IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 3, 2021, accepted August 28, 2021, date of publication September 7, 2021, date of current version September 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111121

Fractional Calculus-Based Slime Mould Algorithm for Feature Selection Using Rough Set

REHAB ALI IBRAHIM[1], DALIA YOUSRI[2], MOHAMED ABD ELAZIZ[1,4,5], SAMAH ALSHATHRI[3], AND IBRAHIM ATTIYA[1]

[1]Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt
[2]Department of Electrical Engineering, Faculty of Engineering, Fayoum University, Faiyum 63514, Egypt
[3]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 84428, Saudi Arabia
[4]Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, United Arab Emirates
[5]School of Computer Science and Robotics, Tomsk Polytechnic University, 634050 Tomsk, Russia

Corresponding author: Samah Alshathri (sealshathry@pnu.edu.sa)

This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Program.

ABSTRACT Features Selection (FS) techniques have been applied to several real-world applications which contain high dimension data. These FS techniques have main objectives that aim to achieve them, such as removing irrelevant features and increasing classification accuracy. This is considered a bi-objectives optimization problem that requires a suitable technique that can balance between the objectives. So, different sets of FS techniques have been developed, and those techniques that depend on meta-heuristic (MH) established their performance overall traditional FS techniques. However, these MH approaches still require more enhancement to neutralize their exploration and exploitation abilities during the searching process. Enhancing the meta-heuristic optimization algorithm using the perspective of fractional calculus (FC) is an attractive and novel approach. In this paper, the slime mould algorithm (SMA) is modified using the FC for handling the optimizer drawback of the inefficient diversification phase. As a result, a fractional-order SMA is proposed to avoid the local solutions and discover the search landscape efficiently via considering a historic memorize of agents' positions. The proposed FOSMA is applied to extract features from a set of real-world data and increase classification accuracy. For boosting the optimizer performance while processing with these datasets, the rough set (RS) is used as the fitness function to handle the uncertainty inside the real-world data. Finally, the proposed FOSMA's results are compared with a set of well-known FS techniques to investigate its performance. The comparison illustrates the superiority of FOSMA in providing high accuracy.

INDEX TERMS Feature selection (FS), fractional calculus, slime mould algorithm (SMA), meta-heuristic (MH).

I. INTRODUCTION

The high advancement achieved in information and communication devices and technologies had led to an increase in the dimension of the collection and the stored data at an exponential rate. This has been performed in different real-world fields including medical [1], engineering [2], [3], industrial [4], and agriculture [5]. However, most of the features of this collected data can be considered as irrelevant and redundant, which leads to degradation of the performance of data analysis techniques [6]. So, the dimension reduction (DR) methods have been more attentive, and they have been applied to several applications [7]–[10].

In that circumstance, there are several DR methods have been proposed, such as transformation and selection approaches [11]–[14]. In the transformation methods, the data is mapped into another domain, and these methods including principal component analysis, independent component analysis, and factor analysis. However, the main limitation of this type of DR is the changing of the originality of the collected information. The second type of DR is the feature selection (FS) approach which aims to find the relevant features without changing the originality of the data [15], [16]. These methods including several methods that can be

The associate editor coordinating the review of this manuscript and approving it for publication was M. Shamim Kaiser.

categorized into the wrapper and filter-based methods. The wrapper approaches depend on the learning approach to find the relevant subset of features, while the filter method doesn't rely on the learning approach.

Recently, several FS techniques based on meta-heuristic (MH) algorithms have been manifested and utilized for various optimization problems. We have some examples for these MHs such as Moth-Flame Optimization (MFO) [17], Salp Swarm Algorithm [18], Runner-Root Algorithm (RRA) [19], arithmetic optimization algorithm (AOA) [20], Aquila Optimizer (AO) [21], Differential Evolution (DE) [22], genetic algorithm (GA) [23], and Grey-wolf optimization Algorithm [24]. Moreover, Shukla *et al.* [25] proposed a hybrid metaheuristics approach for cancer type classification, and diao *et al.* [26] used modified PSO for identifying or detecting and expectation of the existence and usefulness of pipelines leakage. This technique depended on sufficient processing of signals that has acoustic emission and enhanced the variational mode decomposition (VMD) for signal denoising using the particle swarm optimization (PSO) algorithm. Ibrahim *et al.* [27] improved the Grey Wolf Optimizer (GWO) for developing the potency of the exploitation and the exploration of GWO utilizing the approach of chaotic with the map of kind logistic, the popular approach of differential evolution, the Opposition-Based Learning, and a physical operator named the disruption operator. Jingwei *et al.* [28] used the binary form of the Harris hawk optimization (HHO) algorithm for solving some classification missions like feature selection and compared the results of HHO technique with other feature selection methods based on metaheuristics, genetic algorithm (GA), binary multi-verse optimizer (BMVO), as well as binary salp swarm algorithm (BSSA), in addition to the binary differential evolution (BDE).

In that context more FS works depended on combinations of such MH techniques, for example Wenlong *et al.* [29] enhanced a proper active mechanism for forecasting hybridizing decomposition of two layers and enhanced hybrid differential evolution with Harris hawks optimization (IHDEHHO), and kernel extreme learning machine (KELM), in addition to phase space reconstruction (PSR) for wind speed forecasting of kind multi-step based on a decomposition of two layers. Ranyaet al. [30] proposed binary GWO with HHO for obtaining a memetic approach noted HBGWOHHO. The sigmoid is one of the transfer functions that are utilized for transferring the continuous search space into a binary one for satisfying the feature selection nature requirement. A wrapper-based k-Nearest neighbor is utilized for evaluating the goodness of the selected features. Thaher et a. [31] proposed a model for the task of classification that is represented in expectation of the news that is fake through Arabic tweets that use Natural Language Processing (NLP), Machine Learning, and HHO for selecting features task. Where several popular learning algorithms are examined with hybrids of features, obtaining content-based, and word features, in addition to a user profile. The obtained results proved that the Logistic Regression with an approach known as Term Frequency-Inverse Document Frequency (TF-IDF) model gets the rank which is the finest one. Furthermore, feature selection by utilizing the binary version of the HHO algorithm plays is useful in reducing dimensionality, thereby developing the performance of the model for learning for the detection of not true news.

By inspecting the literature, one can detect that the previous works need further investigations in other aspects such as deficiency and uncertainty of information. Therefore, the Rough set (RS) theory attracted some researchers as it is not considered just a target as fitness and operative mathematical gadget, to deal with deficient and uncertainty information, but also a proper active and effective paradigm for computing in order to realize feature selection [32], [33]. We have some examples such as Priyanga *et al.* [34] presented a novel filter-based FS method for the identification of relevant features using RS theory and Hyper-clique based Binary WOA (RS-HCBWoA), where the authors tested their technique on the dataset of the power system attack where the performance of RS-HCBWoA was evaluated according to the size of the reduction, precision, recall, the accuracy of classification, and time complexity. Sahu *et al.* [35] utilized the RS approach for aiding the student to choose a suitable subject and thus give a good service or contribution to the society, especially in such domain. The principal aim of it is to analyzing student's features regarding career, memory, knowledge, interest, environment, and attitude after that, predicting the suitable path for making the career comfortable where the student can conveniently explore much in that area. For choosing the career of the students, the authors proposed a hybridized distance measure under a picture fuzzy environment where the evaluating information regarding students, subjects, and student's features are given in picture fuzzy numbers. They introduced two types of hybridization approaches that are the hybridization of Hausdorff and Hamming distance measures and the hybridization of Hausdorff and Euclidean distance measures. Then, they applied the rough set theory for identifying if a certain subject is appropriate for a student even if there is controversy to select a path. Even though some of the proposed literature has shown remarkable contributions, merging the RS and offering an efficient algorithm for the FS approach is still challenging.

In the context of MH techniques, the Slime mould algorithm (SMA) has been recently proposed to mimic the natural conduct of the slime mould's oscillation. Its simplicity, sensitiveness for controlling parameters, and flexible structure attracted some researchers to implement and modify the optimizer to handle their applications. Kumar *et al.* [36] applied the basic SMA to get optimal parameters of the solar cell models. The SMA has been combined with an artificial neural network (ANN) to enhance the prediction of the demand of urban water [37]. The performance of the resultant combination between ANN and SMA exhibited better performance than traditional ANN. Precup *et al.* [38] presented SMA to tuning parameters of fuzzy controllers for the servo systems. Moreover, there are several fields, SMA has been applied such as photovoltaic modeling [39], [40], Bearing defect identification [41], dynamic monitoring health [42], feature selection [43], fuel cells [44] and others [45]–[49]. Furthermore, Chen *et al.* [50] proposed a chaotic variant for SMA to act as an SVR-based prediction approach. Hassan *et al.* [51] introduced an enhancement version of SMA by using a sine-cosine algorithm operator to solve the economic and emission dispatch (EED) problems. Recently, Rizk-Allah *et al.* [52] presented a chaos-opposition-developed slime mould algorithm (CO-SMA) for reducing the cost of the energy for the high-altitude sites, including wind turbines. In the proposed CO-SMA, the crossover-opposition methodology was conducted to boost the exploration phase. On the other hand, the chaotic search methodology was proposed to enhance the intensification phase of the classical SMA while handling the non-linear tasks [52]. Improving the SMA optimizer is a newly opened avenue that is why it is still several approaches can be proposed to modify the SMA performance. However, there is much room for further modifications for SMA to explore reinforced regions within search landscape and high-quality solutions. In this circumstance, the principal perspective of the 'no free lunch' (NFL) theorem [53] states that the universally preponderant optimizer that can behave with all outstanding optimization features in the best way does not exist. Therefore, based on the NFL, proposing novel algorithm fit is challenging to achieve a remarkable performance compared to state-of-the-art techniques.

