Mathematical Biosciences and Engineering

MBE, 20(3): 5268–5297.
DOI: 10.3934/mbe.2023244
Received: 29 October 2022
Revised: 12 December 2022
Accepted: 22 December 2022
Published: 11 January 2023

http://www.aimspress.com/journal/MBE

Research article

Optimal feature selection using novel flamingo search algorithm for classification of COVID-19 patients from clinical text

Amir Yasseen Mahdi[1,2,*] and Siti Sophiayati Yuhaniz[1]

[1] Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia
[2] Computer sciences and mathematics college, University of Thi_Qar, Thi_Qar, 64000, Iraq

* **Correspondence:** Email: mahdi.amir@graduate.utm.my, amiryasseen@utq.edu.iq.

Abstract: Though several AI-based models have been established for COVID-19 diagnosis, the machine-based diagnostic gap is still ongoing, making further efforts to combat this epidemic imperative. So, we tried to create a new feature selection (FS) method because of the persistent need for a reliable system to choose features and to develop a model to predict the COVID-19 virus from clinical texts. This study employs a newly developed methodology inspired by the flamingo's behavior to find a near-ideal feature subset for accurate diagnosis of COVID-19 patients. The best features are selected using a two-stage. In the first stage, we implemented a term weighting technique, which that is RTF-C-IEF, to quantify the significance of the features extracted. The second stage involves using a newly developed feature selection approach called the improved binary flamingo search algorithm (IBFSA), which chooses the most important and relevant features for COVID-19 patients. The proposed multi-strategy improvement process is at the heart of this study to improve the search algorithm. The primary objective is to broaden the algorithm's capabilities by increasing diversity and support exploring the algorithm search space. Additionally, a binary mechanism was used to improve the performance of traditional FSA to make it appropriate for binary FS issues. Two datasets, totaling 3053 and 1446 cases, were used to evaluate the suggested model based on the Support Vector Machine (SVM) and other classifiers. The results showed that IBFSA has the best performance compared to numerous previous swarm algorithms. It was noted, that the number of feature subsets that were chosen was also drastically reduced by 88% and obtained the best global optimal features.

Keywords: COVID-19; natural language processing; feature selection; binary flamingo search

algorithm; clinical text classification

1. Introduction

A new coronavirus (COVID-19) emerged in Wuhan in December 2019 and quickly swept worldwide [1]. The COVID-19 epidemic was declared a Public Health Emergency of International Concern by the World Health Organization in January 2020 [2]. To counteract, control, lessen, and confine the COVID-19 virus's effects and consequences, several studies are still being done in a variety of fields. A number of models based on artificial intelligence have been developed to diagnose COVID-19 disease [3]. However, there are still a few models based on the machine to diagnosis of infectious epidemics.

This study is focused on clinical text mining related to COVID-19 and applying machine learning algorithms to categorize COVID-19 patients. Individual symptoms, demographic information, diagnosis, laboratory test results, chest x-ray reports, treatments, etc., can all be found in clinical texts, which are narrative texts providing a great deal of information regarding afflicted patients. However, the data in clinical texts are often high dimensional and include uninformative features, that significantly affect the accuracy of the classifier. As a result, the dimensionality of the data must be decreased [4]. Due to the vast amount of the clinical documents size, Feature Selection (FS) is an essential step before the classification process [5]. Their main advantages involve finding a subset of relevant features that will be useful in categorization. In addition to delivering high recognition, easing data comprehension, shortening training time, and resolving the curse of dimensionality problem [6,7]. FS is a challenging and complex problem because it necessitates striking a balance between lowering features and maintaining high classifier accuracy, so it requires an effective search strategy, especially when dealing with clinical text. Complicated issues, such as those involving feature selection, are often tackled with the help of algorithms that take inspiration from nature. In recent years, numerous novel swarm intelligence optimization algorithms have been proposed, such as the binary horse herd optimization algorithm [8], moth flame optimization [9], Binary Particle Swarm Optimization [10,11], binary grey wolf optimizer [12], binary aquila optimizer [13], artificial gorilla troop optimization [14].

