly dimensional due to the nature of the data acquisition processes (Deng et al., 2019; Houssein, Hassaballah et al., 2022). However, features or attributes do not have equal values. Actually, a certain proportion of features can be repetitive, irrelevant, or both. FS can be defined as an operation that chooses a small subset from the relevant features. FS is an essential technique that helps eliminate irrelevant or redundant features to provide faster data mining algorithms, minimize data storage space, enhance predictive accuracy, facilitate interpretation for the researchers, and minimize the issue of overfitting. The main difference between FS

* Corresponding author.
E-mail addresses: essam.halim@mu.edu.eg (E.H. Houssein), diego.oliva@cucei.udg.mx (D. Oliva), emrecelik@duzce.edu.tr (E. Çelik), marwa.khalef@mu.edu.eg (M.M. Emam), RMGhoniem@pnu.edu.sa (R.M. Ghoniem).

https://doi.org/10.1016/j.eswa.2022.119015
Received 31 March 2022; Received in revised form 17 September 2022; Accepted 9 October 2022
Available online 17 October 2022

and feature extraction is that in the extraction process, the subset of features is created by removing the non-desired information (Yadav et al., 2019).

FS is the crucial step for the preprocessing phase in pattern recognition and data mining problems that focus on filtering features and selecting a useful subset from an obtained training dataset (Thaher et al., 2022). The advantages of performing the feature selection include minimizing the dimensionality curse, reducing the training time for the construction of a model, overfitting problems avoided and improving the generalization built models. For several medical problems involving biomedical signal processing, medical image, DNA microarray data, chemical data, and drugs, the collected data from several medical resources have very high dimensions of feature (Houssein, Hassan et al., 2022). To overcome these problems, related literature has demonstrated the positive impact of using feature selection for several medical domain data.

FS methods used in ML and data-mining can be classified into three methods (Wah et al., 2018): (1) filter, (2) wrapper, and (3) hybrid or embedded methods. In filter methods, every feature subset is assessed by employing an objective function based on the correlation among features or relevance to a target without using a classifier. In contrast, wrapper methods use learning algorithms or classifiers to assess the influence of the selected feature subset, where a group of search strategies can be performed to reach the optimal feature subset that has the most significant importance value. The wrapper-based techniques generate better results when compared to filter approaches (Hastie et al., 2005). Despite the wrapper methods employed for FS demand more time, they provide more precise results (Wang et al., 2018). In the same context, the utilization of metaheuristic algorithms (MAs) can be beneficial by adapting the fitness function in these algorithms to evaluate the quality of every feature group of features (Wan et al., 2016). Embedded feature selection methodologies are suggested to overcome the shortcomings reported in filter and wrapper methodologies. Contrary to wrapper approaches, they are efficacious on the computational side. Embedded approaches account for the classifier's bias in contradistinction to filter methods.

There are several metaheuristic algorithms suggested in the state-of-the-art, for instance, White Shark Optimizer (WSO) (Braik et al., 2022), Komodo Mlipir (KMA) algorithm (Suyanto et al., 2022), Golden Jackal Optimization (GJO) (Chopra & Ansari, 2022), Artificial Hummingbird (AHA) algorithm (Zhao et al., 2022), Rat Swarm Optimization (RSO) (Dhiman et al., 2021), Chameleon Swarm (CSA) algorithm (Braik, 2021), Archimedes Optimization (AOA) algorithm (Hashim et al., 2021), Gravitational Search (GSA) algorithm (Rashedi et al., 2009), and Reptile Search (RSA) algorithm (Abualigah et al., 2022). The dynamic search behavior and global search capability discriminate these algorithms, by making superior in solving distinct optimization issues, specifically FS problems. In the literature, distinct metaheuristic algorithms were implemented for solving the FS problem. For instance, whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016), Moth-flame optimization (MFO) algorithm (Mirjalili, 2015), Teaching learning-based optimizer (TLBO) (Rao et al., 2011), in addition to Harris hawks optimizer (HHO) (Heidari et al., 2019). Although many researchers have done significant research on promoting the feature selection methodologies, certain drawbacks were found in some areas of these suggested methodologies. Actually, the classical optimization algorithms shows slow converge in the FS area and can easily trap into the best local solutions (Morales-Castañeda et al., 2020). Furthermore, the search strategies for these algorithms shows restricted capabilities, where a gap can be found in switching between the phases of search, which causes imbalanced performance. For this reason, the performance of key feature selection is restricted, and differentiating the relative basic features is challenging. Therefore, there is a recent direction in the state-of-the-art on feature selection that aims to adjust or hybridize the present techniques to promote one technique so as to enhance the search effectiveness of another.

Sooty Tern Optimization Algorithm (STOA) simulates the behaviors of migration and attacking for sooty terns in a real-life (Dhiman & Kaur, 2019); the algorithm has attracted researchers' attention. Initially, STOA was tested over 44 benchmark test functions and six constrained industrial applications. Due to its performance in Dhiman and Kaur (2019), researchers carried it out on several other problems. For example, in Jia et al. (2021), authors introduced a hybrid version that merges the STOA to differential evolution (DE), called STOA-DE. STOA-DE was employed for the FS process to enhance the search efficiency and convergence rate. Singh et al. (2022) employed the STOA in solar power systems to optimize the parameters of a solar cell/module. In Ali et al. (2020), STOA is suggested to design optimal model predictive control (MPC); the method was used to provide the optimal parameters for MPC and decrease the integral time absolute error (ITAE) of the frequencies and tie-line power deviations. Although the STOA has demonstrated satisfactory results, it may still make flaws/weaknesses which MAs may face in general, like local search problem (Jia et al., 2021) and STOA cannot make the balancing between local and global search. From the convergence curves that have been drawn in this paper (see Section 5.4), for the counterparts algorithms, including the STOA, specifically in plots F1, F2 and F9. It is observed that STOA, with a blue marker, gets good fitness values in the early function evaluations (FE) with fast convergence and high exploration. However, the STOA does not evolve over the remaining FE results with a low exploitation rate; it demonstrates that the STOA gets into local regions instead of searching for the global regions, and the exploration and exploitation transition need to be boosted.

Thus, to address the previous shortcomings and improve the use of STOA, this motivates us to introduce a bio-inspired metaheuristic algorithm called the modified Sooty Tern Optimization Algorithm (mSTOA) for solving complex problems. The introduced mSTOA employs three strategies: the exploration/exploitation balance strategy, self-adaptive for control parameters, and population reduction strategy. The performance of mSTOA is assessed on popular benchmark test functions and well-known data sets as a feature selection mechanism. In the mSTOA, migration is indicated as a seasonal movement for sooty terns from existing place to another for finding the most abundant and prosperous food sources to supply adequate energy.

The mSTOA performance is assessed on the benchmark CEC'2020 test suite and nine common FS datasets. Further, the mSTOA is compared against the original algorithm STOA (Dhiman & Kaur, 2019) and some state-of-the-art algorithmic methods, called the improved multi-operator differential evolution algorithm (IMODE) (Sallam et al., 2020) as one of the CEC'2020 competition winners, gravitational search algorithm (GSA) (Rashedi et al., 2009), grey wolf optimizer (GWO) (Mirjalili et al., 2014), harris hawks optimization (HHO) (Heidari et al., 2019), whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016), and slime mould algorithm (SMA) (Li et al., 2020). According to the test results, the mSTOA has high performance compared with other algorithms. In summary, we can formulate the main contributions of the paper in the following:

- The traditional STOA is enhanced by adding the following mechanisms; self-adaptive for control parameters strategy, balancing exploration/exploitation strategy, and population reduction strategy.
- The mSTOA is introduced as an alternative feature selection methodology.
- The mSTOA is suggested to enhance its local search ability and give a solution to the premature convergence issue.
- The proposed mSTOA realized superior results in comparison with its counterparts.
- Distinct metrics for statistical as well as qualitative analyses assess the performance for the suggested mSTOA.

The rest of this paper is presented as following, Section 2 discusses related work, highlighting several recent related works. The basic concepts of the Sooty Tern Optimization Algorithm and Support Vector Machines (SVM) are presented in Section 3. Section 4 demonstrates the proposed mSTOA describing the Sooty Tern Optimization Algorithm with GA, new hybrid algorithm, fitness function, and classification approaches. Section 5 shows the evaluation of the mSTOA in optimization benchmark problems. Section 6 are presented the results of the mSTOA for feature selection. Finally, Section 7 discusses the conclusion and some future directions.

2. Related work

In FS methodologies, it is prospected to get a high accuracy ratio in the classification of features comprised in the dataset. Reducing the number of features represents the main objective, and it assists in decreasing the search space dimensions, which keeps the most relevant features only (Hussien et al., 2017). In this regard, the FS approaches are indicated as wrapper and filter (Pinheiro et al., 2012). Herein, wrapper approaches are employed in identifying the optimal subset of features. The computational time over large datasets represents the main disadvantage of these approaches. In filter methods, it is not essential to use classifiers. The FS involves four steps: (1) selection of the appropriate features. (2) utilizing different metrics for evaluating the subset. (3) identification of another set. (4) feature validation (Houssein, Saber et al., 2022). On the other side, wrapper-based techniques show better results when compared to filter approaches (Hastie et al., 2005). Despite the wrapper approaches for FS demand more time, their results are more accurate (Wang et al., 2018). To avoid these issues, FS is adapted to be improved by making them search for the best subset of features (Wan et al., 2016). The utilization of MAs can be very helpful. These algorithms employ the fitness function to assess the quality of each feature group. Different FS approaches have been applied in the related literature.

FS can be considered an NP-Hard problem because many potential solutions exist, mainly when the feature domain is of high dimensions. To provide a proper solution, a binary version of different metaheuristics methods is proposed; such approaches work as wrapper techniques for FS. Many opportunities have appeared in the FS field for selecting significant features. In Houssein, Emam et al. (2022), Houssein, Helmy et al. (2022) and Houssein, Hassaballah et al. (2022), FS can be broken down into four steps (1) Choose the suitable features, (2) analyze the subset using various metrics, (3) identify other sets, and (4) feature validation. Other FS methods (Cai et al., 2018) include wrapper, filter, and hybrid methods. Nevertheless, the wrapper method produces more accurate results than those of the filter approach, although it is more time-consuming. The hybrid method integrates two methods. The quality of each set of features is evaluated using the fitness function of MAs, which is a complementary part of FS and machine learning algorithms. These suggested systems' success relies on feature relevance to the target domain (Kılıç et al., 2021).

Many metaheuristics algorithms (Hashim et al., 2022; Houssein, Saad et al., 2020) are applied for solving the FS problem, such as the Search and Rescue optimization algorithm (SAR), which is a hybrid with k-NN, and a wrapper FS method proposed in Houssein, Saber et al. (2022). It minimizes the search space size, finds the best subset features, and increases classification accuracy. In the same context, Henry Gas Solubility Optimizer (HGSO) is merged with boosted HHO utilizing Heavy-tailed distributions to improve search space (Abd Elaziz & Yousri, 2021). This approach has been applied to FS problems for some chemical and UCI datasets. In this study, a developed MA technique addresses the FS problem and avoids the demerits of classic FS methods. Thus, more hybrid MAS are increasingly being developed for this purpose and to improve the quality of results (Abd Elaziz & Yousri, 2021).

Recently, the typical MAs such as particle swarm optimizer (PSO) (Gupta & Saini, 2017), cuckoo search (CS) (Rodrigues et al., 2013), bee colony optimizer (BCO) (Hancer et al., 2018), genetic algorithm (GA) (Kennedy & Eberhart, 1997), enhanced multi-operator differential evolution (IMODE) algorithm (Sallam et al., 2020), gravitational search algorithm (GSA) (Rashedi et al., 2009), grey wolf optimizer (GWO) (Mirjalili et al., 2014), HHO (Heidari et al., 2019), whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016), and slime mould algorithm (SMA) (Li et al., 2020) have been employed for FS. These algorithms have iterative searches in a bounded space for optimal solutions. While the classical GA (Kennedy & Eberhart, 1997) relies on natural selection, the CS, BCO, and PSO are obtained from swarm behaviors. In between those, PSO is a classical approach generally utilized for optimization issues because it gives low computational cost, possesses few parameters for tuning, and has a fast convergence. The PSO is also adjusted to operate with multiple sub-swarms. In the binary PSO, indicated as BPSO, the movement of particle is in state space and limited to 0, and 1 in each dimension is suggested. BPSO is utilized in different problems like location facility, vehicle routing problems, and feature selection. In Wang et al. (2007), the authors presented a modified version from the BPSO based on the operator of the mutation along with genotype–phenotype representation. On the other side, FS and the parameters of SVM kernel (Huang & Dun, 2008) are improved simultaneously by PSO.

The application of the MAs combined with machine learning tools is so common (Hashim et al., 2019; Houssein, Helmy et al., 2021). They are employed in chemical compound design. Herein, the testing is carried out using 80 train samples besides to 20 test samples because the chemical information demands more features. Furthermore, the algorithms of machine learning are implemented in drug design. The commonly utilized approaches include support vector machines, the wisdom of crow, neural networks, k-NN, and deep learning. A popular application is to predict the molecular compounds' properties relevant to virtual screening. The instance that demonstrates the possibility of using machine learning as a classification tool for many problems in cheminformatics is presented in Houssein, Neggaz et al. (2021). FS and support vector machine (SVM) kernel parameters are optimized simultaneously (Huang & Dun, 2008). FS has been widely used in various applications as Bioinformatics has along with history with sequence analysis. Content and signal analysis are two sorts of challenges that can be categorized in FS. The content analysis examines a sequence's general properties, such as its propensity to code for proteins or its ability to perform a specific biological function. In contrast, the signal analysis identifies the sequence's key motifs as regulatory or gene structural elements.