In the context of MH modification approaches, a robust tool in mathematics named fractional calculus (FC) has been recently integrated into the MH to provide a memory of all past solutions to behave as a guide for the individuals while discovering the search space [54]. The FC proved its efficiency with MH in several applications: for example; FC with harries hawk optimizer (FMHHO) for fuel cell modeling [54], FC with cuckoo search (FOCS) for identifying the financial chaotic systems [55] and feature selection [1], furthermore, FC with Marian predator optimizer (FMPA) for COVID-19 image classification [56], Fractional-order Manta ray foraging optimizer (FOMRFO) for image segmentation [57], and fractional-order flower pollination (FOFPA) algorithm for image segmentation [58], moreover fraction-order firefly optimization algorithm for handling chaotic systems [59]. As the SMA suffers from the limitation of using the vector of the individual solutions only while updating the agents' positions, which the local operator knows. This strategy is not adequate to escape from the local solutions. Therefore, this motivated us to present an alternative version of SMA, which aims to enhance the updating process of the solutions through achieving reliable exploration and exploitation. Accordingly, we are attracted to adopt the remarkable features of FC with SMA to overcoming its drawback of the local operator via considering some historical knowledge about the agents' motions throughout the iterations. In addition, the degree of dependency is used as fitness function to assess the performance of each agent inside the population. This leads to improve the final classification process since the selected features are more depended on the target label.

Therefore, the main contribution of this study can be summarized as:

- Enhancing the ability of SMA using fractional calculus (FC) perspective; as a result, fractional-order SMA (FOSMA) is proposed. In addition, the rough set is used as part of the fitness value to increase the developed method's ability to tackle the uncertainty inside the dataset.
- Using the degree of dependency to evaluate the quality of selected features during the process of searching about the optimal subset of features.
- Assessing the performance of FOSMA using a set of UCI datasets.
- Evaluating the efficiency of the developed FOSMA through massive comparison with well-known FS state-of-art-techniques.

The present study is organized as: Section II presented the mathematical formulation of Slime mould algorithm (SMA), Rough set and Fractional Calculus. In section IV, the developed feature selection method is introduced. The results and discussion are given in section IV. The conclusion and future work are given in section V.

II. BACKGROUND

This section describes the basic concepts of the Slime mould algorithm (SMA), Rough set, and Fractional Calculus. Here each of the key steps is detailed and explained for better comprehension.

A. SLIME MOULD ALGORITHM

Li *et al.* [60] presented the algorithm of the Slime Mould Algorithm (SMA) as an efficient technique for global optimization.

The SMA imitates the slime mould's oscillation conduct in nature. The SMA procedures can be modeled mathematically as:

1) *The First Stage (The Food Approach):* The slime mould approach can be modeled through such stage as in the following:

$$Z = \begin{cases} Z_b + v_b \times (W \times Z_A - Z_B) & r < p \\ v_c \times Z & r \geq p \end{cases} \tag{1}$$

In Eq. (1), v_b can be determined through the interval $[-a, a]$ and v_c is reduced from 1 to 0. Z_b identifies the finest solution. Z_A and Z_B are arbitrary solutions, W exemplifies the slime mould weight. As well as p is determined by utilizing Eq. (2).

$$p = \tanh |S(i) - DF|, \quad i = 1, 2, \ldots, N \tag{2}$$

In Eq. 2, $S(i)$ exemplifies the fitness value of the solution Z and DF exemplifies the finest fitness value.

The value a that identifies v_b in Eq. (1) is determined as in Eq. (3).

$$a = arctanh\left(-\left(\frac{t}{max_t}\right)+1\right) \tag{3}$$

In addition, the value of W can be determined using the following formula:

$$W(S_{Ind}(i)) = \begin{cases} 1 + r \times \log\left(\frac{(b_F - S(i))}{(b_F - w_F)} + 1\right) & Cond \\ 1 - r \times \log\left(\frac{(b_F - S(i))}{(b_F - w_F)} 1\right) & otherwise \end{cases} \tag{4}$$

In Eq.(4), $Cond$ exemplifies that $S(i)$ has the rank of the population first half. $r \in [0, 1]$ is generated randomly. w_F and b_F denote the worst and the best local fitness value, respectively. S_{Ind} stores the sorted fitness values and it is defined as:

$$S_{Ind} = sort(S) \tag{5}$$

2) *The Second Stage (Wrap Food):* SMA simulated the slime mould modernized position. The next equation can be used for performing this modernization:

$$Z(t+1) = \begin{cases} LB + rand(UB - LB) & rand < z \\ Z_b(t) + v_b(WZ_A(t) - Z_B(t)) & r < p \\ v_c Z(t) & r \geq p \end{cases} \tag{6}$$

where LB and UB exemplify the search space lower and upper bounds. r and $rand$ are determined randomly through the interval [0, 1].

3) *The Third Stage (Oscillation):* The v_b value is modernized in the interval $[-a, a]$ and v_c in the range $[-1, 1]$.

B. FRACTIONAL CALCULUS CONCEPT

The Fractional calculus (FC) concept depends on several definitions like the Grunwald-Letnikov (GL) definition, and it can be mathematically modeled as podlubny1998fractional:

$$D^{\beta}(Z(t)) = \lim_{h \to 0} \frac{1}{h^{\beta}} \sum_{k=0}^{\infty} (-1)^k \binom{\beta}{k} Z(t - kh), \tag{7}$$

where

$$\binom{\beta}{k} = \frac{\Gamma(\beta+1)}{\Gamma(k+1)\Gamma(\beta-k+1)} = \frac{\beta(\beta-1)(\beta-2)\ldots(\beta-k+1)}{k!}, \tag{8}$$

where, $D^{\beta}(x(t))$ exemplifies the Grunwald-Letnikov (GL) derivative of kind fractional of order β. $\Gamma(t)$ identifies gamma function. The GL in Eq. (7) for discrete-time case can be formulated as:

$$D^{\beta}[Z(t)] = \frac{1}{T^{\beta}} \sum_{k=0}^{r} \frac{(-1)^k \Gamma(\beta+1) Z(t-kT)}{\Gamma(k+1)\Gamma(\beta-k+1)} \tag{9}$$

In Eq. (9), T denotes the period of sampling and r corresponds to the number of expressions from the memory. β identifies the derivative order coefficient.

In particular, while the derivative order has a special value of $\beta = 1$, the formula of Eq. (9) can be rewritten as:

$$D^1[Z(t+1)] = Z(t+1) - Z(t) \tag{10}$$

where $D^1[Z(t)]$ denotes the variation between two tailed juveniles.

C. ROUGH SETS

Recently, a new approach which had been used with optimization algorithms for enhancing their efficiency represented by the mechanism of the Rough set theory [61]–[65] which is utilized more for controlling the uncertainty in discovering the dependencies of data, evaluating the attributes prominence, detecting patterns in data, attribute reduction, and databases rules extraction which can be used as a classifier for unviewed datasets. Rough set is one of the techniques that depend on computational intelligence analysis which doesn't need extra parameters and just depends on the information present in the given data. It can decide if the data is perfect or can not depend on the same data. In the case of incomplete data, RS recommends more information of objects for reaching high efficiency of classification. Unlike that case, in the case of complete data, RS has the ability for identifying the minimum required data which is useful for big domain space, where this can decide the quality of data.

RS can be modeled mathematically as following. Let we have an information system $S = < U, A, V, f >$ where $U = \{x_1, \ldots, x_n\}$ exemplifies the universe of primitive objects, as well as A exemplifies the set of features [61]. In addition, $V = \bigcup_{a \in A} V_a$, where V_a exemplifies the value set of feature $a. f : U \times A \to V$ is an information function which puts the domains of features values for the objects like $\forall a \in A, x \in U$ and $f(a, x) \in V_a$.