For the first time, the flamingo search algorithm (FSA), for handling FS tasks in the healthcare sector, is presented in this work. FSA is an efficient new method for a novel swarm intelligence optimization inspired by the flamingo's lifestyle in the migratory and foraging behavior. Figure 1 depicts flamingo communities and individuals in their natural habitat. Flamingos are known for their foraging and migratory behaviors. To the best of our knowledge, it has not been used in feature selection issues; consequently, in this research, the proposed IBFSA has been developed to minimize the number of features chosen from the clinical text related to COVID-19 while maximizing classification accuracy. The proposed method is a wrapper-based approach. Hence a learning algorithm should be part of the evaluation process. In this investigation, SVMs are used [15,16]. The most important contributions of this study are:

- Development of a swarm algorithm called IBFSA to deal with feature selection process by an improved binary version of FSA is introduced.
- A novel modified Initialization approach has been proposed to enhance diversity and convergence during the research process.

- Levy flight has been incorporated into FSA to increase the diversity of solutions and offer a high level of randomization.
- The local search algorithm is incorporated before and after each iteration of FSA to prevent becoming stuck in local optima.
- Combining term weighting schema (RTF-C-IEF) with IBFSA.
- Propose a new clinical text categorizer by combining IBFSA and SVM.
- Use Two datasets to compare the state-of-the-art techniques with our proposed method.

(a) (b)

Figure 1. Flamingo population (a) flamingo group; (b) flamingo individuals.

The remaining parts of the paper are structured as follows. Section 2 the related works of clinical of COVID-19 and the FS procedure. Section 3: An overview about the FSA. The proposed methodology is outlined in Section 4. The experimental and findings are presented and discussed in Section 5. Finally, Section 6, concludes the paper.

2. Related works

Comparatively few attempts have been made to create intelligent classifiers, including feature selection, for the clinical text categorization of COVID-19 patients than for other topics. To correctly identify COVID-19 patients, the authors of this paper [17] employed Binary Particle Swarm Optimization (BPSO) as a wrapper approach for critical feature selection. According to experiments, it not only beats other methods but also introduces the highest possible degree of accuracy with the lowest possible time overhead. The COVID-19 dataset in [18] to disease diagnosis based on Grasshopper Optimization Algorithm (GOA), was used. The experimental findings demonstrate that

the suggested method provides high classification accuracy. In this paper [19], presents an intelligent strategy for predicting SARS-CoV2 (COVID-19) using genetic feature selection techniques. The proposed model appears to have substantially lower prediction errors than conventional techniques. In this paper [20], the authors propose using a hybrid strategy based on the BOA algorithm and particle swarm optimization (PSO). The suggested methodology has been tested using the COVID-19 dataset. The experimental results show that the proposed model BOAPSO outperforms the PSO, BOA and GWO in terms of improving performance precision and reducing the number of chosen features by 91.07, 87.2, 87.8 and 87.3%, respectively. This paper [14] aims to introduce a unique discrete artificial gorilla troop optimization (DAGTO) approach for dealing with FS challenges in the healthcare sector. After completing a case study on COVID-19 samples and ten medical data sets were using to demonstrate the method's influence in practice. Evidence from statistically shows that it performs the best. In this study [13], the single Aquila optimizer (AO) is suggested as a search technique to find the optimal feature subset. The COVID-19 real-world dataset is used to evaluate the proposed method. Results showed that AO is superior to competing algorithms in terms of accuracy attained with the fewest features. The novel Caledonian crow learning algorithm is used in this study [21] to propose a strategy for selecting features relevant to the COVID-19 illness. The suggested approach for detecting COVID-19 patients is more accurate than a competing method, as demonstrated by experimental findings on the COVID-19 disease dataset at a Brazilian hospital. The best feature subset may be chosen with the help of a mix of the brainstorm optimization algorithm and the firefly algorithm, as described in this article [22]. For the dataset of coronavirus-related diseases, the proposed technique was used. The experimental findings produced demonstrated superior classification accuracy compared to previous approaches. Table 1 provides a brief comparison of earlier works on the COVID-19 detection method.

In conclusion, when comparing machine learning and globally intelligent algorithms to conventional methodologies, most of the experiments on COVID-19 Classification showed good classification results. In addition, swarm intelligence algorithms have been effectively used in the feature selection problem to manage various domains, but they are not extremely applied in clinical text related to COVID-19 categorization. As a result, there is a need and substantial motivation to present a new approach, which includes a weighting scheme, an intelligent feature selection method based on IBFSA, and SVM classifier for classification of the COVID-19 patients from clinical texts.