The work presented in Abd Elaziz et al. (2020) shows the utilization of feature selection methodologies for selecting the essential drug descriptors. Here, FS has deemed as a problem of multi-objective optimization involving two conflicting goals. Minimizing selected feature numbers and maximizing the dependency degree for the descriptor. In Tharwat et al. (2016), it is declared how the FS methods are utilized in drug development. For the FS phase, the features used are discriminated based on rough set-based methods. At this step, three distinct rough set-based approaches were employed to select fewer features from the feature vector: DMFS, QRFS and EBFS. The objective of employing these algorithmic methods is to diminish the entire number of features that lead to a reduction of the classification time besides improving the classification performance. Another example is presented in Abd Elaziz and Yousri (2021), where FS approaches are presented and applied to multiple domains primarily to treat the data of high dimensions. There are diverse FS methods based upon metaheuristic techniques which have been presented to tackle the issue of FS and keep away from the drawbacks of conventional FS methods. Feature selection represents an important complementary part of the machine learning systems, where the most successful ML approaches highly rely on the features employed in training.

On the other hand, in Kılıç et al. (2021), FS can be considered an NP-Hard problem because many potential solutions exist, mainly when the feature domain is high dimensional. Under this assumption, different methods have been suggested using optimization algorithms as metaheuristics. For instance, a multi-population particle swarm optimizer (MPPSO) is suggested for feature selection. In Neggaz et al. (2020), the authors mentioned that FS is an essential preprocessing step that helps avoid the advert influence of noisy, misleading, besides inconsistent features. The proposed method employs a metaheuristic algorithm (MA) for choosing the prominent features so that they can simplify and improve the quality of the datasets with high dimensions, to devise an effective knowledge extraction system. Nevertheless, such approaches often lack the local optimally issue when applied to datasets with massively big feature-size due to the considerably large solution domain. In Hashim et al. (2019), a novel methodology for dimensions reduction by utilizing Henry gas solubility optimizer (HGSO) to select the valuable features to improve classification accuracy.

In Hashim et al. (2020), the authors explain that using classifiers in the filter procedure is unnecessary. In the works presented in Hastie et al. (2005) and Wan et al. (2016), it has been shown that wrapper-based solutions outperform filter approaches with regard to the accuracy of results. Although the wrapper approaches for FS are more time-consuming, they yield more precise results. FS may be tweaked to be more efficient by identifying the best subset of attributes to tackle a range of problems. MAs have many advantages; when they are used in FS, the fitness function assesses the quality of every set of features. Since learning algorithms are involved in the FS process, wrapper-based techniques have attracted significant attention (Cai et al., 2018). Therefore, the performance of learning algorithms influences the selection of significant features (e.g., a ratio of correct classification accuracy).

In Hussien et al. (2017), a swarm-based approach utilizing wrapper FS has been presented to estimate chemical compound activity. The optimal subset from the molecular descriptor across the MAO dataset is chosen using the salp swarm algorithm (SSA). The SSA is compared with other MA, including the moth-flame optimization algorithm (MFO), grasshopper optimization algorithm (GOA), and sine–cosine algorithm (SCA). It is worth noting that SSA, besides the k-NN classifier, had the best accuracy of 87.35% while keeping 783 chemical descriptors. In Houssein, Hosney et al. (2020), two classification methodologies, namely, HHO-SVM, besides HHO-kNN, which is based on the Harris hawks optimizer, have been suggested for the prediction of the design and discovery of the drug.

In the same context, several techniques for FS are explained; for instance, in Abd Elaziz et al. (2020), the strategies for selecting medications based on their features and the importance of chemical descriptors have been presented. FS can be deemed a problem of multi-objective optimization that reflects two competing goals: decreasing the number of features selected and increasing the descriptor dependency degree. In Liu et al. (2015), many feature extraction approaches have been presented. To test their prediction performance. During the testing step, it will be necessary to specify the facial picture view and the recorded eye-gaze locations.

Moreover, the HHO algorithm with k-NN and SVM using the wrapper FS method proposed the best result as in Houssein, Hosney et al. (2020). It has some disadvantages in achieving the balance between global and local solutions, so a new version of this algorithm with genetic operator (Houssein, Neggaz et al., 2021). Also, the Salp swarm optimization algorithm is proposed with the same idea, but it sometimes cannot achieve the requested balance for the solution (Hussien et al., 2017). To sum up the existing studies in the state-of-the-art, Table 1 demonstrates some metaheuristic algorithms besides to machine learning techniques utilized in Cheminformatics for the design and discovery of drugs.

3. Preliminaries

In the subsequent subsections, we will explain the basics of the methods that have been used in this paper.

3.1. The algorithm of Sooty Tern Optimization

This section details the mathematical model behind the implemented algorithms, involving the algorithm of Sooty Tern Optimization, indicated as STOA (Dhiman & Kaur, 2019). Sooty terns, named scientifically *Onychoprion fuscatus*, represent sea birds that exist all over the world. A vast range of sooty terns comprises species that have distinct sizes and masses. The sooty tern is described as an omnivorous bird that eats insects, reptiles, fishes, amphibians, earthworms, and so on. In order to attract earthworms kept out of sight under the ground, the sooty tern produces a rain-like sound made with its feet, and it also uses bread crumbs to attract fishes. Sooty terns generally live in colonies and use their intelligence in finding and attacking their prey. The most eminent things about sooty terns are their attacking and migrating behaviors. Migration is indicated as a seasonal movement for sooty tern from an existing place to another so as to reach the most abundant and prosperous food sources that can supply adequate energy. Such behavior is characterized by the following. Through the migration, sooty terns are grouped together to travel. The initial positions from them are different for the avoidance of collisions among each other. In the grouping, sooty tern travels in the direction of the fittest sooty tern is indicated as the best survival. In other words, the sooty tern to lower fitness value than others. According to the fittest found sooty tern, the initial positions from the other sooty terns can be updated. A flapping mode in flight is used by sooty terns when attacking in the air. Such behaviors are formulated in a way associated with the objective function to be optimized. The rest of the section explains the mathematical model for the STOA operators.

3.1.1. Migration behavior (exploration)

Collision avoidance. Here, S_A is utilized in the computation of positions from the new search agent for the avoidance of collision between its adjoining search agents (ex., the sooty terns). This process is performed as follows:

$$\vec{C}_{st} = S_A \times \vec{P}_{st}(z) \tag{1}$$

where $\vec{C}_{st}$ is the position of a search agent which have not collided with any other search agent. $\vec{P}_{st}(z)$ indicates the current position for the search agent. z denotes the current iteration, whereas S_A refers to the movement of the search agent within a search space, and it is computed as:

$$S_A = C_f - \left(z \times \left(C_f / \mathrm{MAX}_{\mathrm{FEs}}\right)\right) \tag{2}$$

where $z = 0, 1, 2, \ldots, MAX_{\mathrm{FEs}}$. MAX_{FEs} is the maximum number for function evaluations, C_f is a control variable that adjusts the S_A that decreases linearly from C_f till zero. The value of the variable C_f is assigned as 2.

Converge toward the best neighbor's direction. Search agents converge toward the optimal neighbor's direction after collision avoidance. This process is conducted by using Eq. (3).

$$\vec{M}_{st} = C_B \times \left(\overrightarrow{P_{bst}}(z) - \vec{P}_{st}(z)\right) \tag{3}$$

where $\vec{M}_{st}$ refers the different locations for search agent, $\vec{P}_{st}$ towards the optimal fittest search agent $\overrightarrow{P_{bst}}$. C_B represents a random variable that is in charge of a better exploration and it is computed as follows:

$$C_B = 0.5 \times rand \tag{4}$$

where rand refers to a random number that lies between the $[0, 1]$ range.

Table 1
Extensive review on several proposed approaches.

Ref	Year	Method	Dataset	Results
(Zainudin et al., 2017)	2017	Filter-based feature selection approach as well as other medical benchmarks that integrated the relief-f to differential evolution (DE)for selecting the features of the highest relevancy.	QSAR Biodegradation	An accuracy of 85.4% where there were only 16 relevant molecular out of 41 features.
(Hussien et al., 2017)	2017	Wrapper feature selection for predicting chemical compound activity (CCA)	MAO	The highest classification accuracy was 87.35% by only retaining a number of 783 molecular descriptors (MDs) out of 1665 features and introducing the SSA that employed k-NN as classifier.
(Houssein, Hosney et al., 2020)	2020	The HHO-SVM and HHO-kNN were utilized as classification methods for predicting the design and discovery of drugs	MAO and QSAR Biodegradation	For the HHO-k-NN, the best fitness function values were 97.599% and 84.523% over the MAO and QSAR Biodegradation datasets, whereas the HHO-SVM achieved 97.583% and 85.023% over the same datasets, respectively.
(Martínez et al., 2019)	2019	Multi-objective optimization algorithm based FS was suggested for selecting molecular descriptors over the QSAR Biodegradation	QSAR Biodegradation	An accuracy of 84% and selection ratio of 37% were reported as the highest performance over QSAR Biodegradation dataset.
(Martínez et al., 2018)	2018	A Bi clustering based method that aims to decrease number of molecular descriptors (MD) for the prediction of chemical compound Biodegradation where three classifiers were tested, including, Neural Network (NN), Random Committee (RC), and Random Forest(RF)	QSAR Biodegradation	The classifier of the best accuracy was RF which realized 88.81% for only nineteen MD
(Putra et al., 2019)	2019	An integration between NN and SVM was suggested for QSAR modeling	QSAR Biodegradation	A correct classification rate of 82% was reported.
(Dutta et al., 2019)	2018	An algorithm namely HGSE that utilizes stochastic graphlet embedding algorithm (SGE) over several hierarchical configurations to deal with molecular graph dataset	MAO	With accuracy 95.71%
(Goh et al., 2018a)	2018	An old neuronal architecture (Multi-layer perceptron) was fused with recent architecture through the deep learning for predicting chemical activity. Two models including DeepBioD+ and DeepBioD were applied to QSAR Biodegradation dataset	QSAR Biodegradation	The accuracy results were 90% and 87.5% for DeepBioD+ and DeepBioD, respectively.
(Goh et al., 2018b)	2018	Deep learning model was implemented for predicting chemical activity	QSAR Biodegradation	An accuracy of 86.7% was reported
(Atwood & Towsley, 2016)	2016	A graph-structured based data model was produced by carrying out a representational deep learning architecture namely diffusion CNN.	MAO	An accuracy rate of 75.14% was obtained.

Updating corresponding to the optimal search agent: Eventually, a search agent (ex., sooty tern) updates its own position depending on the taken optimal search agent as follows:

$$\vec{D}_{st} = \vec{C}_{st} + \vec{M}_{st} \tag{5}$$

where $\vec{D}_{st}$ is employed to define the gap between both search agent and optimal fittest search agent.

3.1.2. Attacking a behavior (exploitation)

Sooty terns are able to modify the velocity and angle of attack; to perform this change, they move the wing to increase the altitude. To attack the prey, they move in a spiral in the air; this process is described as:

$$x' = R_{\text{adius}} \times \sin(i) \tag{6}$$

$$y' = R_{\text{adius}} \times \cos(i) \tag{7}$$

$$z' = R_{\text{adius}} \times i \tag{8}$$

$$r = u \times e^{kv} \tag{9}$$

R_{adius} indicates the radius for each turn in the spiral, i denotes a variable from the $[0 \leq k \leq 2\pi]$ range. While u and z help in defining the spiral, e represents the base for the natural logarithm. The constants are set as $u = 1$ and $v = 1$. The new positions are computing using as a base Eqs. (6)–(9). The next equation describes the procedure to compute the new positions.

$$\vec{P}_{\text{st}}(z) = \left(\vec{D}_{st} x \left(x' + y' + z'\right)\right) \cdot \vec{P}^{-}_{\text{bst}}(z) \tag{10}$$

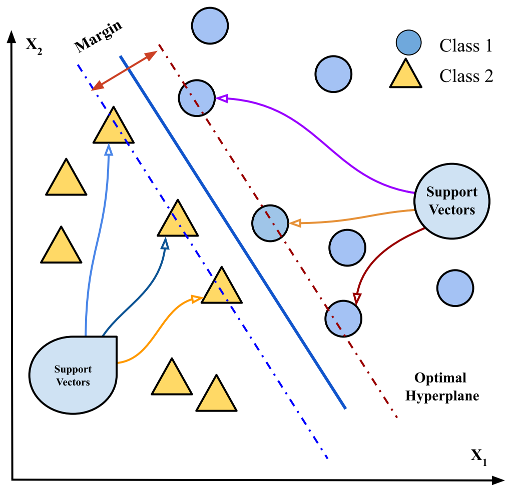


Fig. 1. Classification task by SVM.

Where $\vec{P}_{st}(z)$ updates the positions from other search agents and then retains the optimal solution.

3.2. Support vector machines

Support Vector Machines, abbreviated as SVMs, represent a popular machine learning method (Rodriguez-Perez et al., 2017). It is based on mapping data by using kernel functions. The main purpose is to get the optimal solution. An SVM provides a linear model for several classifications or regression problems.

SVM indicates an extensively used machine-learning methodology in Cheminformatics. One of the applications of SVM is the prediction of toxicity-related qualities such as mutagenic toxicity, HERF blocking, toxicity categorization, and phospholipidosis toxicity. Algorithm 1 shows the pseudo-code for SVM 1 while Fig. 1 provides a graphical explanation.

Algorithm 1 A SVM Algorithm Pseudo-code

Inputs: Loading the data of training and testing.
Outputs:Calculating accuracy.
Select a value for cost C and Γ of SVM.
while (a stopping criterion is not reached) **do**
    Execute train step for SVM on every data point.
    Execute classify step for SVM on every test data point.
end while
Return the *accuracy*

4. The proposed mSTOA

This section explains the suggested mSTOA in detail. The mSTOA has been proposed to overcome the STOA's drawbacks. The shortcomings rely on the problem that is solved and involve: (1) slow convergence, (2) being trapped with sub-optimal regions, together with (3) an improper equilibrium among exploration and exploitation phases. The principle steps of mSTOA are formulated as follows:

- Initialization stage: The mSTOA starts the optimization operation with a population of agents initialized randomly using uniform distribution as follows:

$$\mathrm{P}_{\mathrm{st}}^{initial} = LB_i + \mathrm{rand}_i\left(UB_i - LB_i\right) \quad i = 1, 2, \ldots, NP \tag{11}$$

where X_i^{initial} refers to random initialized ith solution vector, UB_i, LB_i refers upper and lower bounds for the ith solution respectively; further the bounds UB_i, LB_i are problem dependents, for the ith portion of the solution, NP indicates the population size while $\mathrm{rand}_i \in [0, 1]$ symbolizes random value.
- Update stage: A traditional STOA is enhanced by comprising the subsequent mechanisms:

  1. Balancing exploration/exploitation strategy.
  2. Self-adaptive for control parameters strategy.
  3. Population reduction strategy.