The indiscernibility relation of $B \subset A$, symbolized as $IND(B)$ represents an equivalence relation shown as:

$$IND\,(B) = \{(x, y) \in U \times U: \forall a \in B, f\,(a, x) = f\,(a, y)\} \tag{11}$$

According to Eq. (11), the two objects x and y are indiscernible with respect to B, when $(x, y) \in IND(B)$. In addition, the set of all equivalence classes of $IND(B)$ is represented using $U/IND(B)$ (also called U/B) and it is defined as:

$$U/IND(B) = \{a \in B : U/IND(\{a\})\} \tag{12}$$

$$AB = \{X \bigcap Y : \forall X \in A, \forall Y \in B, X \bigcap Y \neq \emptyset\} \tag{13}$$

The two objects x and y are indiscernible based on the features of B, when $(x, y) \in U/IND(B)$, and the equivalence classes of the B-indiscernibility relation are represented by $[x]_B$. Consider $X \subseteq U$, the upper and lower approximations of X are formulated in Eq. (14) and Eq. (15), respectively:

$$\overline{B}(X) = \{[x]_B | \ [x]_B \bigcap X \neq \emptyset\} \tag{14}$$

$$\underline{B}(X) = \{[x]_B \mid [x]_B \subseteq X\} \tag{15}$$

The set of features A is formulated as $A = C \cup D$, where C and D denote the condition and decision features, respectively [61]. Thereafter, the positive region is given as:

$$POS_C(D) = \bigcup \underline{B}(X), \quad x \in U/D \tag{16}$$

Rough set reducts can be determined according to computing the degree of dependency ($\gamma_C(D)$) that defined as [61]:

$$\gamma_C(D) = |POS_C(D)| / |U| \tag{17}$$

In general, $\gamma_C(D)$ is applied to determine the significant features, and this performed by computing the equivalence relations obtained by using different features and determine the best of them.

III. DEVELOPED FRACTIONAL-ORDER SLIME MOULD ALGORITHM

Within such a section, the developed FS technique based on enhancing the performance of SMA using Fractional-order and Rough set is introduced. The structure of this section consists of two parts; the first part is to discuss the process of combining the SMA with Fractional-order. The second part is to discuss the general framework of the developed FS method.

A. FRACTIONAL-ORDER SLIME MOULD ALGORITHM

This section presents a detailed description of the innovative Fractional-order Slime Mould Algorithm. The fractional calculus property is integrated with the second line of Eq. (1), to boost the behavior of the individuals while discovering the search landscape. In the basic SMA, Li *et al.* [60] utilized the vector of the individual solutions at iteration (t) while computing the solutions of $(t + 1)$, which was known by the local operator. Utilizing the individual solutions only in updating their solutions is not adequate to escape from the local solutions. Therefore, in this work, we adopted FC properties to enhance this equation, to support the updating process of the solutions. The derivative of kind fractional-order has a remembrance of all previous values that behaved as a guide for the individuals while discovering the search space [54]. To model the second part/line of Eq. (1) in an adequate formula to be merged with fractional calculus, the following steps are followed:

First subtracting the term of $Z(t)$ from the two sides as in the following equation:

$$Z(t+1) - Z(t) = v_c.Z(t) - Z(t) \tag{18}$$

By inspecting the Eq. (18), one can note that the left-side of the equation is in the form of the particular case of Eq. (10). For generalizing Eq. (18) to include several remembrance expressions based on fractional calculus property the GL definition in a discrete-time formula of Eq. (9) is utilized with general derivative coefficient β at $T = 1$ as represented below in details:

$$D^{\beta}[Z(t+1)] = Z(t+1) - Z(t) = Z(t+1) + \sum_{n=1}^{r} \frac{(-1)^n \Gamma(\beta+1) Z(t+1-n)}{\Gamma(n+1)\Gamma(\beta-n+1)} \tag{19}$$

Use the formula of Eq. (19) then substitute in Eq. (18). The modified recipe of the updated solutions based on the fractional calculus is written as below for r terms of previous memory:

$$Z(t+1) = -\sum_{n=1}^{r} \frac{(-1)^k \Gamma(\beta+1) Z(t+1-k)}{\Gamma(k+1)\Gamma(\beta-k+1)} + v_c.Z(t) - Z(t) \tag{20}$$

If we use only first four memory terms ($r = 4$) in FOSMA with derivative coefficient β, the Eq. (20) of the velocity can be written as follows:

$$\begin{aligned} Z(t+1) = {} & \frac{1}{1!}\beta Z(t) + \frac{1}{2!}\beta(1-\beta)Z(t-1) \\ & + \frac{1}{3!}\beta(1-\beta)(2-\beta)Z(t-2) \\ & + \frac{1}{4!}\beta(1-\beta)(2-\beta)(3-\beta)Z(t-3) \\ & + v_c.Z(t) - Z(t) \end{aligned} \tag{21}$$

Hence the structure of the Wrap food phase of the basic SMA is modified to be as modeled in the following formula:

$$Z(t+1) = \begin{cases} LB + rand(UB - LB) & rand < z \\ Z_b(t) + v_b(WZ_A(t) - Z_B(t)) & r < p \\ Eq.(21) & r \geq p \end{cases} \tag{22}$$

The Algorithm 1 sums up the main structure of the proposed FOSMA where the optimizer starts with a set of randomly generated solutions. The previous old solutions have been initialized and stored in the memory. For a certain number of iterations, a set of operators are computed and the initial solutions are modified based on FOSMA main structure as reported in Algorithm 1. The memorized terms are updated at each iteration based on the first in first out approach as depicted in Fig.1. The implementation of the FOSMA is stopped while the final iterations number has been satisfied.

B. FRAMEWORK OF DEVELOPED FS METHOD

The main steps of the developed FS method based on the enhanced SMA along with Fractional-order are introduced in this section and depicted in Fig.2. The presented method, named FOSMA, starts by splitting the dataset into training and testing sets. Then the presented method sets the initial

Algorithm 1 FOSMA Algorithm

```
Input: N the solutions number and entire iterations number (t_max), the derivative order β, the memory length r.
Construct a random population (Z).
Store the initial values for the memory window with length (r).
t = 1.
while t <= t_max do
    For each Z_i compute the fitness value (F_i).
    Update the value of the finest solution Z_b.
    Calculate W using Eq. (14)
    for i = 1 : N do
        Modify variables of v_b, v_c, and p
        Use Eq. (22) to update the position of agents.
    end for
    Update memory window based on first in first out approach as depicted in Fig.1.
    t = t + 1.
end while
Return the best solution X_b.
```

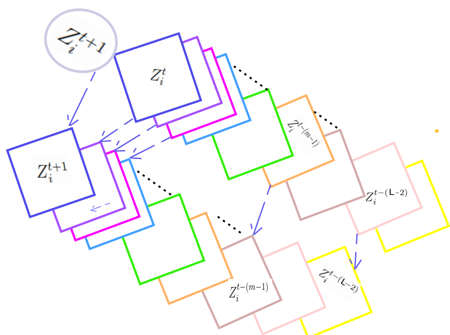


FIGURE 1. Memory update based on FIFO.

value for the population and computes the fitness value for each agent using the training set. This fitness value depends on computing the degree of dependency between the selected features and target label, as well as, the ratio of selected features. The next step is to find the agent that has the largest fitness value. This followed by updating the agent in the current population using the operator of FOSMA that discussed in Algorithm 1. The updating process the agents is conducted until reached the terminal conditions, then return by the best agent. The next step is to reduce the testing set according to the best agent and evaluate the performance measures. The stages of the developed method are given in the following sections.

1) INITIAL PHASE

The developed FS approach starts by dividing the input dataset into testing and training set which represents 20% and 80%, respectively. This followed by forming the initial population X which has N solutions and this modeled as:

$$Z_i = LB_i + rand \times (UB_i - LB_i) \tag{23}$$

In Eq. (23), UB_i and LB_i denote the limits of search domain.

2) UPDATING POPULATION PHASE

The presented method starting this phase by computing the fitness value for each agent. This achieved in two steps; the first step is to allocate the relevant and irrelevant features which corresponding to ones and zeros, respectively, in the Boolean version of the current agent. This process performed using the following equation.

$$BZ_{ij} = \begin{cases} 1 & if\ Z_{ij} > 0.5 \\ 0 & otherwise \end{cases} \tag{24}$$

The second step is to compute the degree of dependency based on the selected features from the training set and calculate the ratio of selected features. This formulated as maximization problem which defined as:

$$Fit_i = \lambda \times \gamma_i + (1 - \lambda) \times \left(1 - \frac{|BZ_i|}{Dim}\right), \tag{25}$$

where $(\frac{|BZ_i|}{Dim})$ represents the ratio of selected features. γ_i is the degree of dependency which defined in Eq. (17). Whereas, λ is the weight value of the two parts of Eq. (25).

Thereafter, the current population Z is updated using the operators of FOSMA which discussed in Algorithm 1. The presented FS method repeats its updating steps until reached the termination criteria.

3) EVALUATION PHASE

In this phase, the developed FS approach reduces the testing based on the Boolean version of the best agent by removing the irrelevant features. Then computing the performance of classification the testing set using KNN classifier. This classifier is applied since it is simple and easy to be implemented. The main framework of the proposed method is given in Figure 2.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experiments conducted in this paper are performed using twenty UCI datasets frank2010uci. The description of these datasets is given in Table 1. It can be noticed from this description that these datasets are collected from different real applications. From this table, it can be noticed the characteristics of the tested datasets which collected from different fields and has a different number of features and instances.