3. Overview of standard FSA

The FSA is an evolutionary algorithm with biological inspiration that is modeled after how flamingos in nature find food. Each candidate solution to the optimization issue in this algorithm is represented by a flamingo, and each flamingo has two primary characteristics, namely, its foraging and migrating patterns. Flamingos have no idea where most of the food is in the present (the globally ideal) search region. Therefore, flamingos look for a food site with more plentiful food than the known food in the search region by sharing information with each other, updating the location of each flamingo, and affecting changes in the locations of other flamingos in the group (the optimal solution Global). Identifying the globally best solution inside a specified search area is a significant aim of the swarm intelligence algorithm, and the flamingos' behavior is a fitting metaphor for this purpose [23].

The fundamental steps of this algorithm are described below:

Step 1. The population is initialized, set as P, the maximum number of iterations is $Iter_{Max}$, and the proportion of migrating flamingos in the first part is MP_b.

Step 2. The number of foraging flamingos in the ith iteration of flamingo population renewal is $MP_r = rand[0,1] \times P \times (1 - MP_b)$. The number of migrating flamingos in the first part of this iteration is $MP_o = MP_b \times P$. The number of migratory flamingos in the second part of this iteration is $MP_t = P - MP_o - MP_r$. Individual flamingo fitness levels are calculated, and the entire flamingo population is then ranked by fitness. The flamingos with low fitness MP_b and high fitness MP_t are classified as migrants, while the others are classified as foraging flamingos.

Step 3. Migrating flamingos are modified based on Eq (2), and foraging flamingos are modified based on Eq (1).

$$\boldsymbol{x_{ij}^{t+1}} = \left(x_{ij}^{t} + \varepsilon_1 \times xb_j^t + G_2 \times \left|G_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t\right|\right)/K \tag{1}$$

In Eq (2), x_{ij}^{t+1} presents the location of the ith flamingo in the ith dimension of the population in the $(t+1)$th iteration, x_{ij}^t represents the location of the ith flamingo in the jth dimension in the t iteration of the flamingo population, namely, the location of the flamingo's feet. xb_j^t represents the jth dimension location of the flamingo with the best fitness in the population in the t iteration. $K = K(n)$ is a diffusion factor, which is a random number that follows the chi-square distribution of n degrees of freedom. It is utilized to increase the size of the foraging-group for flamingos and simulate the possibility of individual selection in nature, enhancing its the global ability to search for the best opportunity. The random numbers $G_1 = N(0,1)$ and $G_1 = N(0,1)$ have a conventional normal distribution, ε_1 and ε_2 are determined by −1 or 1 at random.

$$\boldsymbol{x_{ij}^{t+1}} = x_{ij}^t + \beta \times \left(xb_j^t - x_{ij}^t\right) \tag{2}$$

In Eq (2), x_{ij}^{t+1} and xb_j^t represents same meaning as the previous Eq (1). $\beta = N(0,1)$ is a set of random integers with the same distribution across all trials; it is employed to broaden the search area during flamingo migration and simulate the randomness of individual flamingo behaviors during the particular migration process.

Step 4. Make sure there are no flamingos that are off-bounds.

Step 5. Move to Step 6 if the allotted number of iterations has been used; otherwise, go to Step 2.

Step 6. Result in the ideal solution and optimal value.

The FSA pseudo code is displayed in Algorithm 1.

Table 1. A summary comparison of earlier works on the COVID-19 detection method.

Method	Advantages	Disadvantages
Aquila Optimizer (AO) and ML [13]	AO significantly outperforms other comparison algorithms and has been shown to be more effective in terms of predictive accuracy and reducing the number of features selected.	The COVID-19 patient data set used is small, and was not of very high dimensionality for the method to be explored effectively
AGTO and ML [14]	Efficient in reducing the number of features used with better accuracy, also this approach has been demonstrated to be successful in real-world practical applications using real-world COVID-19 datasets.	The majority of the time, AGTO takes longer to implement. In addition, the database is not very highly dimensional. However, different approaches can be used to enhance the efficiency of the algorithm by applying advanced initialization procedures.
PSO and DBNB classification [17]	The suggested method attempts to accurately identify infected patients with the least time penalty based on the more effective features elected by APSO.	Even though it is effective at diagnosing COVID-19 patients, the suggested method is only based on numerical data. Additionally, the dataset used is not insufficient to diagnose COVID-19 and is limited just to clinical laboratory data. However, analyzing CT scan reports may be helpful to confirm infection.
GOA and CNN [18]	Easy to implement and takes little time by optimizing CNN by GOA.	By utilizing a more detailed dataset with more images from all three classes, the proposed method can be further enhanced.
BOA, PSO and ML [20]	Compared to conventional classification methods, the proposed hybrid model is more effective at classifying COVID-19 patients.	The COVID-19 patient data set used is small, and was not of very high dimensionality.
CA and ANN [21]	ANN is a powerful classification technique.	The patient election has potential bias because the database is so unbalanced that only the number of infected people in it is 10% of the total number.
BSO, FA and ML [22]	Compared to conventional classification methods, the proposed hybrid model is more effective at classifying COVID-19 patients.	The COVID-19 dataset contains limited data limited only to symptoms and its small size, plus a lot of missing data. So, it needs other methods of pre-processing.