The steps of the suggested mSTOA methodologies are detailed in the subsequent subsections.

4.1. Strategy for balancing exploration/exploitation

The proposed mSTOA adopts two different exploration strategies; the main idea of these strategies is adopted from Houssein, Çelik et al. (2022) to boost the introduced mSTOA performance.

The introduced strategies retain both diversity and convergence in the evolutionary process. Specifically, the first method reduces the population diversity and further concentrates on exploiting the search space nearer population best solutions within the population in the traditional STOA. The second moderate strategy targets to enhance the search process through a combination of the stochastic nature represented in a stochastic selection and the local random walk for directing the search operation to exploit a neighborhood for the current solution. Further, a newly proposed method called the self-adaptive control method is applied to the two inherited strategies. It adjusts the adaptive parameters and guarantees a soft balance and transition between the exploration/exploitation stages, considering the various search space.

- Stochastic moderate exploration/exploitation strategy:
This strategy is inherited from the study in Das et al. (2009). It targets to enhance the search process through a combination of the stochastic nature represented in a stochastic selection and the local random walk for directing the search operation to exploit a neighborhood for the current solution. The moderate strategy has two terms; the distance between two randomly selected space solutions (exploration), the distance between the current taken solution, and the optimal one in the population (exploitation). Additionally, the greedy method called "DE/target-to-best/1/bin" scheme is applied for the mutation target and provided in Eq. (12). Where u_j is the j^th portion of the newly generated solution, $X_{i,j}$ indicates the j^th portion of the current i^th solution. F1 and F2 refer to the scaling factors, $P_{st}(j)$ refers to the j^th position of the best solution, and $X_{r1} - X_{r2}$ are two solutions chosen randomly from between all population's solutions.

$$u_j = X_{i,j} + F1 \times (X_{i,j} - \vec{P}_{st}(j)) + F2 \times (X_{r1} - X_{r2}). \tag{12}$$

- Stochastic short exploration strategy:
This strategy aims to reduce population diversity. It simulates the idea in Črepinšek and Mernik (2013), further concentrating on exploiting the search space nearer population best solutions within the population in the traditional STOA. The short-term exploration strategy is the attacking behavior operation in Eq. (10) in the original STOA.
Furthermore, the selection method is based on the adaptive crossover rate (CR) operator. Thus, this strategy represents a moderate strategy that works between the exploration side for Eq. (12) and the exploitation side for Eq. (10) to achieve the equilibrium in the exploration/exploitation phases of the algorithm as follows:

$$Vec_j = \begin{cases} u_j & \text{if } \left(CR \geq rand \ or \ j = j_{rand}\right) \\ w_j & otherwise \end{cases} \tag{13}$$

where Vec_j is the moderate distance generated solution, u_j is the mutated target. j symbolizes the j^th part in the solution dimension, j_{rand} refers to a discrete number produced randomly $\in [1, D]$ where D symbolizes the dimension from a solution. The $j = j_{rand}$ guarantee that the condition is correct at least once, i.e., the mutation is performed at least one time, the w_j represents the stochastic short exploration strategy and computed as follows:

$$w_j = \begin{cases} \vec{P}_{st}(z) \ in\ Eq.\ (10) & \text{if } (t \le rand) \\ X_{i,j} & otherwise \end{cases} \tag{14}$$

where $\vec{P}_{st}(j)$ represents the exploitation walk of solutions and explained in Eq. (10), $X_{i,j}$ refers to the $j^t h$ portion of the current $i^t h$ solution, rand symbolizes a random number generated across the range [0,1], the t operator control the algorithm walk either direct the solution to exploit around the promising regions in the last iteration or maintain the solution useful information through the early iterations, the t is adjusted as follows:

$$t = \left(1 - \frac{FE}{MAX\_FEs}\right) e^{\left(a2 * \left(\frac{FE}{MAX\_FEs}\right)\right)} \tag{15}$$

where FE symbolizes the current number for function evaluations, while MAX_FEs indicates the maximum number for function evaluations, $a2$ is a random number produced by:

$$a2 = \begin{cases} rand & \text{if } (rand \le 1/2) \\ 1, & otherwise \end{cases} \tag{16}$$

where rand symbolizes a random number generated over the range [0,1].

4.2. Self-adaptation of the control parameters in the mSTOA

The control parameters significantly affect optimization algorithms' performance, especially in the equilibrium between exploration/exploitation processes (Houssein, Rezk et al., 2022; Oliva et al., 2020). To set the values of these parameters, a self-adaptive procedure is required (Houssein, Çelik et al., 2022). The best configuration of the control variables is unknown and can be tuned for each optimization problem (Eiben & Schoenauer, 2002). The proposed mSTOA introduces a self-adaptation of control parameters method. In these phases, stochastic moderate exploration/exploitation strategy, crossover rate CR, and scaling factors ($F1$, $F2$) are updated adaptively to guarantee a soft balance and transition among the exploration/exploitation stages considering the various optimization problems' search space.

$$CR = rand \tag{17}$$

The CR is updated with Eq. (17), in case the newly produced solution is more optimal than the other current solutions as depicted in Algorithm 2. Otherwise, the CR parameter is updated based on Eq. (18) as follows:

$$CR = rand \times (0.1 - 0.05) + 0.05 \tag{18}$$

The scaling factors F1 and F2 are generated as follows:

$$F1 = \cos t + (rand - 0.5) \tag{19}$$

$$F2 = \sin t + (rand - 0.5) \tag{20}$$

where $rand$ indicates uniform distributed random number $\in [0, 1]$, t is computed by Eq. (15).

4.3. Population reduction feature

The number of search agents NP in population-based algorithms is vital in adjusting the algorithm convergence rate (Morales-Castañeda et al., 2021). Further explaining, on the one hand, the small size of population agents can converge rapidly but, in contrast, raise the probability of getting into a local optimum. Furthermore, the population with a large agent size converges slower but provides a better exploration of the search space. The proposed mSTOA applies linear reduction of the population method as follows:

$$NP^{(t+1)} = \text{round}\left[\left(\frac{NP_{\max} - NP_{\min}}{MAX\_FEs}\right) * FE + NP_{\min}\right] \tag{21}$$

where NP_{max} refers size for the initial population (NP), NP_{min} symbolizes the specified minimum population size, $NP_{min} = 16$ in current study, FE indicates the current number of function evaluations whereas MAX_FEs represents the maximum number for function evaluations.

The mSTOA optimization process steps are formulated in Algorithm 2. Furthermore, detailed steps for mSTOA processes exhibited are provided in the flowchart of Fig. 2.

4.4. Time complexity

A time complexity for the mSTOA is concentrated fundamentally on the operation of evolving the optimization solution through the position update. Thus, it can be expressed as in the following:

$$O(mSTOA) = (O(\text{ mSTOA position update })) \tag{22}$$

$$O(mSTOA) = O(Moderate\_distance\_strategy) + O(Short\_distance\_strategy) \tag{23}$$

$$O(mSTOA) = O(NP \times MAX\_FEs) \tag{24}$$

where $Moderate\_distance\_strategy$ refers to the moderate update strategy proposed in (12),

$Short\_distance\_strategy$ refers to the short update strategy proposed in (10), NP is the population size, MAX_FEs symbolizes the maximum number for function evaluations. It is worthy attention that the big-O from both STOA and mSTOA is the same. Hence, the developed approach is found to be competitive with the standard STOA with regard to the big-O.

5. Experimental series 1: Applying mSTOA in solving CEC'20 test suit

The experiment utilizes functions from the CEC'2020 benchmark to assess the proposed mSTOA performance, involving quantitative and qualitative measures. The quantitative measures used are: (1) the mean and (2) the standard deviation (STD) for the optimal solutions provided by all the algorithms utilized in the comparison. Furthermore, the qualitative metrics encompass (1) search history, (2) average fitness history, together with (3) optimization history. To realize fair comparison assessment, the mSTOA results were compared to other seven MAs, namely Improved Multi-operator Differential Evolution Algorithm (IMODE) (Sallam et al., 2020), Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), Grey wolf optimization (GWO) (Mirjalili et al., 2014), Harris hawks optimizer (HHO) (Heidari et al., 2019), whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016), Slime mould algorithm (SMA) (Li et al., 2020) and original STOA. For all investigated algorithms, population size along with maximum number for function evaluations (FEs) have been chosen as 30 and 10,000, respectively. Besides that, each algorithm is executed in 30 runs for every function, eventually, the results are taken up to the average performance for these runs with $Dim = 10$. Table 2 gives the parameter settings applied to every algorithm.

Instead of employing the default parameter values, the Taguchi tuning method is utilized to get the proper parameter values. The comparative algorithms' parameters are tuned firstly to get the convenient parameters' settings providing the best performance. The Taguchi (Byrne, 1986) method depends on the Orthogonal Array (OA) along with the mean analysis, so as to assess the impact of an algorithm-tuned

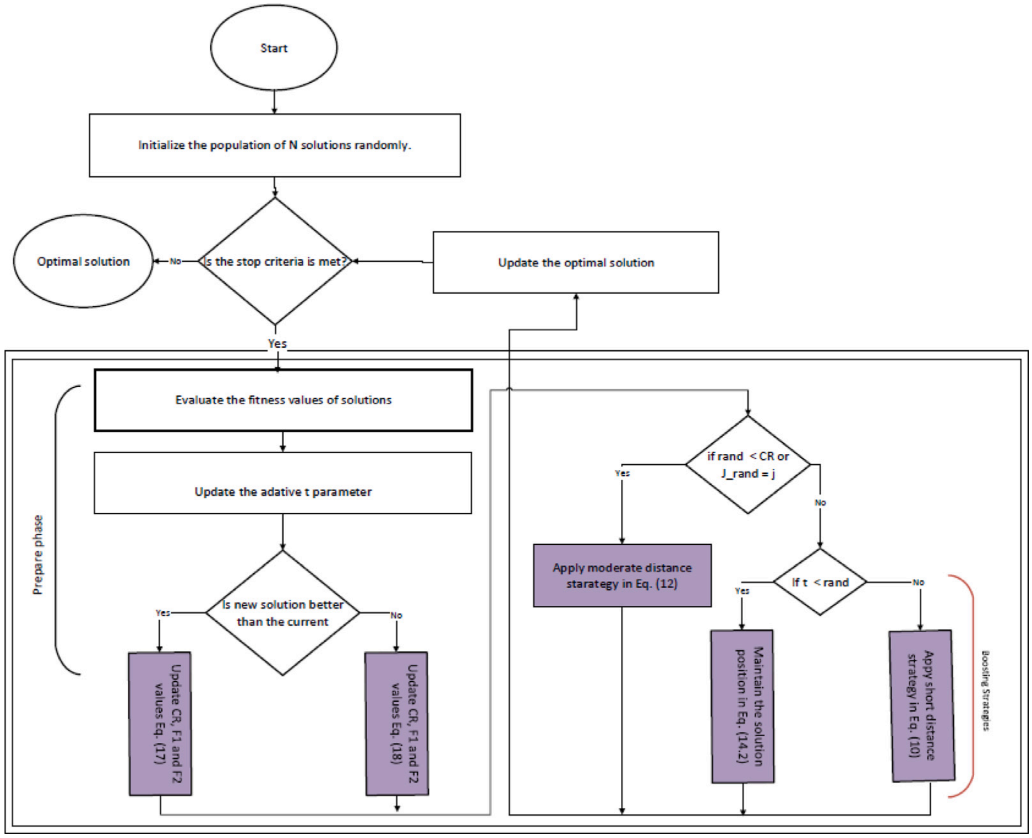


Fig. 2. The flowchart of the proposed mSTOA.

Table 2
Parameter settings in counterpart algorithms.

Algorithm	Parameters settings
Common settings	The Population size: $NP = 30$ Maximum function evaluations : $MAX_{FEs} = 30000$ Dimensions of problem $Dim = 10$ Independent runs number: 30
IMODE	$arch\_rate = 2.6$
GSA	$G0 = 100, alpha = 20,$ $Rpower = 1, Rnorm = 2$
GWO	a linearly decreases from 2.3 till 0
HHO	$E0 = 1.67,\ E1 = 1,\ beta = 1.5$
WOA	$\alpha = 1$
SMA	$z = 0.13$
STOA	$C_f = 0$

parameters regarding statistical analysis of the conducted experiments. Further, the OA represents fractional factorial matrix from the numbers organized so that every row refers to the factors' level at each run, while the columns represent the factors changed from each run. Table 2 illustrates the best parameters setting in each algorithm that have been obtained using the Taguchi tuning method .

5.1. Description of the function set for CEC'2020 benchmark

This assessment process for the new version of the Sooty Tern Optimization Algorithm (STOA), called mSTOA, is discussed in this section. Consequently, the test problems were chosen from the IEEE Congress on an Evolutionary Computation (CEC) (Mohamed et al., 2020) to measure the performance shown by the suggested algorithms. First of all, the function set for the CEC'2020 benchmark involves 10 test functions involving (1) unimodal, (2) multimodal, and (3) hybrid, besides (4) composition functions. In this regard, the mathematical formulation and the attributes for this benchmark test are shown in Table 3; 'Fi*' indicates the value of optimum global with $Dim = 10$.

Fig. 3 gives a 3D visualization for the functions in the CEC'2020 to facilitate understanding of the nature and differences of each problem.

5.2. The statistical result analysis

Table 4 gives the mean besides to (standard deviation) STD for the best values taken from the suggested algorithm and the rest of the method for every CEC'2020 benchmark function, of $Dim = 10$, the optimal results (the minimum values) are highlighted in bold.