A. PERFORMANCE MEASURES

The performance of the FS methods is computed by using a set of metrics including Accuracy, and fitness value. The definition of these measures is formulated as:

- *Average Accuracy:* It is defined as the average of the rate of samples that classified correctly. It is formulated as in Eq. (26).

$$AVG_{Acc} = \frac{1}{N_r}\sum_{k=1}^{N_r} Acc_{Best}^k,$$
$$Acc_{Best}^k = \frac{TP + TN}{TP + FN + FP + TN} \tag{26}$$

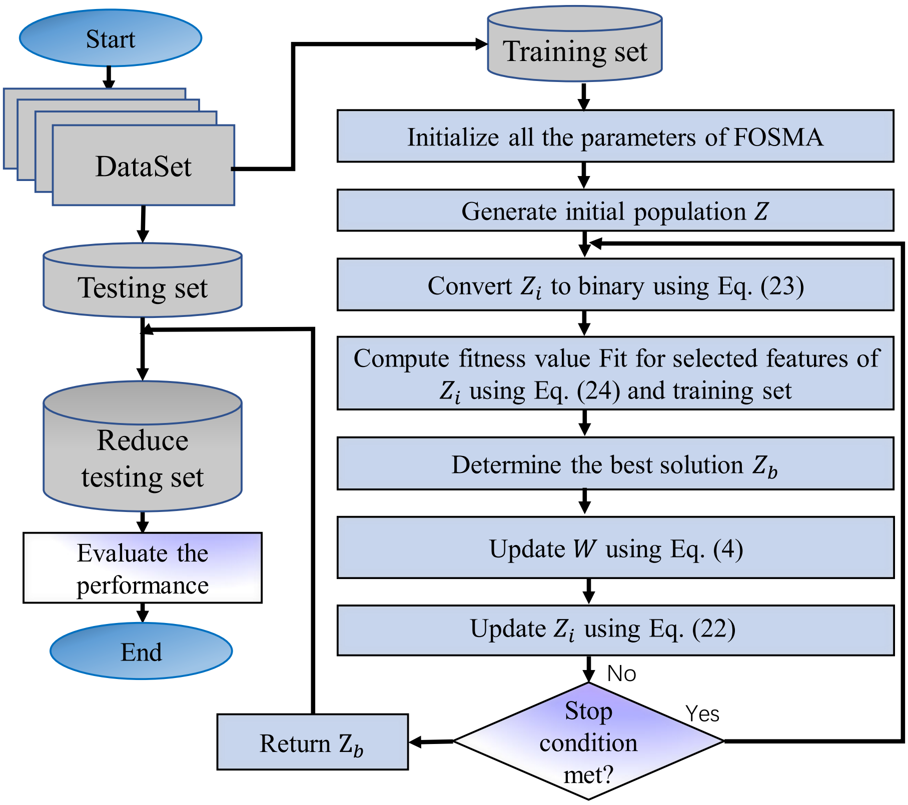


FIGURE 2. Framework for the presented FS method.

TABLE 1. Characteristic of datasets of Low dimensionality (LD) and High dimensionality (HD).

Datasets	Number of features	Number of instances	Number classes	Data of category	Dimension
Breastcancer (DS_1)	9	699	2	Biology	LD
BreastEW (DS_2)	30	569	2	Biology	LD
CongressEW(DS_3)	16	435	2	Politics	LD
Exactly(DS_4)	13	1000	2	Biology	LD
Exactly2(DS_5)	13	1000	2	Biology	LD
HeartEW(DS_6)	13	270	2	Biology	LD
IonosphereEW(DS_7)	34	351	2	Electromagnetic	LD
Lymphography(DS_8)	18	148	2	Biology	LD
M-of-n(DS_9)	13	1000	2	Biology	LD
PenglungEW(DS_{10})	325	73	2	Biology	LD
SonarEW(DS_{11})	60	208	2	Biology	LD
SpectEW(DS_{12})	22	267	2	Biology	LD
Tic-tac-toc(DS_{13})	9	958	2	Game	LD
Vote(DS_{14})	16	300	2	Politics	LD
WineEW(DS_{15})	13	178	3	Chemistry	LD
Zoo(DS_{16})	16	101	6	Artificial	LD
9_Tumors(DS_{17})	5726	60	9	Life	HD
Leukemia 2(DS_{18})	11225	72	3	Life	HD
Prostrate Tumors(DS_{19})	10509	102	2	Life	HD
warpAR10P(DS_{20})	2400	130	10	Face image	HD

where $N_r = 30$ is the total number of runs for each algorithm while, TN, TP, FN and FP are defined in Table 2.

- *Average Fitness Value:* It is defined as the average of the ability of the algorithm to minimize the number of selected features and simultaneously the error of classification. This formulated as:

$$AVG_{Fit} = \frac{1}{N_r} \sum_{k=1}^{N_r} Fit_{Best}^k \tag{27}$$

TABLE 2. Confusion matrix.

	Predected class	
Actual class	**Positive**	**Negative**
Postive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP	True Negative (TN)

- *Average of Relevant Features:* It computes the average of relevant features that gives the smallest fitness value and it is computed as:

$$AVG_{|BZ_b|} = \frac{1}{N_r} \sum_{k=1}^{N_r} \left| BZ_b^k \right| \tag{28}$$

In Eq. (28), $|BZ_b^k|$ denotes the number of features corresponding to ones in BZ_b^k at k^{th} run.

The results of the developed FOSMA are compared with other FS methods including Henry gas solubility optimization (HGSO), grey wolf optimization (GWO), Harris Hawks Optimization (HHO), and traditional SMA. The parameters of these methods are set according to the original implementation. In addition, the common parameter such as the maximum number of iterations is 50 and the size of the population is 10. Each one of these methods is conducted with 30 independent runs. The experiments are conducted using Matlab R2020b which installed on a PC with a Windows 10 64-bit, system of 3.40 GHz processor with 4GB RAM.

TABLE 3. Average of the accuracy obtained by each algorithm.

	bGWO	SMA	FOSMA	HHO	HGSO	WOA
DS_1	0.9414	0.9286	**0.9671**	0.9514	0.9514	0.9414
DS_2	**0.9298**	0.9246	0.9263	0.9193	0.9281	0.9070
DS_3	**0.9471**	0.9310	0.9172	0.9149	0.9149	0.9057
DS_4	0.9300	**0.9580**	0.9560	0.9310	0.8810	0.9270
DS_5	0.7010	0.6990	**0.7310**	0.7080	0.6930	0.6890
DS_6	0.6741	0.7037	0.6704	0.7000	0.7148	**0.7185**
DS_7	0.8761	0.8507	0.8225	0.8901	**0.8930**	0.8310
DS_8	0.6554	0.6600	**0.7333**	0.6713	0.6579	0.6867
DS_9	0.9470	**0.9920**	0.9710	0.9390	0.9230	0.9850
DS_{10}	0.5733	0.7705	**0.9200**	0.8231	0.5874	0.7619
DS_{11}	0.7429	**0.7810**	**0.7810**	0.7667	0.7381	0.7000
DS_{12}	0.7333	0.7148	**0.8222**	0.8037	0.7185	0.7148
DS_{13}	0.7729	**0.8344**	0.8125	0.8146	0.7729	0.7906
DS_{14}	**0.9767**	0.9033	0.9067	0.9200	0.9267	0.9200
DS_{15}	0.8611	0.8944	0.8611	0.8444	**0.9778**	0.9111
DS_{16}	0.9429	0.8857	0.9143	0.8667	0.9619	**0.9810**
DS_{17}	0.8500	0.9000	**0.9644**	0.9155	0.9231	0.9367
DS_{18}	0.8667	0.8667	**1.0000**	0.9333	0.9667	0.8667
DS_{19}	0.9048	0.9476	**0.9752**	0.9619	0.9143	0.9429
DS_{20}	0.6538	0.8615	0.9154	0.7308	0.8760	0.7308

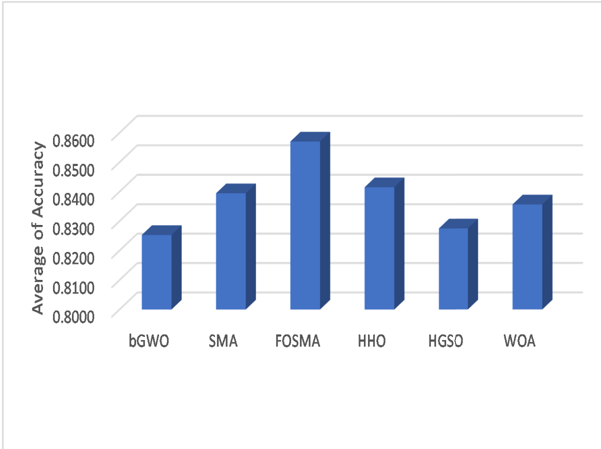
TABLE 4. Standard deviation of the accuracy obtained by each algorithm.