Algorithm 1: Standard Flamingo Search Algorithm

Input:

$M - maximum\ number\ of\ iterations$

$N - total\ number\ of\ flamingo$

$MP_b - number\ of\ migrating\ flamingo$

Output:

$X_{best} - Global\ optimal\ position$

$f_{best} - Fitness\ of\ global\ optimal\ position$

Start
Initialize a swarm of N $flamingo$s and its relevant parameters;
$t \leftarrow 1$;
While $t < M$ **do**
  R← $rand(1)$; /* randomly assign the alarm value in [0,1] */;
  $MP_r \leftarrow R \times P \times (1 - MP_b)$
  $MP_o \leftarrow MP_b$
  $MP_t \leftarrow P - MP_b - MP_r$
  for flamingo migration $i = 1,2, \dots, MP_b$ **do**
    for j= $1,2, \dots, n$ **do /*n is dim size**
      Update the flamingo's position using Eq (2);
    end for
  end for
  for flamingo foraging $i = 1 + MP_o\ to\ MP_o + MP_r$ **do**
    for j= $1,2, \dots, n$ **do**
      Update the flamingo's position using Eq (1);
    end for
  end for
  for $i = 1 + MP_o + MP_r\ to\ P$ **do**
    for j= $1,2, \dots, n$ **do**
      Update the flamingo's position using Eq (2);
    end for
  end for
  Find the current new location X_i^{t+1}
  $x_{ij}^t = call\ function\ the\ boundar\ detection(lb, ub)$
  Re-rank the whole swarm in ascending order based on the fitness values f(x)
  Find the current global optimum position X_{best}^{t+1}; /* First individual in the ranking*/
  $f_g \leftarrow f(X_{best})$
  $t \leftarrow t + 1$
end while
  Return X_{best}, f_g /*X_{best} is top optimal of a solution got by the algorithm */

4. Proposed methods

In order to predict a COVID-19 diagnosis from clinical texts, our strategy described in this work includes six processing stages, namely collection and describe the dataset, text pre-processing, extract features, features selection, use of machine learning methods, and performance evaluation. The suggested model's block diagram is shown in Figure 2.