According to the uni-modal test function F1, it has a uni-modal space with global region to test the optimizer exploration, the mSTOA and IMODE reached the optimal solution with a stable performance reflected on the mean and STD metrics, these measures prove the convenient way of applying the moderate-distance-strategy with the adaptive CR parameter, on the other side, the traditional STOA and GWo show the worst mean and STD values, over the testing methods of the F2-F4 functions in multi-modal spaces, these functions types have a global and local regions, such that to test the algorithm exploration and exploitation, the proposed mSTOA exhibits a comparative performance compared to the IMODE (CEC'2020 competition winner) algorithm, this behavior reflects the effective local-optima avoidance obtained after applying the diversity maintain in population size reduction method and exploration/exploitation balance in Eq. (13), while the HHO and GSA algorithms got a limited performance, the hybrid functions include F5, F6 and F7, the proposed mSTOA obtain a better performance after the IMODE algorithm; on contrary, the GSA and WOA algorithms show a limited performance compared to the remaining optimizers, on the last composition functions symbolized as F8, F9 and F10, the proposed mSTOA obtain near-optimal solution, however the IMODE algorithm

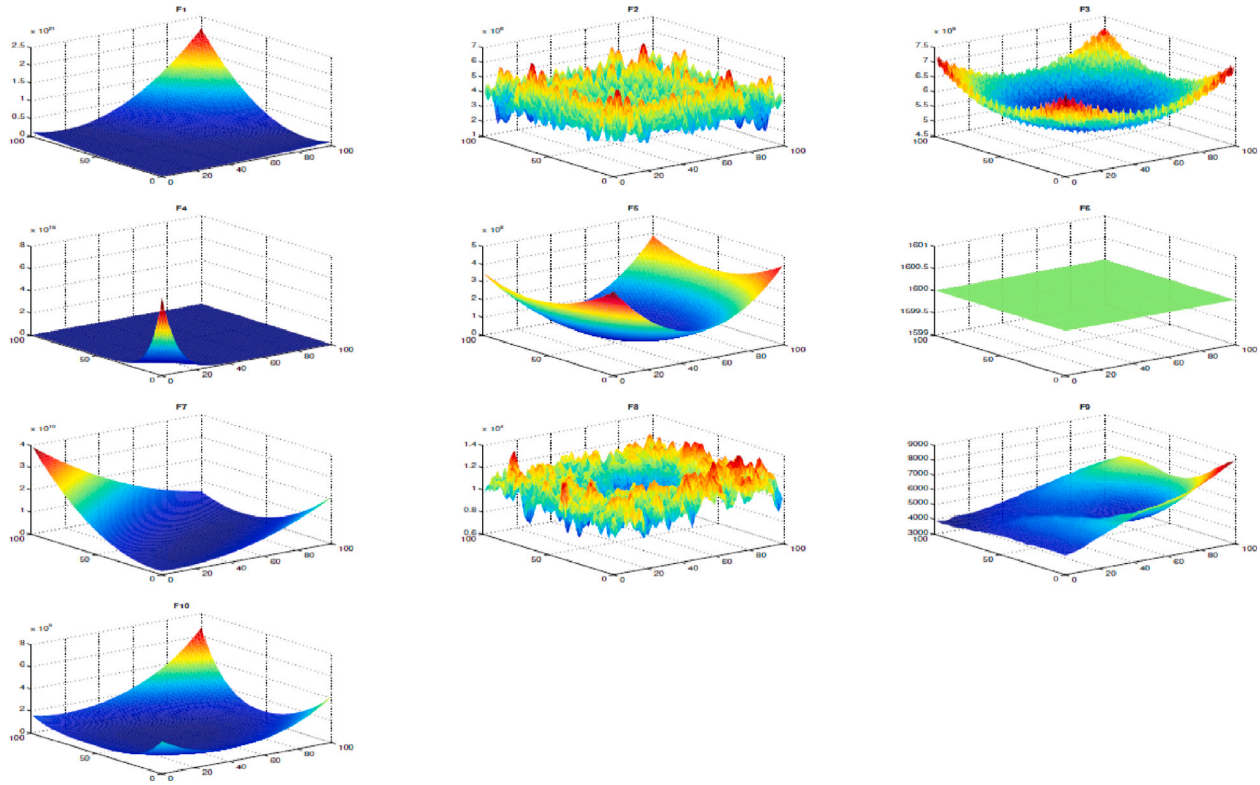


Fig. 3. A 3D visualization for the functions of CEC'2020 benchmark.

Table 3
Summary on the function set for CEC'2020 benchmark.

No.	Description of function	Fi*
Unimodal function		
F1	Shifted and Rotated Bent Cigar Function(CEC2017 F1)	100
Multimodal functions		
F2	Shifted and Rotated Schwefel's Function(CEC2014 F11)	1100
F3	Shifted and Rotated Lunacek bi-Rastrigin Function(CEC2017 F7)	700
F4	Expanded Rosenbrock's plus Griewangk's Function(CEC2017 F19)	1900
Hybrid functions		
F5	Hybrid Function1 (N = 3)3)(CEC2014 F17)	1700
F6	Hybrid Function2 (N = 4)(CEC2017 F16)	1600
F7	Hybrid Function3 (N = 5)(CEC2014 F21)	2100
Composition functions		
F8	Composition Function1 (N = 3)(CEC2017 F22)	2200
F9	Composition Function2 (N = 4)(CEC2017 F24)	2400
F10	Composition Function3 (N = 5)(CEC2017 F25)	2500

show better stability than mSTOA on F9; additionally the superior performance on F10 is assigned to the mSTOA algorithm.

With respect to mean and STD, the results reflected that the suggested algorithm is actually superior in solving six functions from the CEC'2020 benchmark in comparison to the other competitors. Furthermore, mSTOA took the first rank with regard to Friedman's average rank-sum test.

5.3. Boxplot behavior analysis

The boxplot analysis can be utilized in displaying data distribution characteristics, as too many local minima are associated with that class of functions. So as to realize a better understanding of the distribution of the results, Fig. 4 is employed to view the boxplot of results from every algorithm and function. Boxplots are tools used for depicting the distributions of data into quartiles. A maximum and minimum indicate the lowest and largest data points the algorithm reaches, representing the edges for the whiskers. In this context, the upper and lower quartile are delimited using the rectangles' ends. Also, a narrow boxplot indicates a high agreement among data. Fig. 4 reveals the results of boxplot for ten functions with $Dim = 10$. Additionally, boxplots for the proposed mSTOA are very narrow, in most functions, in comparison to the distribution of other algorithms and, therefore, with the most lower values. The proposed mSTOA acts better than the rest competitors on the majority of test functions while yielding limited performance on only F1 and F7.

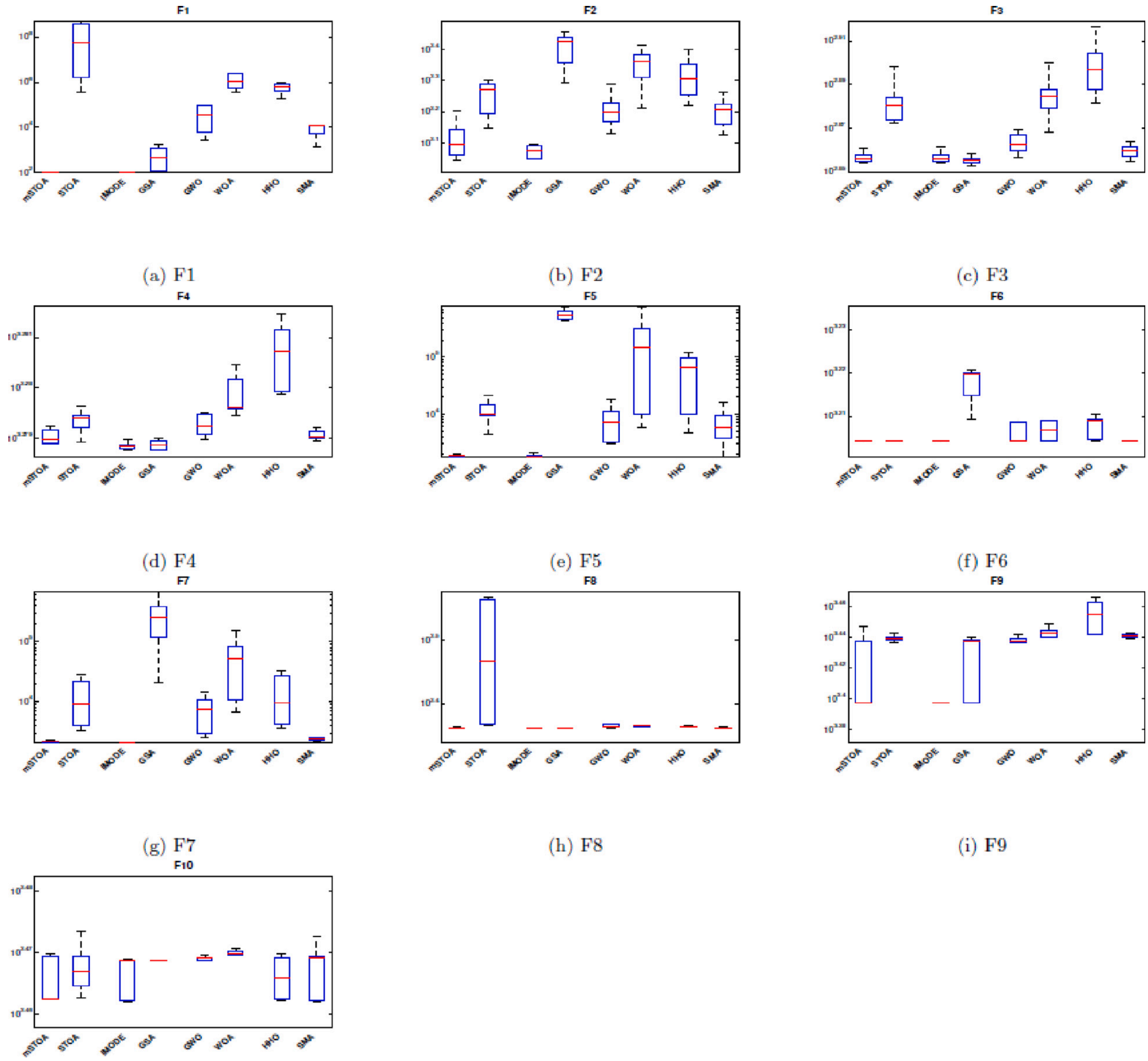
5.4. Analyses on convergence behavior

Convergence analysis for the suggested mSTOA is presented in this subsection in contrast to the other competitive algorithms. Fig. 5 depicts the convergence curves for the proposed mSTOA along with counterparts for the functions of CEC'2020. In most functions, it is noted that the proposed mSTOA succeeded in reaching a stable point.

In Fig. 5, the convergence curves are depicted for F1 test function in a uni-modal space. The mSTOA algorithm achieves an early exploration more clearly than the IMODE, GSA, and STOA algorithms. The mSTOA convergence follows a degradation phase in the middle

Table 4
The calculated mean and STD from fitness values on 30 runs taken by other competitor algorithms by using the functions of CEC'2020 ($Dim = 10$).

Functions	IMODE		GWO		GSA		WOA		HHO		SMA		STOA		mSTOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	1.00E+02	1.77E−14	3.80E+07	1.08E+08	6.35E+02	6.10E+02	2.75E+06	5.04E+06	6.14E+05	2.58E+05	9.34E+03	3.93E+03	1.66E+08	1.92E+08	1.00E+02	2.71E−05
F2	1.18E+03	5.90E+01	1.60E+03	1.74E+02	2.54E+03	3.12E+02	2.26E+03	3.85E+02	2.03E+03	2.69E+02	1.58E+03	1.58E+02	1.78E+03	2.10E+02	1.30E+03	1.78E+02
F3	7.19E+02	4.08E+00	7.32E+02	1.27E+01	7.16E+02	2.81E+00	7.66E+02	1.47E+01	7.89E+02	1.93E+01	7.24E+02	4.89E+00	7.61E+02	1.45E+01	7.18E+02	3.42E+00
F4	1.90E+03	2.73E−01	1.90E+03	2.46E+00	1.90E+03	4.47E−01	1.91E+03	2.77E+00	1.91E+03	2.75E+00	1.90E+03	3.95E−01	1.90E+03	9.36E−01	1.90E+03	5.68E−01
F5	1.80E+03	1.01E+02	8.17E+03	5.51E+03	5.62E+05	1.10E+05	2.15E+05	2.38E+05	6.00E+04	4.42E+04	7.14E+03	5.13E+03	1.12E+04	5.26E+03	1.82E+03	5.64E+01
F6	1.60E+03	2.68E−01	1.61E+03	2.41E+01	1.66E+03	2.56E+01	1.61E+03	8.51E+00	1.61E+03	9.13E+00	1.60E+03	3.08E−01	1.60E+03	2.67E−01	1.60E+03	2.15E−01
F7	2.11E+03	8.73E+00	8.00E+03	4.41E+03	2.64E+05	2.03E+05	5.66E+04	5.03E+04	2.03E+04	2.66E+04	2.52E+03	3.31E+02	1.21E+04	9.56E+03	2.19E+03	8.79E+01
F8	2.30E+03	2.30E+03	2.31E+03	2.30E+03	2.30E+03	2.30E+03	2.37E+03	2.31E+03	2.31E+03	2.31E+03	2.30E+03	2.25E+03	2.99E+03	2.32E+03	2.30E+03	2.30E+03
F9	2.54E+03	1.09E+02	2.74E+03	1.15E+01	2.65E+03	1.27E+02	2.76E+03	5.75E+01	2.81E+03	1.24E+02	2.76E+03	6.12E+00	2.75E+03	1.06E+01	2.58E+03	1.31E+02
F10	2.93E+03	2.29E+01	2.94E+03	1.49E+01	2.94E+03	1.38E+01	2.95E+03	1.18E+01	2.92E+03	2.44E+01	2.93E+03	2.61E+01	2.94E+03	3.77E+01	2.91E+03	2.31E+01
Friedman mean rank	4.5		2.3		6.8		4.7		6.8		6.2		3.3		1.4	
Rank	4		2		7		5		8		6		3		1	



(a) F1
(b) F2
(c) F3
(d) F4
(e) F5
(f) F6
(g) F7
(h) F8
(i) F9

Fig. 4. The resulted boxplot curves for the suggested mSTOA and the competitors taken by the testing functions involved in CEC'2020, with $Dim = 10$.

Table 5
The description of the used datasets.