	bGWO	SMA	FOSMA	HHO	HGSO	WOA
DS_1	0.0137	0.0295	**0.0108**	0.0198	0.0155	0.0128
DS_2	0.0206	0.0295	0.0133	0.0169	**0.0039**	0.0192
DS_3	0.0252	0.0244	0.0221	0.0300	0.0192	**0.0096**
DS_4	0.0735	0.0525	**0.0373**	0.0744	0.0699	0.0703
DS_5	**0.0114**	0.0192	0.0307	0.0157	0.0217	0.0342
DS_6	0.0649	0.0655	**0.0401**	0.0779	0.0903	0.0479
DS_7	0.0461	0.0635	**0.0126**	0.0391	0.0354	0.0263
DS_8	0.0731	0.0548	0.0850	**0.0349**	0.0915	0.1017
DS_9	0.0669	**0.0084**	0.0292	0.0284	0.0463	0.0146
DS_{10}	0.0760	0.0750	**0.0298**	0.1570	0.0612	0.1116
DS_{11}	**0.0353**	0.0593	0.0742	0.0682	0.0558	0.0494
DS_{12}	**0.0336**	0.0465	0.0361	0.0361	0.0401	0.0549
DS_{13}	0.0267	**0.0093**	0.0285	0.0280	0.0282	0.0170
DS_{14}	0.0149	0.0298	0.0190	**0.0139**	0.0253	0.0506
DS_{15}	0.0878	0.0569	0.0393	0.0576	**0.0124**	0.0719
DS_{16}	0.0398	**0.0261**	0.0398	0.0398	0.0621	0.0261
DS_{17}	0.1414	**0.0314**	0.0400	0.1643	0.0786	0.0536
DS_{18}	**0.0000**	**0.0000**	0.0943	0.4243	0.0000	0.2828
DS_{19}	0.0673	0.0337	**0.0317**	0.1347	0.2357	0.0337
DS_{20}	0.0437	0.0742	**0.0000**	0.0801	0.0165	0.0588

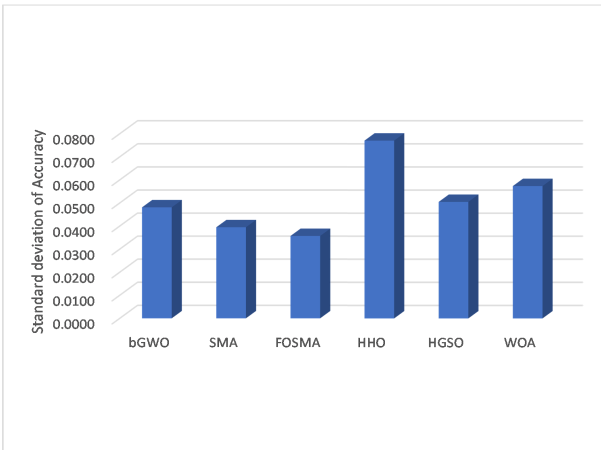
B. RESULTS AND DISCUSSION

The comparison results between the developed FOSMA method and other methods are given in Tables 3-8. It can be noticed from the average accuracy results that the developed FOSMA has the ability to improve the classification accuracy of nine datasets. Followed by traditional SMA which has the best accuracy at four datasets and GWO established its performance at three datasets. In addition, by analysis, the stability of the developed FOSMA represented in Table 4. One can be observed the stability of the developed FOSMA is better than other models. Whereas, the GWO and SMA nearly have the same performance in terms of standard deviation since SMA and GWO provides a better standard deviation at four and three datasets, respectively. Figure 3 depicts the average of accuracy and standard deviation overall for the tested datasets. One can be observed from these results that the FOSMA model has the largest average of accuracy with the smallest standard deviation. This indicates the high effect of FO on improving the performance of SMA.

Moreover, Table 5 depicts the average of fitness value for each method and it can be noticed from these results that the fitness value of the FOSMA is the best at seven datasets. While HGSO has largest fitness value at four datasets, followed by HHO, SMA and bGWO that have the higher average of fitness at three datasets. Whereas, WOA provides better fitness value at only one dataset. From the results of the standard deviation of fitness value, as provided in Table 6, it can be seen from these results that the FOSMA is more stable than other models which provide the smallest standard deviation at eleven datasets. Followed by HGSO and SMA which provide better standard deviation at seven and two datasets, respectively. Moreover, by analyzing the results of each algorithm in terms of best fitness value it can be seen that nearly HHO, and HGSO have the same performance which provides the best fitness value at four datasets. Whereas, FOSMA has better results at ten datasets. Finally, from



(a) Average of Accuracy



(b) Standard deviation of Accuracy

FIGURE 3. Comparison results between FOSMA and other models in terms of accuracy.

Table 8 that shows the worst fitness value obtained by each algorithm, it can be seen that the bGWO, FOSMA, HHO, and HGSO are better than other methods at five datasets.

TABLE 5. Average of fitness value for each method.

	bGWO	SMA	FOSMA	HHO	HGSO	WOA
DS_{1}	0.6835	0.6835	0.6819	0.6975	**0.7145**	0.6928
DS_{2}	0.6867	0.6867	0.7009	0.6767	0.6848	0.6849
DS_{3}	**0.7431**	**0.7431**	0.7191	0.7197	0.7220	0.7014
DS_{4}	0.8154	0.8154	0.8077	0.8308	**0.8692**	0.8308
DS_{5}	0.9359	0.9359	**0.9399**	0.9363	0.9308	0.9154
DS_{6}	0.7937	0.7937	0.7893	**0.8021**	0.7963	0.7979
DS_{7}	0.6264	0.6264	0.6249	0.6236	0.6086	**0.6483**
DS_{8}	0.6917	0.6917	**0.7288**	0.6938	0.6950	0.6963
DS_{9}	0.8462	0.8462	0.8000	0.8538	**0.8692**	0.8097
DS_{10}	0.6031	0.6031	**0.6868**	0.5265	0.5102	0.5622
DS_{11}	0.5933	0.5933	**0.6417**	0.5945	0.5778	0.5953
DS_{12}	0.8593	0.8593	0.8560	**0.8834**	0.8524	0.8673
DS_{13}	**0.9001**	**0.9001**	0.8918	0.8957	0.8965	0.8928
DS_{14}	0.7475	0.7475	0.7113	**0.7500**	0.7092	0.7250
DS_{15}	**0.9293**	0.8500	0.8966	0.8511	0.8160	0.7966
DS_{16}	0.7050	0.7050	**0.7125**	0.6888	0.7088	0.7063
DS_{17}	0.8896	0.8087	0.9995	0.9984	**0.9996**	0.9997
DS_{18}	0.8920	0.8274	**0.9996**	0.9984	0.9975	0.9996
DS_{19}	0.8934	0.8231	**0.9999**	0.9991	0.9862	0.9943
DS_{20}	0.5793	**0.8384**	0.8265	0.6060	0.7266	0.7401

TABLE 6. Standard deviation of fitness value for each method.

	bGWO	SMA	FOSMA	HHO	HGSO	WOA
DS_{1}	0.0154	0.0249	**0.0057**	0.0253	0.0176	0.0185
DS_{2}	0.0125	0.0252	**0.0103**	0.0143	0.0192	0.0277
DS_{3}	0.0330	0.0321	0.0166	0.0183	**0.0106**	0.0376
DS_{4}	0.0501	**0.0344**	0.0385	0.0439	0.0439	0.0439
DS_{5}	0.0176	0.0261	**0.0153**	0.0181	0.0172	0.0172
DS_{6}	0.0099	0.0171	**0.0063**	0.0196	0.0115	0.0076
DS_{7}	0.0068	0.0213	0.0186	0.0146	**0.0054**	0.0371
DS_{8}	0.0230	0.0143	**0.0079**	0.0183	0.0119	0.0204
DS_{9}	0.0385	0.0471	**0.0172**	0.0172	0.0439	0.0211
DS_{10}	0.0054	0.0082	0.0042	0.0151	**0.0014**	0.0509
DS_{11}	0.0109	0.0831	**0.0059**	0.0355	0.0063	0.0497
DS_{12}	0.0208	0.0129	0.0112	0.0076	**0.0042**	0.0141
DS_{13}	0.0102	0.0007	**0.0006**	0.0080	0.0075	0.0085
DS_{14}	0.0107	**0.0100**	0.0144	0.0351	0.0119	0.0321
DS_{15}	0.01437	0.01571	0.0588	0.01392	0.01327	0.03899
DS_{16}	0.0396	0.0440	0.0169	0.0248	**0.0137**	0.0218
DS_{17}	0.0009	0.0077	**0.0000**	0.0020	**0.0000**	**0.0000**
DS_{18}	0.0002	0.0049	**0.0000**	0.0001	0.0009	0.0005
DS_{19}	0.0022	0.0087	**0.0000**	0.0001	0.0187	0.0005
DS_{20}	**0.0622**	0.0794	0.0825	0.0623	**0.0888**	0.0790

Followed by SMA and WOA which have the best value at four datasets. From Figure 5 shows the average of each algorithm overall the tested datasets. It can be observed that FOSMA is the best overall other algorithms among the average, worst, and best of fitness value. In addition, it is more stable than other methods.

To analysis the behaviour of the developed FOSMA to reduce the number of features with preserving the quality of classification, the average of selected features is given in Table 9. It can be seen from the results, the efficiency of FOSMA to select the smallest number of features among the tested twenty datasets. For example, the developed FOSMA has the best average of selected features at ten datasets which represents nearly, the 50% from the tested datasets. In addition, the HGSO and SMA provide results better than other algorithm, since each of them has smallest number of selected features at six and four datasets, respectively. The average of selected features overall the tested is given in Figure 4 that indicates the high efficiency of FOSMA to reduce the number of features.

TABLE 7. Best of fitness value for each method.

	bGWO	SMA	FOSMA	HHO	HGSO	WOA
DS_{1}	0.7007	0.7007	0.6953	**0.7347**	0.7294	0.7240
DS_{2}	0.6998	0.6998	**0.7419**	0.6995	0.7172	0.7116
DS_{3}	0.7759	0.7759	**0.8085**	0.7364	0.7320	0.7474
DS_{4}	0.8846	0.8846	0.8846	0.8846	**0.9231**	0.8462
DS_{5}	0.9562	0.9562	0.9408	0.9562	**0.9615**	0.9512
DS_{6}	0.8063	0.8063	0.8132	**0.8332**	0.8118	**0.8021**
DS_{7}	0.6333	0.6333	**0.6912**	0.6466	0.6154	0.6524
DS_{8}	0.7283	0.7283	**0.7476**	0.7156	0.7114	**0.7288**
DS_{9}	0.8846	0.8846	0.8462	0.8846	**0.9231**	0.8308
DS_{10}	0.6123	0.6123	0.6915	0.5477	0.5123	**0.6923**
DS_{11}	0.6000	0.6000	**0.6833**	0.6333	0.5870	0.6500
DS_{12}	0.8783	0.8783	**0.8985**	0.8907	0.8595	0.8812
DS_{13}	**0.9137**	**0.9137**	0.9111	0.9098	0.9098	0.8959
DS_{14}	0.7604	0.7604	0.7813	**0.7896**	0.7250	0.7475
DS_{15}	**0.9293**	0.8799	0.9006	0.8873	0.8262	0.8190
DS_{16}	0.7563	0.7563	**0.7625**	0.7188	0.7188	0.7325
DS_{17}	0.8902	0.8142	0.9983	0.9998	0.9925	0.9997
DS_{18}	0.8922	0.8309	0.9900	0.9999	0.9813	1.0000
DS_{19}	0.8949	0.8292	0.9857	0.9999	0.9794	0.9998
DS_{20}	0.5524	0.7206	0.8056	0.5945	**0.6752**	0.6745

TABLE 8. Worst of fitness value for each method.

	bGWO	SMA	FOSMA	HHO	HGSO	WOA
DS_{1}	0.6675	0.6675	0.6730	0.6774	0.6693	**0.6792**
DS_{2}	0.6665	0.6665	**0.6817**	0.6630	0.6694	0.6727
DS_{3}	0.6983	0.6983	**0.7119**	0.6936	0.7047	0.6782
DS_{4}	0.7692	0.7692	0.7692	0.7692	**0.8077**	0.7692
DS_{5}	**0.9231**	**0.9231**	**0.9231**	**0.9231**	**0.9231**	0.8846
DS_{6}	0.7840	0.7840	0.7831	0.7831	0.7794	**0.7947**
DS_{7}	**0.6176**	**0.6176**	0.6029	0.6065	0.6003	0.6025
DS_{8}	0.6751	0.6751	0.6836	0.6685	0.6794	0.6920
DS_{9}	0.8077	0.8077	0.7692	**0.8462**	0.8077	0.8077
DS_{10}	0.5985	0.5985	**0.6400**	0.5123	0.5092	0.5185
DS_{11}	0.5750	0.5750	**0.6333**	0.5667	0.5711	0.5667
DS_{12}	0.8258	0.8258	0.8391	**0.8751**	0.8485	0.8649
DS_{13}	**0.8928**	**0.8928**	0.8915	0.8915	**0.8928**	0.8915
DS_{14}	**0.7354**	**0.7354**	0.6958	0.7104	0.6958	0.6958
DS_{15}	**0.9293**	0.8250	0.8319	0.8251	0.8022	0.4733
DS_{16}	0.6563	0.6563	0.6800	0.6563	**0.6938**	0.6875
DS_{17}	0.8890	0.8033	0.9998	0.9969	0.9998	0.9997
DS_{18}	0.8919	0.8240	1.0000	0.9998	0.9984	0.9992
DS_{19}	0.8919	0.8170	0.9999	0.9998	0.9887	0.9990
DS_{20}	0.6925	0.8763	0.8962	**0.8716**	0.7478	0.7541

To provide more evidence about the high efficiency of the FOSMA and the impact of FO on enhancing the behavior of SMA, a non-parametric test named the Friedman test is used. This test aims to determine if the obtained results by FOSMA are significantly different from results obtained using other methods or not. This test is performed at a significant level of 0.05 and the obtained results (i.e., mean rank) are given in Table 10. From these results, it can be seen that there is no significant difference between the competitive algorithms in terms of accuracy since P-value for the average (standard deviation) of accuracy is 0.1113 (0.2394) which is greater than 0.05. However, the FOSMA has the best mean rank in average and standard deviation of accuracy. Moreover, the

TABLE 9. Comparison results between FOSMA and other algorithms in terms of selected features.

	bGWO	SMA	FOSMA	HHO	HGSO	WOA
DS1	2.8	2.4	**2.2**	3.2	3.6	3.2
DS2	9.0	8.6	9.6	8.4	**8.0**	10.8
DS3	6.4	**5.0**	5.2	6.0	6.0	7.2
DS4	8.2	8.0	**6.6**	8.6	9.6	8.6
DS5	10.8	10.8	**9.0**	11.0	11.2	10.8
DS6	6.6	6.8	**5.0**	7.0	6.2	7.0
DS7	7.6	8.2	8.4	7.8	**6.0**	9.6
DS8	6.2	**5.6**	6.6	6.0	5.8	7.2
DS9	9.0	7.8	**6.4**	9.2	9.6	8.6
DS10	67.0	121.4	110.4	17.2	**6.6**	40.4
DS11	11.2	17.0	14.6	11.2	**7.6**	11.0
DS12	8.0	8.6	8.8	9.0	**7.8**	8.8
DS13	8.0	**6.0**	**6.0**	**6.0**	7.0	8.0
DS14	6.0	5.4	5.8	7.0	**5.2**	6.6
DS15	3.6	4.0	**3.4**	4.4	4.2	4.8
DS16	5.2	**4.8**	5.4	5.6	6.0	6.8
DS17	2257.0	2190.0	**2181.5**	2353.4	2771.2	2821.6
DS18	**2420.0**	3797.0	2523.0	3988.7	5304.7	5459.9
DS19	5085.9	3589.0	**2208.0**	3327.9	5101.8	5163.3
DS20	670.0	586.0	**488.0**	823.0	652.0	726.3

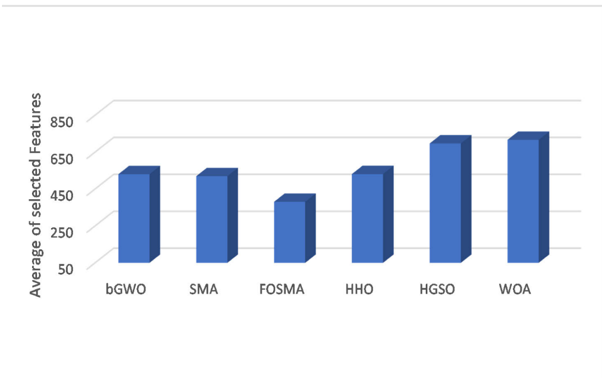


FIGURE 4. Average of selected features.

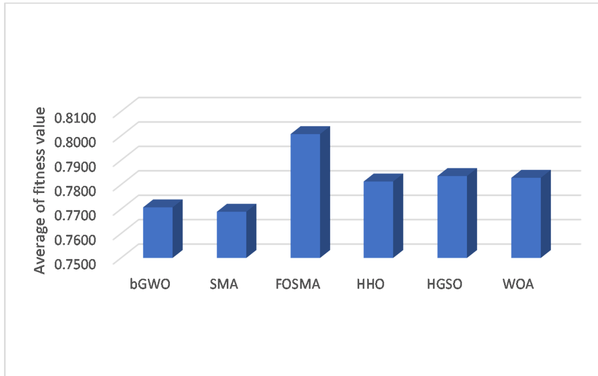
TABLE 10. Mean rank for each algorithm.

		bGWO	SMA	FOSMA	HHO	HGSO	WOA
Accuracy	Average	2.93	3.50	4.50	3.55	3.48	3.05
	STD	3.55	3.35	2.73	4.23	3.60	3.55
Fitness value	Average	3.33	3.28	3.78	3.73	3.20	3.70
	STD	3.70	4.40	2.13	3.68	2.88	4.23
	Worst	3.33	3.23	4.03	3.45	3.38	3.60
	Best	3.23	3.13	4.85	3.73	2.93	3.15
Number of selected feature		3.30	2.75	2.53	4.03	3.43	4.98

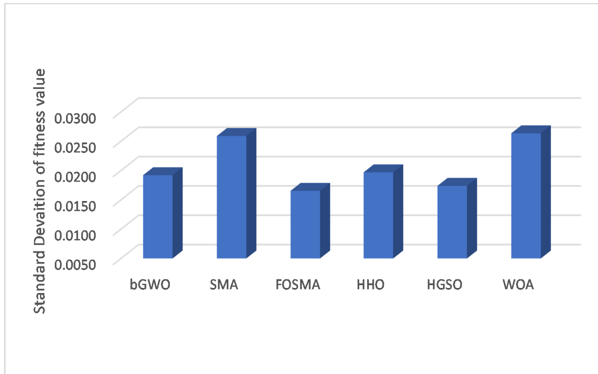
FOSMA has the first mean rank in terms of best, worst, and standard deviation of fitness value. However, at average of fitness value it can be seen that HHO allocates the first rank followed by FOSMA. In addition, the FOSMA has the smallest mean rank in terms of number of selected features, followed by SMA.