Dataset	No of features	No of instance
baseBrainT21	326	73
baseBrainT91	5527	60
Lymphography	18	148
HeartEW	19	270
ZOO	16	101
vote	16	300
ND	326	103
WineEW	13	178
Mofn	13	1000

iterations as a balance exploration/exploitation middle phase. Over the final iterations, the mSTOA shows comparative exploitation regarding the other algorithms. In contrast, the STOA, GWO, and WOA algorithms show the worst convergence rates. This performance is due to applying adaptive parameters to balance short (exploration) and long (exploitation) walks in the proposed mSTOA. The convergence over the F2-F4 test methods, are illustrated in Fig. 5(b-d), in multi-modal spaces, the mSTOA achieves a comparative performance towards the promising regions, which is reflected in the convergence behavior. Furthermore, the mSTOA is working well over the hybrid functions, which is illustrated in Fig. 5(e-g) by F5, F6, together with F7. Furthermore, the composition functions symbolized by F8, F9, along with F10, as depicted in Fig. 5(h-j) illustrated that the suggested mSTOA realized comparative performance in solving problems of complex spaces in a very near away to that of a real world. Such quick convergence for the (near)-optimal solutions is observed and indicates that the proposed mSTOA is promising to solve optimization issues that demand fast computation, including online optimization issues. Although the mSTOA realizes the lowest average from the optimal so-far solutions with the stable convergence in the majority of the test functions; however the mSTOA has a limit on F2, F9, and late exploration on F10. These limits can be probable that the mutation CR and the population reduction step are not working in a proper way. This limit can be handled by a middle coordination step between the two parameters.

5.5. The qualitative metrics analysis

Although the result analyses shown previously emphasize that the proposed mSTOA is of high performance, more tests and analyses

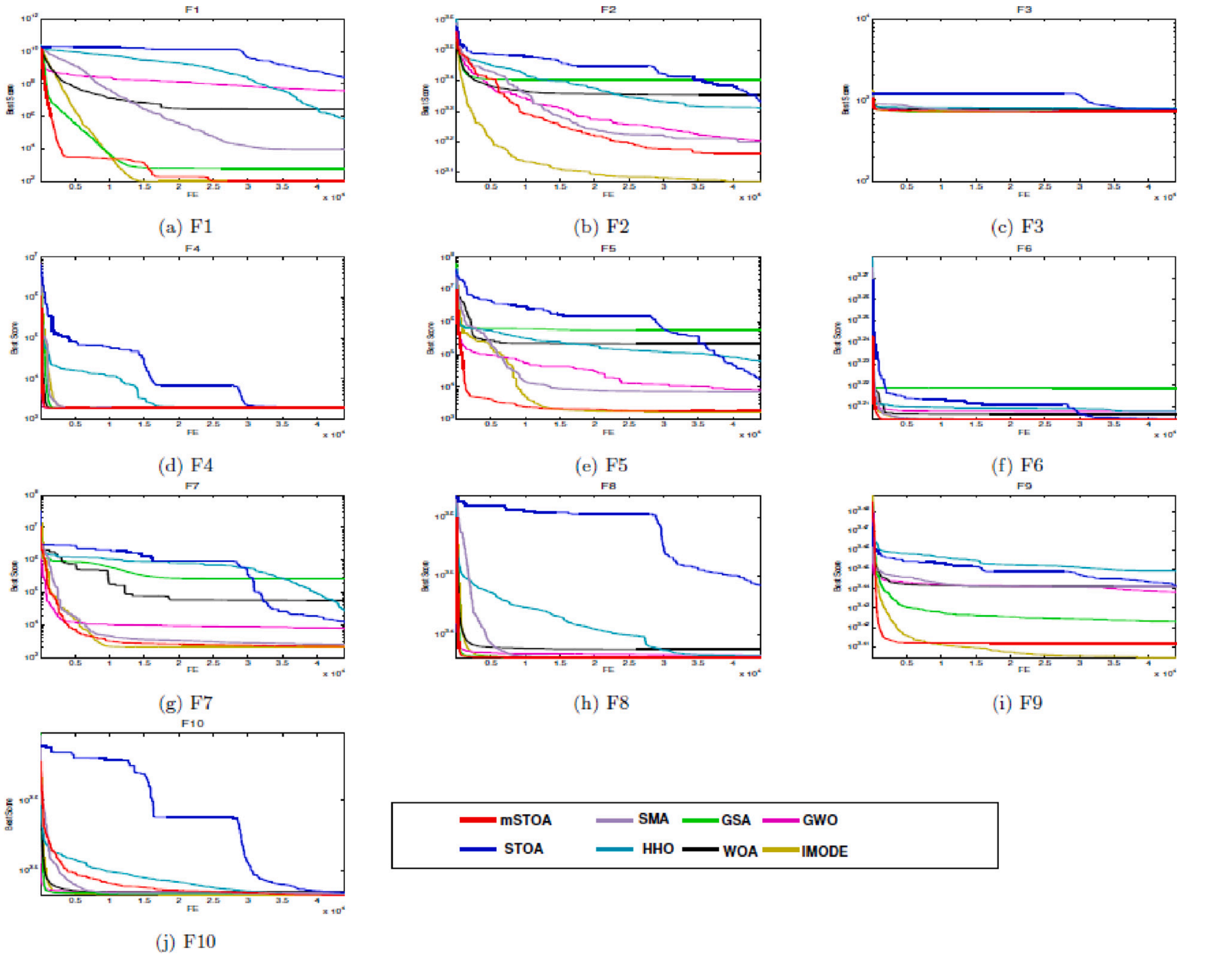


Fig. 5. The resulted convergence curves for the suggested mSTOA and the competing algorithms taken by the testing functions involved in CEC'2020, with $Dim = 10$.

can permit us to deduce robust conclusions concerning the algorithm performance in real problem-solving. As an example, observing the particles' behavior involving the search agents gives more insights into an optimization searching process and algorithm's convergence. Also, qualitative analysis for the proposed mSTOA is depicted in Fig. 6. In addition, the agent's behaviors are viewed in Fig. 6, which involve 3D views for (1) the functions, (2) search history, and (3) average fitness history, in addition to convergence curves.

The subsequent points are worthwhile in contrast to the previous qualitative analysis:

- *With respect to domain's topology — the functions in a 3D view :* For the first column in Fig. 6, the function is illustrated in 2-dimensional space. In this context, the functions have a specific topology. This gives insights into determining the type/shape of functions that makes the algorithm realize the best performance.
- *Respecting the search history:* For the second column in Fig. 6, search history is displayed for agents, starting from the first iteration till reaching the last one. Counter lines are used to depict the search domain, for which we can notice gradation from the shown blue-colored lines to the red-colored lines, which indicates an increase in the fitness value. The search history reflects that the proposed mSTOA, for some functions, can find the areas in which fitness values are the lowest.
- *With regard to average fitness history:* For the third column in Fig. 6, an average fitness history is presented, i.e., the averages for fitness value which is a function for the iteration number. In this regard, this average gives insights into the agents' general behavior and role in the improvement operation. It is noted that all history curves decrease, reflecting the population's improvement over each iteration. Such constant enhancement substantiates a behavior of collaborative searching and boosts the efficiency of the law of updating particles.

6. Experimental series 2: Application of mSTOA on feature selection (FS)

For assessing the availability of mSTOA in reality, in this work, we benchmark the mSTOA by executing the problem of FS. As a critical phase of the classification problem, FS represents a challenging operation of tackling the search space of high dimensions. The goal of FS represents the choice of the fewest representative feature sets from an original data set to realize the best accuracy of classification, which means that FS implements a dimensionality reduction procedure. Furthermore, the classification accuracy for the classifier in this processing is utilized to validate the efficiency of dimensionality reduction.

6.1. Description of the datasets

We have used some standard data set as shown in (see Table 5), that collected from Machine Learning Repository.[1]

[1] https://archive.ics.uci.edu/ml/index.php

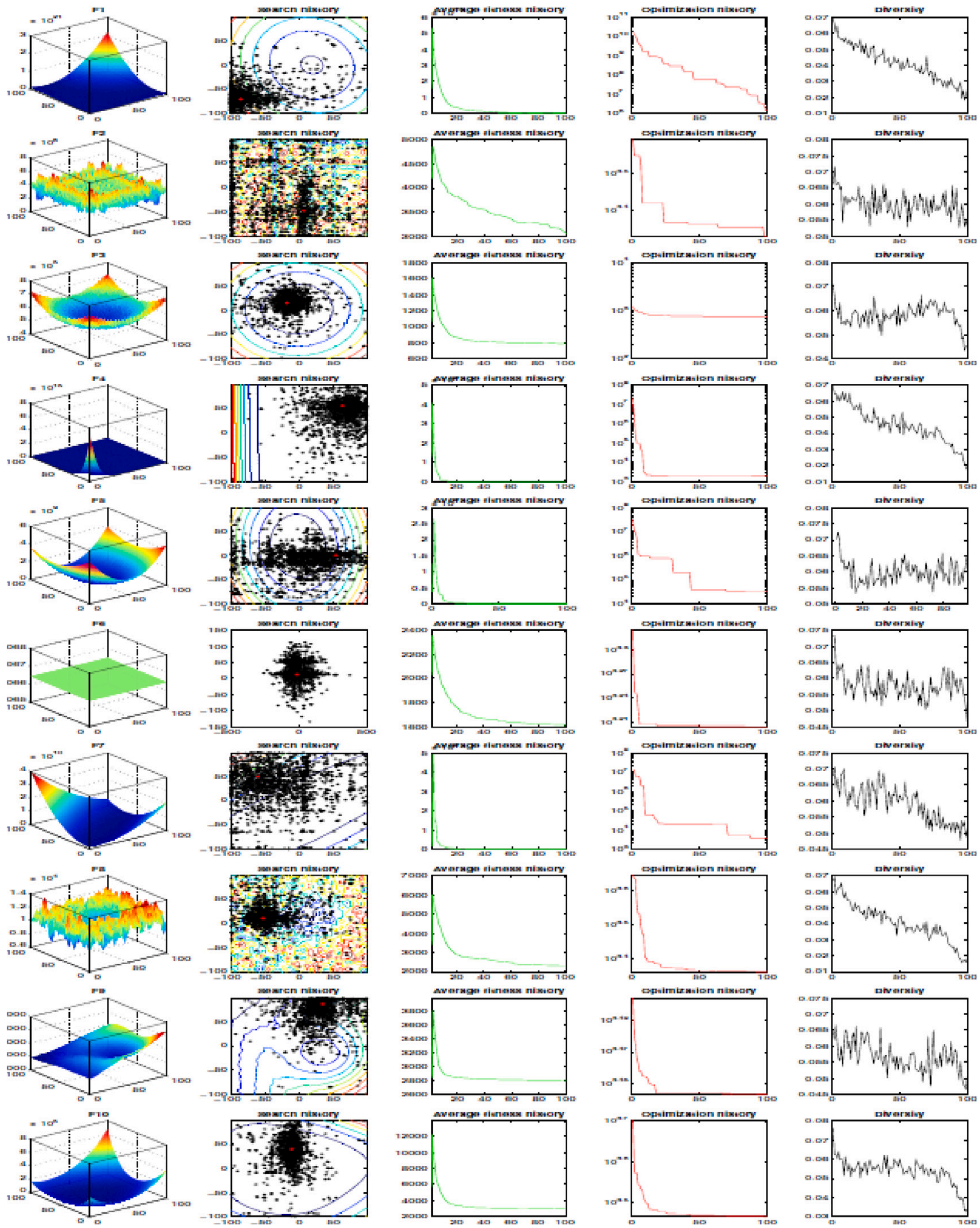


Fig. 6. The resulted qualitative metrics by the testing functions involved in the CEC'2020 benchmark: the 3D views for the functions, the search history, an average fitness history, along with an optimization history.

6.2. Architecture of FS based on mSTOA

The wrapper FS method is implemented to select the critical feature from several features and remove unnecessary ones. In this step, mSTOA-SVM produces N swarm agents over the first population. Every individual is considered a part of molecular features (descriptors) to be chosen for evaluation. That step is significant in both convergence and aptitude for the best solution. In this context, the population $Positions$ is produced randomly using:

$$Positions^i = lb^i + \lambda^i \times (ub^i - lb^i), \;\; i = 1, 2, \ldots, N \tag{25}$$

The lower (lb^i) and upper (ub^i) bounds for every candidate solution i lie in the $[0, 1]$ range. The λ^i represents a random number $\in [0, 1]$. For selecting a feature subset, the intermediate step named binary conversion is essential before the fitness evaluation process. Thus, each solution $Positions^i$ undergoes binary conversion ($Positions^i_{bin}$) by using:

$$Positions^i_{bin} = \begin{cases} 1 & \text{if } Positions^i > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

For clearing deeply the operation of conversion, a solution $Positions^i$ with ten features given as $Positions^i = [0.6, 0.2, 0.9, 0.33, 0.15, 0.8, 0.2, 0.75, 0.1, 0.9]$ is considered. Eq. (26) applies the operation of conversion to produce a binary vector $Positions^i_{bin} = [1, 0, 1, 0, 0, 1, 0, 1, 0, 1]$, where 1 implies that the features are picked out; otherwise, are not picked out. This indicates that $1st$, $3rd$, $6th$, $8th$, and the last features of original

Algorithm 2 Pseudo-code for mSTOA.

```
Initialization: Initialize the population of agents P_st^initial
Initialize the parameter S_A and C_B
Assess the fitness for every agent.
Assign P_bst to the best search agent.
while (FE < MAX_FEs) do
    for each i search agent do
        for (every solution portion (X_{i,j})) do
            if the rand ≤ CR or the j_rand = j then        ▷ Update strategy.
                Vec_j = moderate_distance_strategy(X_{i,j}) Eq. (12)
            else
                if t ≤ rand then                            ▷ Update strategy.
                    Vec_j = short_distance_strategy(X_{i,j}) Eq. (10)
                else
                    Vec_j = P_st(j) Eq. (14)
                end if
            end if
        end for
        if f(Vec) < f(X_i) then
            Update values of CR (Eq. (17)), F1 (Eq. (19)), F2 (Eq. (20))
        else
            Update values of CR (Eq. (18)), F1 (Eq. (19)), F2 (Eq. (20))
        end if
    end for
    Update parameter S_A and C_B
    Assess the fitness from each search agent.
    Update P_bst to the optimal search agent.
    Update t with Eq. (15)
    Call method of feature selection.
    Call SVMs classifier.
    FE = FE + 1
end while
```

datasets represent relevant ones that must be selected, whereas the others represent irrelevant features that must be ignored. After determining the selected features subset, the calculation of fitness function is done for each agent $Positions^{i}_{bin}$ to decide the quality of the features. Using the fitness function in the problem of feature selection is of two main goals: selecting a less number of feature items and realizing the smallest classification error or reaching highest classification accuracy. Herein, the optimal solution is indicated as the one with the most accurate classification together with selecting the smallest amount of features. The SVM classifier evaluates solutions. At each iteration, a smaller subset of features is selected by each solution, then the SVM classification algorithm is employed for training data samples by the selected feature subset and calculating the accuracy. So, the objective function presented in Eq. (27) is used to define the fitness for the ith solution as follows:

The solutions provided by mSTOA must be assessed through the iterative process to verify their performance. Herein, the fitness function, referred to as $fobj$, and employed by the mSTOA is formulated using:

$$fobj = \alpha + \beta \frac{|R|}{|C|} - G. \tag{27}$$

$$\beta = \alpha \tag{28}$$

$$fobj > T \tag{29}$$

The R denotes the classification error rate while the C refers to the number of feature elements within the dataset, α, and β represent two parameters referring to the significance of classification quality (computed by the classifiers), along with the length of the subset, respectively. The range [0, 1] defines the α. G denotes the classifier's group column (i.e., label column in the dataset), and T indicates the condition in which every algorithm is compared to the fitness function (i.e., the iteration numbers). Accordingly, the $fobj$ has to be larger than T to maximize the solution.