V. CONCLUSION

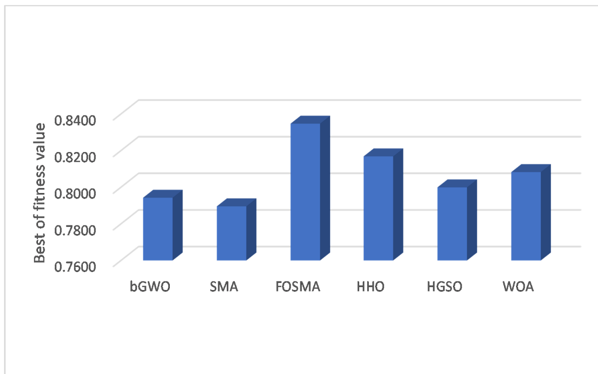
In this paper, a modified version of the Slime Mould Algorithm (SMA) has been developed as a feature selection



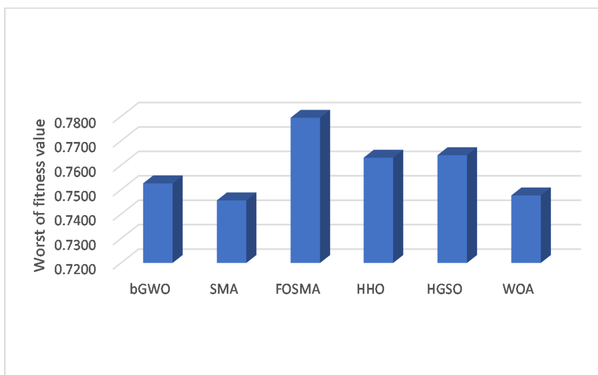
(a) Average of Fitness value



(b) Standard deviation of Fitness value



(c) Best of Fitness value



(d) Worst of Fitness value

FIGURE 5. Comparison results between FOSMA and other models in terms of fitness value.

method. This modification has been performed using the fractional calculus (FC) and Rough set. Each of these techniques has its task to improve SMA, such as FC has been applied to enhance the searchability of the agents of the classical SMA during the search process through counting several terms from memory based on FC perspective. Whereas the Rough set is applied to assess the quality of selected features since it can deal with uncertainty in the dataset. The results of FOSMA using twenty UCI datasets have been compared with other FS methods such as Henry gas solubility optimization (HGSO), grey wolf optimization (GWO), Harris Hawks Optimization (HHO), and traditional SMA. The obtained results show the ability of FOSMA which has a high capacity to increase the classification accuracy by reducing the number of relevant features. In addition, its results are better than other MH techniques used as FS methods.

Besides the efficiency of the FOSMA, it can be used in future works in different applications such as cloud computing, security, image segmentation, and other fields. In addition, it can be re-implemented as a multi-objective technique.

ACKNOWLEDGMENT

This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Program.

REFERENCES

[1] D. Yousri, M. Abd Elaziz, L. Abualigah, D. Oliva, M. A. A. Al-qaness, and A. A. Ewees, "COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions," *Appl. Soft Comput.*, vol. 101, Mar. 2021, Art. no. 107052.

[2] M. S. Satu, M. Z. Abedin, S. Khanom, J. Ouenniche, and M. S. Kaiser, "Application of feature engineering with classification techniques to enhance corporate tax default detection performance," in *Proc. Int. Conf. Trends Comput. Cogn. Eng.* Singapore: Springer, 2021, pp. 53–63.

[3] S. S. Sarkar, K. H. Sheikh, A. Mahanty, K. Mali, A. Ghosh, and R. Sarkar, "A harmony search-based wrapper-filter feature selection approach for microstructural image classification," *Integrating Mater. Manuf. Innov.*, vol. 10, no. 1, pp. 1–19, 2021.

[4] W. Li, L. Chen, J. Zhao, and W. Wang, "Embedded feature selection based on relevance vector machines with an approximated marginal likelihood and its industrial application," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jan. 25, 2021, doi: 10.1109/TSMC.2021.3049597.

[5] O. Elsherbiny, Y. Fan, L. Zhou, and Z. Qiu, "Fusion of feature selection methods and regression algorithms for predicting the canopy water content of rice based on hyperspectral data," *Agriculture*, vol. 11, no. 1, p. 51, Jan. 2021.

[6] R. A. Ibrahim, M. Abd Elaziz, A. A. Ewees, M. El-Abd, and S. Lu, "New feature selection paradigm based on hyper-heuristic technique," *Appl. Math. Model.*, vol. 98, pp. 14–37, Oct. 2021.

[7] M. Abd Elaziz, Y. S. Moemen, A. E. Hassanien, and S. Xiong, "Toxicity risks evaluation of unknown FDA biotransformed drugs based on a multi-objective feature selection approach," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 105509.

[8] L. Abualigah, A. H. Gandomi, M. A. Elaziz, H. A. Hamad, M. Omari, M. Alshinwan, and A. M. Khasawneh, "Advances in meta-heuristic optimization algorithms in big data text clustering," *Electronics*, vol. 10, no. 2, p. 101, 2021.

[9] L. Abualigah, A. H. Gandomi, M. A. Elaziz, A. G. Hussien, A. M. Khasawneh, M. Alshinwan, and E. H. Houssein, "Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis," *Algorithms*, vol. 13, no. 12, p. 345, 2020.

[10] A. Dahou, M. A. Elaziz, J. Zhou, and S. Xiong, "Arabic sentiment classification using convolutional neural network and differential evolution algorithm," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–16, Feb. 2019.

[11] E. Armengol, D. Boixader, A. García-Cerdaña, and J. Recasens, "T-generable indistinguishability operators and their use for feature selection and classification," *Fuzzy Sets Syst.*, vol. 360, pp. 33–48, Apr. 2019.

[12] B. Liu, Y. Chai, Y. Liu, C. Huang, Y. Wang, and Q. Tang, "Industrial process fault detection based on deep highly-sensitive feature capture," *J. Process Control*, vol. 102, pp. 54–65, Jun. 2021.

[13] V. N. Gopal, F. Al-Turjman, R. Kumar, L. Anand, and M. Rajesh, "Feature selection and classification in breast cancer prediction using IoT and machine learning," *Measurement*, vol. 178, Jun. 2021, Art. no. 109442.

[14] F.-J. Lin, S.-G. Chen, and C.-W. Hsu, "Intelligent backstepping control using recurrent feature selection fuzzy neural network for synchronous reluctance motor position servo drive system," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 3, pp. 413–427, Mar. 2019.

[15] S. Kashef and H. Nezamabadi-pour, "A label-specific multi-label feature selection algorithm based on the Pareto dominance concept," *Pattern Recognit.*, vol. 88, pp. 654–667, Apr. 2019.

[16] Z. Manbari, F. AkhlaghianTab, and C. Salavati, "Hybrid fast unsupervised feature selection for high-dimensional data," *Expert Syst. Appl.*, vol. 124, pp. 97–118, Jun. 2019.

[17] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[18] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.

[19] F. Merrikh-Bayat, "The runner-root algorithm: A Metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature," *Appl. Soft Comput.*, vol. 33, pp. 292–303, Aug. 2015.

[20] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 376, Apr. 2021, Art. no. 113609.

[21] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-qaness, and A. H. Gandomi, "Aquila optimizer: A novel meta-heuristic optimization algorithm," *Comput. Ind. Eng.*, vol. 157, Jul. 2021, Art. no. 107250.

[22] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2006.

[23] S. C. Wang, "Genetic algorithm," in *Interdisciplinary Computing in Java Programming*. Boston, MA, USA: Springer, 2003, pp. 101–116.

[24] Z.-J. Teng, J.-L. Lv, and L.-W. Guo, "An improved hybrid grey wolf optimization algorithm," *Soft Comput.*, vol. 23, no. 15, pp. 6617–6631, 2019.

[25] A. K. Shukla, P. Singh, and M. Vardhan, "Gene selection for cancer types classification using novel hybrid metaheuristics approach," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100661.

[26] X. Diao, J. Jiang, G. Shen, Z. Chi, Z. Wang, L. Ni, A. Mebarki, H. Bian, and Y. Hao, "An improved variational mode decomposition method based on particle swarm optimization for leak detection of liquid pipelines," *Mech. Syst. Signal Process.*, vol. 143, Sep. 2020, Art. no. 106787.

[27] R. A. Ibrahim, M. A. Elaziz, and S. Lu, "Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization," *Expert Syst. Appl.*, vol. 108, pp. 1–27, Oct. 2018.