The mSTOA-SVM algorithm optimization process steps are formulated in Algorithm 2. Furthermore, steps of mSTOA processes with details are given in the flowchart of Fig. 7. The SVM is employed as a classifier in the phase of FS. Regarding the classification strategy, we employed the hold-out, which specializes 80% as a train set, whereas the remaining data is assigned as a test set. SVM (Algorithm 1).

Moreover, the FS method based on mSTOA combined with SVM steps are detailed in Fig. 7.

6.3. Performance measures

The subsequent metrics are employed to estimate the efficiency of mSTOA performance in each of the two problems: optimization and feature selection:

- Mean fitness value (Mean): This measure assesses the algorithms that identifies the relationship between the reduction of selection ratio and minimization of classification error ratio. The lower values refers to the better fitness value which is formulated as in the following:

$$Mean = \frac{1}{T} \sum_{i=1}^{T} fobj \tag{30}$$

- Best fitness (Best): The best fitness is calculated using the following equation:

$$Best = \max_{i=1} Q_i \tag{31}$$

- Worst fitness (Worst): The worst fitness is calculated using the following equation:

$$Best = \min_{i=1} Q_i \tag{32}$$

- Standard deviation (STD): the metric is employed to assessing the algorithm quality by analyzing the results computed across different runs as formulated in the next equation:

$$STD = \sqrt{\frac{1}{T-1} \sum_{i=1}^{T} (fobj - Mean)^2} \tag{33}$$

- Average Selection Size of features (ASS-AD): An average size for the features selected from the dataset to the overall features is calculated for T times by using the following equation:

$$ASS - AD = \frac{1}{T} \sum_{i=1}^{T} \frac{AVGsize^i}{L} \tag{34}$$

- Sensitivity: Is a performance metric to a binary classification. It defines the percentage of recognized positive instances, computed as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{35}$$

- Specificity: Is the proportion of detected negatives between the negative population in medical diagnosis, shown as follows:

$$Specificity = \frac{TN}{TN + FP} \tag{36}$$

where T is the whole number of iterations, and Q symbolizes the best score realized so far in each iteration. The $AVGsize^i$ is the number of feature elements selected from a dataset, and L denotes the original dataset's feature count. Moreover, the TP, TN, FN, and

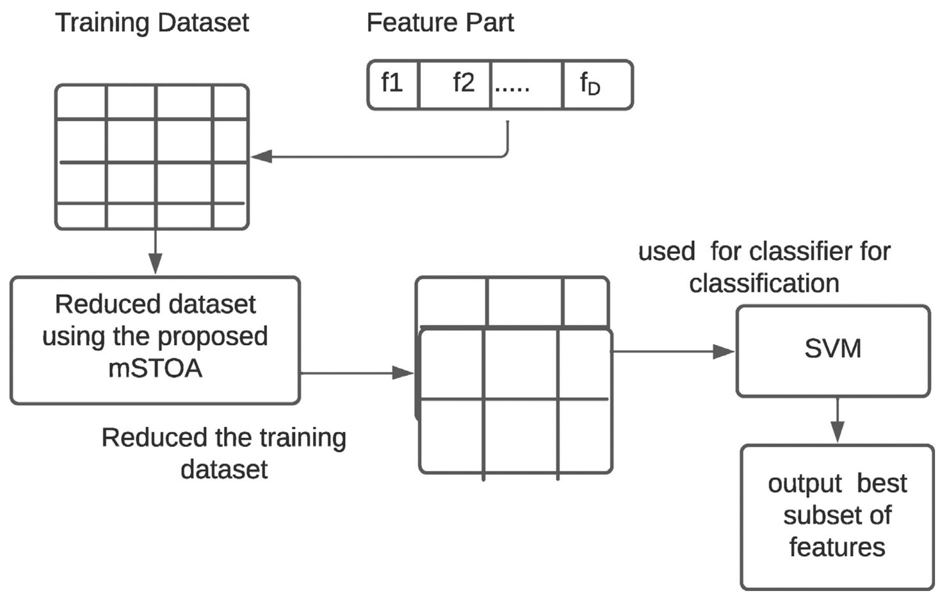


Fig. 7. The architecture of FS based on mSTOA.

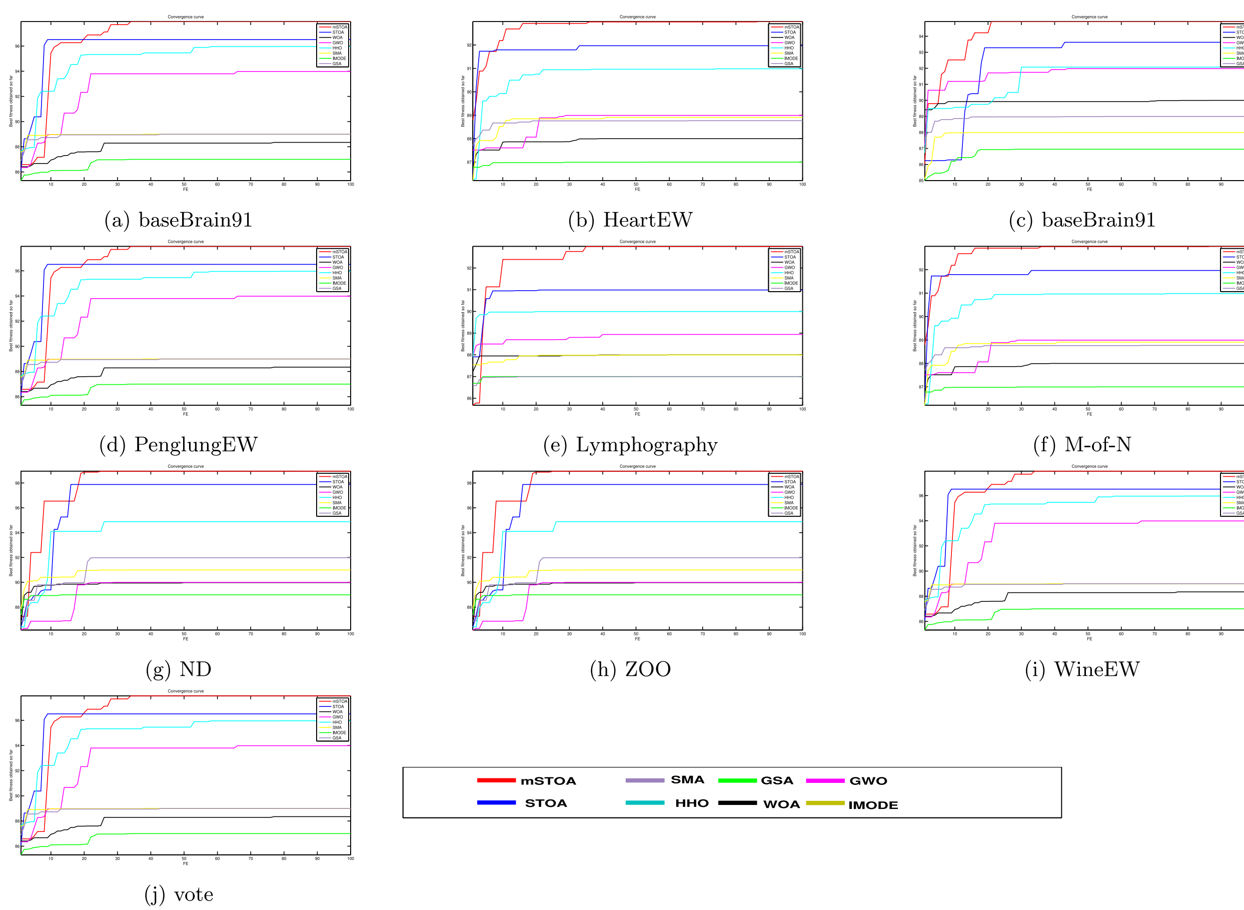


Fig. 8. The obtained convergence curves for the suggested mSTOA and the competitors taken on SVM.

FP symbolize the true positive, true negative, false negative, and false positive, respectively.

Besides the previously mentioned measures, Wilcoxon's rank-sum test (P-value) (Wilcoxon, 1992) is used which helps to determine whether the hypothesis is correct or not. If the p-value is found less than 0.05, the best values for (P-value) might be considered significant evidence versus the null hypothesis.

6.4. Statistical results analysis

Firstly, the proposed mSTOA-SVM and other compared algorithms are assessed on ten medical datasets. The proposed mSTOA has

Table 6
The values of Mean, STD, Best, Worst, sensitivity, specificity, ASS-AD, and *P*-value taken by the counterparts algorithms based on the SVM over the first five datasets.

Dataset	Algorithms	Mean	STD	Best	Worst	Attributes	Sensitivity	Specificity	ASS-AD	*P*-value
baseBrainT21	mSTOA	0.95	0.052705	98.230	97.998	2957	91.990	90.191	0.00028795	3.5039e−11
	STOA	0.94	0.002700	96.100	95.198	1775	89.100	88.100	0.00028780	2.5030e−10
	WOA	0.89	0.041000	91.030	89.090	372	86.121	85.090	0.0014495	2.2030e−10
	GWO	0.80	0.031623	88.011	85.011	59	84.110	80.010	0.011578	1.1130e−9
	HHO	0.91	0.041000	91.030	90.190	2957	86.121	85.090	0.00028795	3.5039e−10
	SMA	0.92	0.052000	94.130	93.990	202	88.100	90.191	0.00028795	2.5039e−9
	IMODE	0.79	0.021621	90.110	89.110	1073	82.010	81.111	0.00093185	1.0010e−8
	GSA	0.76	0.001621	89.110	88.110	1070	81.010	80.111	0.00093180	1.0020e−8
baseBrainT91	mSTOA	0.970	0.155280	97.131	96.190	587	89.291	88.889	0.0017036	0.03125
	STOA	0.96182	0.031623	95.101	94.290	310	88.110	87.910	0.0032218	3.4680e−08
	WOA	0.89571	0.082435	88.778	87.271	3121	82.134	81.190	0.0014495	2.1031e−10
	GWO	0.79	0.021600	86.110	84.001	759	82.100	81.100	0.001578	1.0130e−8
	HHO	0.92238	0.054973	90.120	89.090	2127	84.020	83.080	0.00027781	1.5038e−10
	SMA	0.901	0.022100	91.031	90.100	2836	66.667	77.778	0.00035261	1.4039e−6
	IMODE	0.78	0.020420	89.010	88.102	1179	80.010	79.011	0.00084185	1.0110e−7
	GSA	0.76	0.010421	88.010	86.102	1170	80.017	78.011	0.00074185	1.0010e−7
Lymphograph	mSTOA	0.94333	0.035312	94.117	93.120	10	80.010	97.180	0.0003071	0.02100
	STOA	0.91	0.016102	92.121	90.170	6	86.110	84.801	0.0032200	2.0181e−07
	WOA	0.94236	0.023095	86.160	84.660	14	79.120	78.011	0.061096	0.00019302
	GWO	0.92	0.017213	82.333	81.001	5	80.110	79.110	0.001048	1.1020e−8
	HHO	0.88	0.015713	87.110	85.160	8	80.111	79.460	0.125	0.00097656
	SMA	0.86897	0.025691	89.121	88.110	11	75.130	72.008	0.090909	0.015625
	IMODE	0.90667	1.1703e−16	84.667	83.167	7	80.100	79.101	0.090909	0.00019644
	GSA	0.91667	1.1503e−16	83.660	82.160	6	80.120	78.101	0.070909	0.00009644
HeartEW	mSTOA	0.92037	0.017568	96.030	95.020	9	88.190	87.870	0.14286	0.015625
	STOA	0.91081	0.020620	94.600	92.271	9	87.211	86.711	0.0022208	2.1680e−07
	WOA	0.89815	0.023505	88.593	86.185	7	80.130	79.090	0.12788	0.0078125
	GWO	0.87407	0.011501	83.010	82.100	7	81.120	80.110	0.11111	0.00048828
	HHO	0.87407	0.011712	89.021	88.191	7	80.010	82.081	0.00017780	1.2030e−9
	SMA	0.82407	0.033100	92.032	90.120	9	63.660	73.178	0.00032220	1.2019e−4
	IMODE	0.8937	0.017568	88.593	87.037	7	78.110	77.110	0.00024085	2.0120e−6
	GSA	0.91037	0.017560	87.593	86.034	6	77.110	76.110	0.00014085	2.0020e−5
BreastEW	mSTOA	0.98440	0.0026980	98.123	97.246	16	88.190	83.670	0.0625	2.5562e−06
	STOA	0.98509	0.0059206	97.123	96.368	7	85.110	83.211	0.0022208	0.03125
	WOA	0.87802	0.01502	88.390	87.180	6	80.031	77.080	0.11780	0.0068023
	GWO	0.8607	0.010500	85.000	84.004	6	80.100	79.100	0.11010	0.00047820
	HHO	0.87407	0.011712	91.021	90.191	7	80.010	82.081	0.00017780	1.2030e−9
	SMA	0.82407	0.033100	93.032	92.120	9	63.660	73.178	0.00032220	1.2019e−4
	IMODE	0.89030	0.017568	86.290	84.030	6	77.100	76.100	0.00023074	1.0120e−5
	GSA	0.90030	0.007568	85.190	83.030	5	76.120	75.100	0.00013074	1.0020e−6

achieved the highest value, but the GSA algorithm achieved the lowest results for mean and STD statistical results, as shown in Tables 6, and 7. The proposed mSTOA-SVM minimizes accuracy and decreases the number of features. Also, the mSTOA-SVM achieved the best value for mean, STD, worst, best, and CPU time across all datasets. Based on the previously presented analysis, the proposed mSTOA-SVM approach has realized better results than its other counterparts. The STOA represents the second-ranked algorithm, whereas the GSA comes in the last rank. With the aim of fair comparison that employs the same parameter settings, the search agent's number was assigned as 30 in all experiments with a different number of dimensions, according to the different dimensions of datasets.

6.5. Convergence curves analysis

The convergence curves for the suggested mSTOA compared with other algorithms are presented in Fig. 8 over ten datasets. The proposed mSTOA reaches a stable point in all datasets, suggesting the ability of mSTOA to converge well. Furthermore, mSTOA has achieved the largest average among the best solutions and is the quickest algorithm over most datasets. This quick convergence for the (near)-optimal solutions is observed, which makes the suggested mSTOA a promising optimizer in solving FS problems, and realizes high accuracy compared to other algorithms, as declared in Fig. 8.

6.6. Boxplots curves analysis

The boxplot can be used to evaluate the performance of several datasets as a non-parametric. However, in descriptive statistics, it is known that a boxplot is a method to graphically depict groups of numerical data according to their quartets. Boxplots can have lines extended in a vertical way from the boxes, denoting variability outside the lower and upper quartets; thus, the terms "box-and-whisker diagram" and "box-and-whisker plot". The maximum or minimum are the largest or the lowest data points realized by the algorithm. The plot of outlines can be shown as individual points. They view variation in the statistical population's samples without making assumptions about the statistical distribution. It is also obvious that space between the various parts in the box refers to the data's spread (degree of dispersion) and skewness and displays outlines. In the experiments, the boxplots for mSTOA -SVM over the ten datasets are depicted in Fig. 9. The boxplots of suggested mSTOA are very narrow compared to other distributions for the competitors in most datasets; thus, it has the most significant values. Actually, the proposed mSTOA performs well and outperforms the other algorithms in most of the datasets.

6.7. Discussion

Firstly, the proposed mSTOA and other compared algorithms are assessed on the CEC'20 benchmark. After that, ten medical datasets are used to evaluate the proposed mSTOA-SVM performance. For the

Table 7
The values of Mean, STD, Best, Worst, sensitivity, specificity, ASS-AD, and *P*-value taken by the counterparts algorithms based on the SVM over the second five datasets.

Dataset	Algorithm	Mean	STD	Best	Worst	Attributes	sensitivity	specificity	ASS-AD	*P*-value
ZOO	mSTOA	0.99330	0.021082	96.020	94.021	9	86.090	85.761	0.1	0.0039063
	STOA	0.98095	0.02459	95.010	94.101	10	85.090	84.110	0.0003108	1.3081e−08
	WOA	0.86511	0.012035	85.070	83.001	14	79.021	78.010	0.0015090	1.0020e−8
	GWO	0.78	0.001701	84.021	83.120	12	81.120	80.120	0.000218	1.0120e−7
	HHO	0.931	0.011080	89.100	89.130	13	83.010	80.120	0.00017780	1.5008e−9
	SMA	0.90167	0.022100	92.031	90.100	12	66.667	77.778	0.00035261	1.4039e−6
	IMODE	0.76	0.021221	85.120	84.102	12	78.120	77.100	0.00172080	1.0020e−5
	GSA	0.75	0.011221	84.120	82.102	10	76.120	75.100	0.00072080	1.0020e−6
vote	mSTOA	0.99330	0.021082	99.020	98.021	9	86.090	85.761	0.1	0.0039063
	STOA	0.98095	0.02459	97.010	96.101	10	85.090	84.110	0.0003108	1.3081e−08
	WOA	0.86511	0.012035	87.070	86.001	10	79.021	78.010	0.0014091	1.0120e−7
	GWO	0.77	0.000701	83.001	82.100	10	81.120	80.120	0.000218	1.0100e−6
	HHO	0.930	0.021080	89.520	88.120	15	80.112	79.100	0.0000708	1.5109e−7
	SMA	0.95160	0.062100	93.031	91.100	12	66.667	77.778	0.00035261	1.4039e−6
	IMODE	0.76	0.009220	87.100	86.110	10	78.120	77.100	0.00172080	1.0020e−5
	GSA	0.74	0.001220	86.100	84.110	11	77.100	75.100	0.00072080	1.0010e−6
ND	mSTOA	0.99420	0.010081	98.825	97.020	12	85.092	84.460	0.27105	0.0048063
	STOA	0.98195	0.03551	96.619	95.951	11	87.199	86.190	0.0004188	1.3090e−07
	WOA	0.84501	0.001030	87.170	86.041	11	77.120	76.917	0.0094090	1.0120e−7
	GWO	0.77	0.002800	84.200	83.102	11	82.126	81.325	0.000318	1.0120e−7
	HHO	0.920	0.091780	88.621	89.621	12	79.910	78.122	0.0008708	1.6129e−7
	SMA	0.94169	0.099100	93.730	92.190	12	78.660	77.778	0.00175261	1.4259e−6
	IMODE	0.77	0.018220	86.120	85.196	10	79.127	78.120	0.00172080	1.0120e−7
	GSA	0.76	0.008221	85.100	84.190	11	77.120	76.120	0.00072080	1.0020e−6
M-of-n	mSTOA	0.994	0.013499	97.920	95.125	10	84.990	83.669	0.27105	0.0048063
	STOA	0.98195	0.03551	96.619	95.951	11	87.199	86.190	0.0004188	1.3090e−07
	WOA	0.84501	0.001030	89.170	87.041	11	77.120	76.917	0.0094090	1.0120e−7
	GWO	0.77	0.002800	84.200	83.102	11	82.126	81.325	0.000318	1.0120e−7
	HHO	0.920	0.091780	92.621	91.621	12	79.910	78.122	0.0008708	1.6129e−7
	SMA	0.94169	0.099100	93.730	92.190	12	78.660	77.778	0.00175261	1.4259e−6
	IMODE	0.77	0.018220	87.120	88.196	10	79.127	78.120	0.00172080	1.0120e−7
	GSA	0.74	0.008200	84.120	83.120	12	75.100	75.100	0.00052180	1.0011e−4
WineEW	mSTOA	0.99722	0.0087841	96.920	92.025	5	82.191	80.061	0.10101	0.0028060
	STOA	0.97222	1.1703e−16	95.110	94.129	6	84.091	82.021	0.0012088	1.2010e−06
	WOA	0.85501	0.00130	89.011	88.140	9	76.021	75.920	0.0074091	1.0021e−6
	GWO	0.76	0.001801	84.101	83.012	10	81.122	80.120	0.001313	1.0110e−6
	HHO	0.900	0.041731	93.420	92.120	10	77.112	76.021	0.0013718	1.3120e−6
	SMA	0.94444	0.011712	94.121	91.031	10	79.121	76.172	0.00015260	1.3209e−2
	IMODE	0.75	0.017201	86.011	87.092	9	77.021	75.141	0.00052081	1.0020e−6
	GSA	0.74	0.017200	84.011	85.091	8	76.020	74.140	0.00052000	1.0011e−5

CEC'20 benchmark, quantitative and qualitative metrics are used to assess mSTOA performance. The proposed mSTOA has achieved the highest value. However, the GSA algorithm achieved the lowest results for mean and STD statistical results, as illustrated in Table 4 and the best figure is the minimum convergence curve and boxplot as drawn in Fig. 8, and Fig. 9. Fig. 3 shows the parameter space used for the 3D visualization of the CEC'20 functions to understand each problem's nature and differences. The qualitative metrics are used to draw robust conclusions concerning the algorithm's performance in a real problem to confirm the high performance for the proposed mSTOA, as shown in Fig. 6.

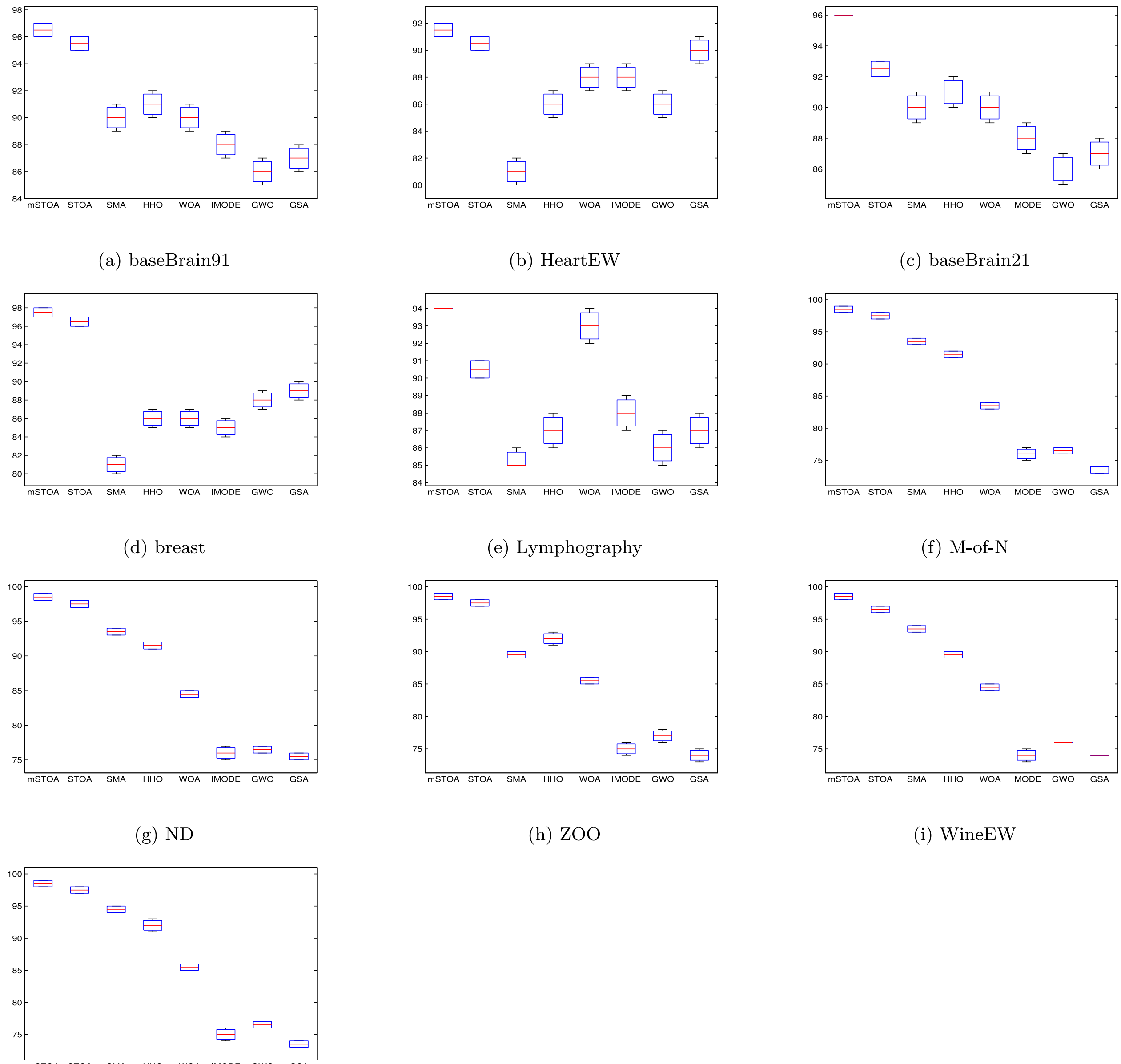
For FS, the proposed mSTOA-SVM maximizes accuracy and decreases the feature number. The mSTOA -SVM achieved the best mean, STD, the best value, and worst value, along with CPU time as shown in Tables 6, and 7, over all datasets. The convergence curves is utilized as evidence to support this fact, which depicts how the mSTOA -SVM reaches a stable point over ten medical datasets, as illustrated in Fig. 8. The convergence curves were chosen since they represent the relationship between the fitness functions and number of FE. They indicate the best-performing algorithm depending on the comparison among different approaches. Boxplot analysis indicates that mSTOA -SVM achieved higher performance than other algorithms, as shown in Fig. 9.

Based on the previous analysis, the suggested mSTOA -SVM approach has achieved better results than the other counterparts. The STOA represents the second-ranked algorithm, whereas the GSA ranks last. For making a fair comparison using the same parameter settings, the search agent's number was assigned as 30 in all experiments with a different number of dimensions.

7. Conclusion and future directions

This paper suggested an alternative method called mSTOA, a modified version of the original Sooty Tern Optimization Algorithm (STOA). The mSTOA employs three strategies: the adopted exploration/exploitation balance strategy, novel self-adaptive for control parameters, and an inherited population reduction strategy to improve the original STOA, so that it avoids local optimal and achieves a soft transition/balance between both exploration and exploitation stages. The mSTOA performance is validated on the CEC2020 benchmark test suite, and a set of graphical representation methods are provided, i.e., boxplots, convergence curves, and qualitative metrics. The results demonstrate the superior performance of the mSTOA; further, the introduced mSTOA is applied to optimize the feature selection problem using a set of well-known datasets and classification approaches. The SVM approach was utilized for data classification and realized a promising average accuracy rate of 98.718% and 85.718%. Furthermore, the utilization of mSTOA as a feature extractor significantly raised the SVM classification performance. Eventually, the experimental results reported high classification results for the proposed algorithm in contrast to other methods.