[28] Too, Abdullah, and M. Saad, "A new quadratic binary Harris hawk optimization for feature selection," *Electronics*, vol. 8, no. 10, p. 1130, Oct. 2019.

[29] W. Fu, K. Zhang, K. Wang, B. Wen, P. Fang, and F. Zou, "A hybrid approach for multi-step wind speed forecasting based on two-layer decomposition, improved hybrid DE-HHO optimization and KELM," *Renew. Energy*, vol. 164, pp. 211–229, Feb. 2021.

[30] R. Al-Wajih, S. J. Abdulkadir, N. Aziz, Q. Al-Tashi, and N. Talpur, "Hybrid binary grey wolf with Harris hawks optimizer for feature selection," *IEEE Access*, vol. 9, pp. 31662–31677, 2021.

[31] T. Thaher, M. Saheb, H. Turabieh, and H. Chantar, "Intelligent detection of false information in Arabic tweets utilizing hybrid Harris hawks based feature selection and machine learning models," *Symmetry*, vol. 13, no. 4, p. 556, Mar. 2021.

[32] J. Zhang, J. Miao, K. Zhao, and Y. Tian, "Multi-task feature selection with sparse regularization to extract common and task-specific features," *Neurocomputing*, vol. 340, pp. 76–89, May 2019.

[33] F. Nie, S. Yang, R. Zhang, and X. Li, "A general framework for auto-weighted feature selection via global redundancy minimization," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2428–2438, Dec. 2018.

[34] S. Priyanga, M. R. Gauthama Raman, S. S. Jagtap, N. Aswin, K. Kirthivasan, and V. S. Shankar Sriram, "An improved rough set theory based feature selection approach for intrusion detection in SCADA systems," *J. Intell. Fuzzy Syst.*, vol. 36, no. 5, pp. 3993–4003, May 2019.
[35] R. Sahu, S. R. Dash, and S. Das, "Career selection of students using hybridized distance measure based on picture fuzzy set and rough set theory," *Decis. Making: Appl. Manage. Eng.*, vol. 4, no. 1, pp. 104–126, Mar. 2021.
[36] C. Kumar, T. D. Raj, M. Premkumar, and T. D. Raj, "A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters," *Optik*, vol. 223, Dec. 2020, Art. no. 165277.
[37] S. L. Zubaidi, I. H. Abdulkareem, K. S. Hashim, H. Al-Bugharbee, H. M. Ridha, S. K. Gharghan, F. F. Al-Qaim, M. Muradov, P. Kot, and R. Al-Khaddar, "Hybridised artificial neural network model with slime mould algorithm: A novel methodology for prediction of urban stochastic water demand," *Water*, vol. 12, no. 10, p. 2692, Sep. 2020.
[38] R.-E. Precup, R.-C. David, R.-C. Roman, E. M. Petriu, and A.-I. Szedlak-Stinean, "Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems," *Int. J. Comput. Intell. Syst.*, vol. 14, no. 1, pp. 1042–1052, 2021.
[39] M. Mostafa, H. Rezk, M. Aly, and E. M. Ahmed, "A new strategy based on slime mould algorithm to extract the optimal model parameters of solar PV panel," *Sustain. Energy Technol. Assessments*, vol. 42, Dec. 2020, Art. no. 100849.
[40] D. Yousri, M. Abd Elaziz, D. Oliva, L. Abualigah, M. A. A. Al-qaness, and A. A. Ewees, "Reliable applied objective for identifying simple and detailed photovoltaic models using modern metaheuristics: Comparative study," *Energy Convers. Manage.*, vol. 223, Nov. 2020, Art. no. 113279.
[41] G. Vashishtha, S. Chauhan, M. Singh, and R. Kumar, "Bearing defect identification by swarm decomposition considering permutation entropy measure and opposition-based slime mould algorithm," *Measurement*, vol. 178, Jun. 2021, Art. no. 109389.
[42] S. Tiachacht, S. Khatir, C. L. Thanh, R. V. Rao, S. Mirjalili, and M. Abdel Wahab, "Inverse problem for dynamic structural health monitoring based on slime mould algorithm," *Eng. with Comput.*, pp. 1–24, Mar. 2021.
[43] A. A. Ewees, L. Abualigah, D. Yousri, Z. Y. Algamal, M. A. A. Al-qaness, R. A. Ibrahim, and M. Abd Elaziz, "Improved slime mould algorithm based on firefly algorithm for feature selection: A case study on QSAR model," *Eng. with Comput.*, pp. 1–15, Mar. 2021.
[44] A. M. Othman and A. A. El-Fergany, "Optimal dynamic operation and modeling of parallel connected multi-stacks fuel cells with improved slime mould algorithm," *Renew. Energy*, vol. 175, pp. 770–782, Sep. 2021.
[45] R. Djekidel, B. Bentouati, M. S. Javaid, H. R. E. H. Bouchekara, A. S. Bayoumi, and R. A. El-Sehiemy, "Mitigating the effects of magnetic coupling between HV transmission line and metallic pipeline using slime mould algorithm," *J. Magn. Magn. Mater.*, vol. 529, Jul. 2021, Art. no. 167865.
[46] D. I. zci and S. Ekinci, "Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller," *Electrica*, vol. 21, no. 1, pp. 151–159, 2021.
[47] M. K. Naik, R. Panda, and A. Abraham, "Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm," *J. King Saud Univ.-Comput. Inf. Sci.*, Nov. 2020.
[48] A. Draz, M. M. Elkholy, and A. A. El-Fergany, "Slime mould algorithm constrained by the relay operating time for optimal coordination of directional overcurrent relays using multiple standardized tripping curves," *Neural Comput. Appl.*, vol. 33, pp. 11875–11887, Mar. 2021.
[49] T. Gush, C.-H. Kim, S. Admasie, J.-S. Kim, and J.-S. Song, "Optimal smart inverter control for PV and BESS to improve PV hosting capacity of distribution networks using slime mould algorithm," *IEEE Access*, vol. 9, pp. 52164–52176, 2021.
[50] Z. Chen and W. Liu, "An efficient parameter adaptive support vector regression using K-means clustering and chaotic slime mould algorithm," *IEEE Access*, vol. 8, pp. 156851–156862, 2020.
[51] M. H. Hassan, S. Kamel, L. Abualigah, and A. Eid, "Development and application of slime mould algorithm for optimal economic emission dispatch," *Expert Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115205.
[52] R. M. Rizk-Allah, A. E. Hassanien, and D. Song, "Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites," *ISA Trans.*, 2021.
[53] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[54] D. Yousri, S. Mirjalili, J. A. T. Machado, S. B. Thanikanti, O. Elbaksawi, and A. Fathy, "Efficient fractional-order modified Harris hawks optimizer for proton exchange membrane fuel cell modeling," *Eng. Appl. Artif. Intell.*, vol. 100, Apr. 2021, Art. no. 104193.
[55] D. Yousri and S. Mirjalili, "Fractional-order cuckoo search algorithm for parameter identification of the fractional-order chaotic, chaotic with noise and hyper-chaotic financial systems," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103662.
[56] A. T. Sahlol, D. Yousri, A. A. Ewees, M. A. A. Al-qaness, R. Damasevicius, and M. A. Elaziz, "COVID-19 image classification using deep features and fractional-order marine predators algorithm," *Sci. Rep.*, vol. 10, no. 1, pp. 1–15, Dec. 2020.
[57] M. Abd Elaziz, D. Yousri, M. A. Al-qaness, A. M. AbdelAty, A. G. Radwan, and A. A. Ewees, "A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation," *Eng. Appl. Artif. Intell.*, vol. 98, Feb. 2021, Art. no. 104105.
[58] D. Yousri, M. A. Elaziz, and S. Mirjalili, "Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation," *Knowl.-Based Syst.*, vol. 197, Jun. 2020, Art. no. 105889.
[59] Y. Mousavi and A. Alfi, "Fractional calculus-based firefly algorithm applied to parameter estimation of chaotic systems," *Chaos, Solitons Fractals*, vol. 114, pp. 202–215, Sep. 2018.
[60] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
[61] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, "Rough sets," *Commun. ACM*, vol. 38, no. 11, pp. 88–95, 1995.
[62] M. A. El Aziz and A. E. Hassanien, "Modified cuckoo search algorithm with rough sets for feature selection," *Neural Comput. Appl.*, vol. 29, no. 4, pp. 925–934, Feb. 2018.
[63] M. A. El Aziz and A. E. Hassanien, "An improved social spider optimization algorithm based on rough sets for solving minimum number attribute reduction problem," *Neural Comput. Appl.*, vol. 30, no. 8, pp. 2441–2452, 2018.
[64] R. A. Ibrahim, M. Abd Elaziz, D. Oliva, and S. Lu, "An improved runner-root algorithm for solving feature selection problems based on rough sets and neighborhood rough sets," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 105517.
[65] M. E. A. Elaziz, A. A. Ewees, D. Oliva, P. Duan, and S. Xiong, "A hybrid method of sine cosine algorithm and differential evolution for feature selection," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, Nov. 2017, pp. 145–155.