In future directions, the proposed mSTOA can be utilized in the following future perspectives, such as; (1) solving other large-scale



(a) baseBrain91 (b) HeartEW (c) baseBrain21

(d) breast (e) Lymphography (f) M-of-N

(g) ND (h) ZOO (i) WineEW

(j) vote

Fig. 9. The boxplot curves for the proposed mSTOA and the competing algorithms taken over SVM.

and real-world optimization issues. (2) Solving different real-world and engineering issues that have unknown search domains. (3) tackle different problems involving feature selection, parameter identification, and task scheduling. (4) solving multi-objective issues can be considered in future studies.

Compliance with ethical standards

This article does not contain any studies with human participants or animals performed by any of the authors.

Credit author statement

Essam H. Houssein: Supervision, Software, Methodology, Conceptualization, Formal analysis, Investigation, Visualization, Writing - review & editing. Diego Oliva: Supervision, Methodology, Conceptualization, Formal analysis, Resources, Investigation, Validation, Writing - review & editing. Emre Çelik: Methodology, Conceptualization, Investigation, Visualization, Formal analysis, Writing - review & editing. Marwa M. Emam: Conceptualization, Formal analysis, Resources, Data curation, Writing - original draft, Writing - review & editing. Rania M. Ghoniem: Methodology, Conceptualization, Formal analysis, Resources, Data curation, Validation, Writing - original draft, Writing - review & editing. All authors read and approved the final paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Abd Elaziz, M., Moemen, Y. S., Hassanien, A. E., & Xiong, S. (2020). Toxicity risks evaluation of unknown fda biotransformed drugs based on a multi-objective feature selection approach. *Applied Soft Computing*, *97*, Article 105509.

Abd Elaziz, M., & Yousri, D. (2021). Automatic selection of heavy-tailed distributions-based synergy henry gas solubility and harris hawk optimizer for feature selection: case study drug design and discovery. *Artificial Intelligence Review*, *54*, 4685–4730.

Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, *191*, Article 116158.

Ali, H. H., Fathy, A., & Kassem, A. M. (2020). Optimal model predictive control for lfc of multi-interconnected plants comprising renewable energy sources based on recent sooty terns approach. *Sustainable Energy Technologies and Assessments*, *42*, Article 100844.

Atwood, J., & Towsley, D. (2016). Diffusion-convolutional neural networks. In *Advances in neural information processing systems* (pp. 1993–2001).

Braik, M. S. (2021). Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems. *Expert Systems with Applications*, *174*, Article 114685.

Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, *243*, Article 108457.

Byrne, D. M. (1986). The taguchi approach to parameter design. *ASQ's Annual Quality Congress Proceedings*, *40*(168), 1986.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79.

Chopra, N., & Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications*, *198*, Article 116924.

Črepinšek, S., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms. *ACM Computer Surveys*, *45*, 1–33.

Das, S., Abraham, A., Chakraborty, U. K., & Konar, A. (2009). Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation*, *13*, 526–553.

Deng, X., Li, Y., Weng, J., & Zhang, J. (2019). Feature selection for text classification: A review. *Multimedia Tools & Applications*, *78*.

Dhiman, G., Garg, M., Nagar, A., Kumar, V., & Dehghani, M. (2021). A novel algorithm for global optimization: rat swarm optimizer. *Journal of Ambient Intelligence and Humanized Computing*, *12*, 8457–8482.

Dhiman, G., & Kaur, A. (2019). Stoa: a bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of Artificial Intelligence*, *82*, 148–174.

Dutta, A., Riba, P., Lladós, J., & Fornés, A. (2019). Hierarchical stochastic graphlet embedding for graph-based pattern recognition. *Neural Computing and Applications*, 1–18.

Eiben, A. E., & Schoenauer, M. (2002). Evolutionary computing. *Information Processing Letters*, *82*, 1–6.

Goh, G. B., Sakloth, K., Siegel, C., Vishnu, A., & Pfaendtner, J. (2018a). Multimodal deep neural networks using both engineered and learned representations for biodegradability prediction. arXiv preprint arXiv:1808.04456.

Goh, G. B., Siegel, C., Vishnu, A., & Hodas, N. (2018b). Using rule-based labels for weak supervised learning: a chemnet for transferable chemical property prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 302–310).

Gupta, Y., & Saini, A. (2017). A novel fuzzy-pso term weighting automatic query expansion approach using combined semantic filtering. *Knowledge-Based Systems*, *136*, 97–120.

Hancer, E., Xue, B., Zhang, M., Karaboga, D., & Akay, B. (2018). Pareto front feature selection based on artificial bee colony optimization. *Information Sciences*, *422*, 462–479.

Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. (2020). A modified henry gas solubility optimization for solving motif discovery problem. *Neural Computing and Applications*, *32*, 10759–10771.

Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. (2022). Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, *192*, 84–110.

Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems*, *101*, 646–667.

Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S., & Al-Atabany, W. (2021). Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *51*, 1531–1551.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*, 83–85.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872.

Houssein, E. H., Abohashima, Z., Elhoseny, M., & Mohamed, W. M. (2022). Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision. *Expert Systems with Applications*, Article 116512.

Houssein, E. H., Çelik, E., Mahdy, M. A., & Ghoniem, R. M. (2022). Self-adaptive equilibrium optimizer for solving global, combinatorial, engineering, and multi-objective problems. *Expert Systems with Applications*, *195*, Article 116552.

Houssein, E. H., Emam, M. M., & Ali, A. A. (2021). Improved manta ray foraging optimization for multi-level thresholding using covid-19 ct images. *Neural Computing and Applications*, *33*, 16899–16919.

Houssein, E. H., Emam, M. M., & Ali, A. A. (2022). An optimized deep learning architecture for breast cancer diagnosis based on improved marine predators algorithm. *Neural Computing and Applications*, 1–19.

Houssein, E. H., Hassaballah, M., Ibrahim, I. E., AbdElminaam, D. S., & Wazery, Y. M. (2022). An automatic arrhythmia classification model based on improved marine predators algorithm and convolutions neural networks. *Expert Systems with Applications*, *187*, Article 115936.

Houssein, E. H., Hassan, H. N., Al-Sayed, M. M., & Nabil, E. (2022). Gene selection for microarray cancer classification based on manta rays foraging optimization and support vector machines. *Arabian Journal for Science and Engineering*, *47*, 2555–2572.

Houssein, E. H., Helmy, B. E.-d., Oliva, D., Elngar, A. A., & Shaban, H. (2021). A novel black widow optimization algorithm for multilevel thresholding image segmentation. *Expert Systems with Applications*, *167*, Article 114159.

Houssein, E. H., Helmy, B. E.-d., Oliva, D., Jangir, P., Premkumar, M., Elngar, A. A., & Shaban, H. (2022). An efficient multi-thresholding based covid-19 ct images segmentation approach using an improved equilibrium optimizer. *Biomedical Signal Processing and Control*, *73*, Article 103401.

Houssein, E. H., Hosney, M. E., Oliva, D., Mohamed, W. M., & Hassaballah, M. (2020). A novel hybrid harris hawks optimization and support vector machines for drug design and discovery. *Computers & Chemical Engineering*, *133*, Article 106656.

Houssein, E. H., Neggaz, N., Hosney, M. E., Mohamed, W. M., & Hassaballah, M. (2021). Enhanced harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities. *Neural Computing and Applications*, *33*, 13601–13618.

Houssein, E. H., Rezk, H., Fathy, A., Mahdy, M. A., & Nassef, A. M. (2022). A modified adaptive guided differential evolution algorithm applied to engineering applications. *Engineering Applications of Artificial Intelligence*, *113*, Article 104920.

Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. (2020). Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *94*, Article 103731.

Houssein, E. H., Saber, E., Ali, A. A., & Wazery, Y. M. (2022). Centroid mutation-based search and rescue optimization algorithm for feature selection and classification. *Expert Systems with Applications*, *191*, Article 116235.

Huang, C.-L., & Dun, J.-F. (2008). A distributed pso–svm hybrid system with feature selection and parameter optimization. *Applied Soft Computing*, *8*, 1381–1391.

Hussien, A. G., Hassanien, A. E., & Houssein, E. H. (2017). Swarming behaviour of salps algorithm for predicting chemical compound activities. In *In 2017 eighth international conference on intelligent computing and information systems, Vol. 31* (pp. 5–320). IEEE.

Jia, H., Li, Y., Sun, K., Cao, N., & Zhou, H. M. (2021). Hybrid sooty tern optimization and differential evolution for feature selection. *Computer System Science Engineering*, *39*, 321–335.

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Computational cybernetics and simulation*, *1997 IEEE international conference on systems, man, and cybernetics, Vol. 5* (pp. 4104–4108). IEEE.

Kılıç, F., Kaya, Y., & Yildirim, S. (2021). A novel multi population based particle swarm optimization for feature selection. *Knowledge-Based Systems*, *219*, Article 106894.

Li, S., Chen, H., Wang, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*.

Liu, W., Yu, X., Raj, B., Yi, L., Zou, X., & Li, M. (2015). Efficient autism spectrum disorder prediction with eye movement: A machine learning framework. In *2015 international conference on affective computing and intelligent interaction* (pp. 649–655). IEEE.

Martínez, M. J., Dussaut, J. S., & Ponzoni, I. (2018). Biclustering as strategy for improving feature selection in consensus qsar modeling. *Electronic Notes in Discrete Mathematics*, *69*, 117–124.

Martínez, M. J., Razuc, M., & Ponzoni, I. (2019). Modesus: A machine learning tool for selection of molecular descriptors in qsar studies applied to molecular informatics. *BioMed Research International*, (2019).

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, *89*, 228–249.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Mohamed, A. W., Hadi, A. A., Mohamed, A. K., & Awad, N. H. (2020). Evaluating the performance of adaptive gainingsharing knowledge based algorithm on cec 2020 benchmark problems. In *2020 IEEE congress on evolutionary computation* (pp. 1–8). IEEE.

Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F., & Rodríguez, A. (2020). A better balance in metaheuristic algorithms: Does it exist? *Swarm and Evolutionary Computation*, *54*, Article 100671.

Morales-Castañeda, B., Zaldívar, D., Cuevas, E., Rodríguez, A., & Navarro, M. A. (2021). Population management in metaheuristic algorithms: Could less be more? *Applied Soft Computing*, *107*, Article 107389.

Nadimi-Shahraki, M. H., & Zamani, H. (2022). Dmde: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization. *Expert Systems with Applications*, *198*, Article 116895.

Neggaz, N., Houssein, E. H., & Hussain, K. (2020). An efficient henry gas solubility optimization for feature selection. *Expert Systems with Applications*, *152*, Article 113364.

Oliva, D., Rodriguez-Esparza, E., Martins, M. S., Elaziz, M. Abd., Hinojosa, S., Ewees, A. A., & Lu, S. (2020). Balancing the influence of evolutionary operators for global optimization. In *2020 IEEE congress on evolutionary computation* (pp. 1–8). IEEE.

Pinheiro, R. H., Cavalcanti, G. D., Correa, R. F., & Ren, T. I. (2012). A global-ranking local feature selection method for text categorization. *Expert Systems with Applications*, *39*, 12851–12857.

Putra, R. I. D., Maulana, A. L., & Saputro, A. G. (2019). Study on building machine learning model to predict biodegradable-ready materials. In *AIP conference proceedings, Vol. 2088*. AIP Publishing LLC, Article 060003.

Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*, 303–315.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). Gsa: a gravitational search algorithm. *Information Sciences*, *179*, 2232–2248.

Rodrigues, D., Pereira, L. A., Almeida, T., Papa, J. P., Souza, A., Ramos, C. C., & Yang, X.-S. (2013). Bcs: A binary cuckoo search algorithm for feature selection. In *2013 IEEE international symposium on circuits and systems* (pp. 465–468). IEEE.

Rodriguez-Perez, R., Vogt, M., & Bajorath, J. (2017). Support vector machine classification and regression prioritize different structural features for binary compound activity and potency value prediction. *ACS Omega*, *2*, 6371–6379.

Sallam, K. M., Elsayed, S. M., Chakrabortty, R. K., & Ryan, M. J. (2020). Improved multi-operator differential evolution algorithm for solving unconstrained problems. In *2020 IEEE congress on evolutionary computation* (pp. 1–8). IEEE.

Singh, A., Sharma, A., Rajput, S., Mondal, A. K., Bose, A., & Ram, M. (2022). Parameter extraction of solar module using the sooty tern optimization algorithm. *Electronics*, *11*(564).

Suyanto, S., Ariyanto, A. A., & Ariyanto, A. F. (2022). Komodo mlipir algorithm. *Applied Soft Computing*, *114*, Article 108043.

Thaher, T., Chantar, H., Too, J., Mafarja, M., Turabieh, H., & Houssein, E. H. (2022). Boolean particle swarm optimization with various evolutionary population dynamics approaches for feature selection problems. *Expert Systems with Applications*, *195*, Article 116550.

Tharwat, A., Moemen, Y. S., & Hassanien, A. E. (2016). A predictive model for toxicity effects assessment of biotransformed hepatic drugs using iterative sampling method. *Scientific Reports*, *6*, 1–13.

Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S., & Fong, S. (2018). Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science & Technology*, *26*.

Wan, Y., Wang, M., Ye, Z., & Lai, X. (2016). A feature selection method based on modified binary coded ant colony optimization algorithm. *Applied Soft Computing*, *49*, 248–258.

Wang, Y., Feng, L., & Zhu, J. (2018). Novel artificial bee colony based feature selection method for filtering redundant information. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *48*, 868–885.

Wang, X., Yang, J., Teng, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, *28*, 459–471.

Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics, Vol. 19* (pp. 6–202). Springer.

Yadav, D., Sharma, A., Singh, M., & Goyal, A. (2019). Feature extraction based machine learning for human burn diagnosis from burn images. *IEEE Journal of Translational Engineering in Health and Medicine*, *7*, 1–7.

Zainudin, M., Sulaiman, M., Mustapha, N., Perumal, T., Nazri, A., Mohamed, R., & Manaf, S. (2017). Feature selection optimization using hybrid relief-f with self-adaptive differential evolution. *International Journal Intelligent Engineering System*, *10*, 21–29.

Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2022). Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Computer Methods in Applied Mechanics and Engineering*, *392*, Article 114616.

Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, *388*, Article 114194.