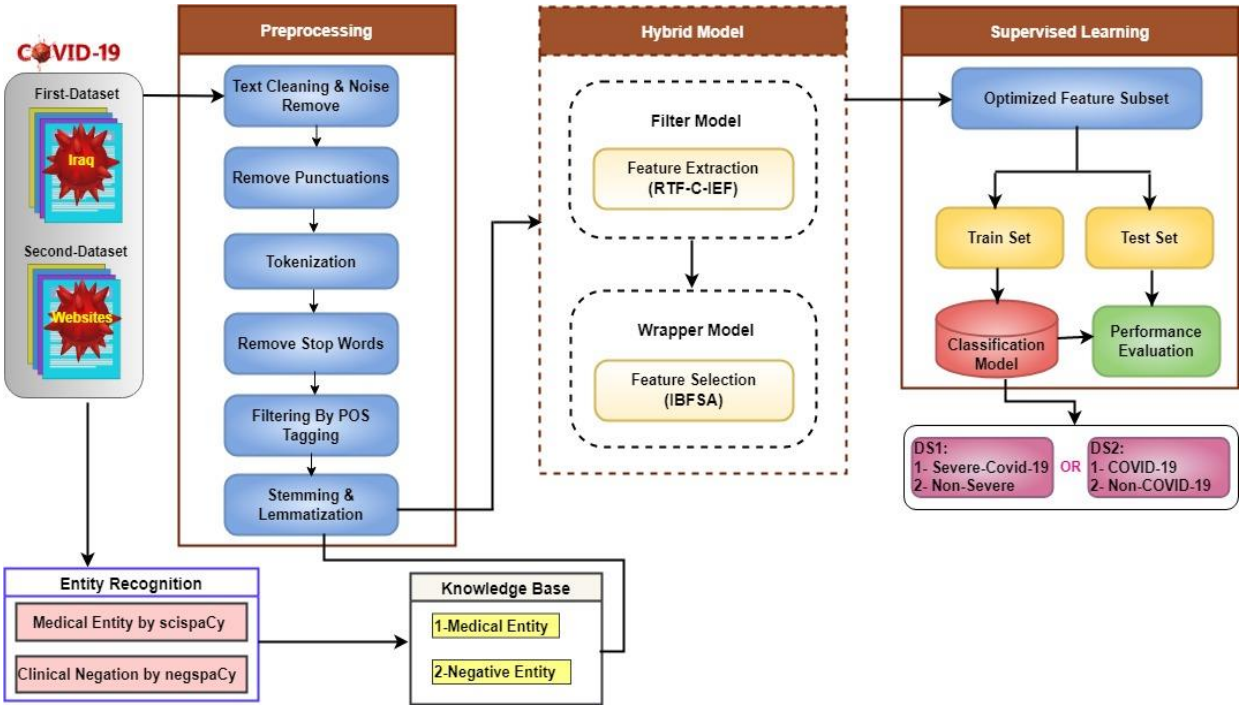

Figure 2. Diagram of the workflow of the study.

4.1. Dataset description

Two sets of clinical data related to Coronavirus (COVID-19) were collected to validate the effectiveness of the suggested method. First Dataset (DS1) was collected from several hospitals in Iraq of patients with SARS-CoV2. In contrast, other clinical text reports were collected to form the second data set (DS2) from various sources, including includes GitHub (https://github.com/Akibkhanday/Meta-data-of-Coronavirus.), the Italian Society of Medical and Interventional Radiology (SIRM) (https://www.sirm.org/category/senza-categoria/covid-19/), in addition to other cases reports, that were collected from medical publications related to COVID-19 on some websites such as Hindawi (https://www.hindawi.com/), Infection and Chemotherapy (https://www.jiac-j.com/), NIH (https://www.ncbi.nlm.nih.gov/pmc/), and ScienceDirect (https://www.sciencedirect.com/science/article/pii/S1477893921002106).

Both datasets contain "demographic" information, such as age, sex, and comorbidities, in addition to other needed diagnostics information and related tests, including symptoms, vital signs, lab results, values from routine blood tests, and chest CT imaging results, disposition, admission to an ICU, and survival to hospital discharge. The two datasets consist of 3053 and 1446 patients, respectively. Table 2 summarizes the used datasets comprising varying samples and attributes.

Table 2. Details of datasets.

No	Type	No. of records	Label	Rate of Occurrences
DS1	Clinical Text	3053	Severe	55%
			Non-Severe	45%
DS2	Clinical Text	1446	COVID-19 Positive	62%
			COVID-19 Negative	38%

4.2. Text preprocessing

Clinical texts present a difficult challenge to extract the hidden features from, since they are always presented in an unstructured format. Thus, to train a classifier, data must be presented in a readable manner and undergo pre-processing. Since some symbols and words may not be beneficial for categorization, the pre-processing method aims to improve the data's quality and clean it up. Several pre-processing steps were used to convert unstructured clinical texts into a word vector. It includes removing punctuation, and numbers, stopping words and other characters, converting letters, short-word removal, tokenization, parts-of-speech tagging, stemming, and lemmatization.

4.3. Feature extraction

In order to complete NLP tasks, it is crucial to identify an effective text representation system [24]. From the pre-processed clinical texts, different features are extracted. The feature engineering described here relies on the use of two steps. SpaCy and ScispaCy were employed in the first step to extract medical entities from clinical text. Symptoms with more than one word were then converted into a single expression (e.g., "shortness of breath") in some reports. ScispaCy provides a robust rule-matching engine and Fast Models for Biomedical Natural Language Processing [25].

In the second stage, the RTF-C-IEF weighting method [26] is used to transform the extracted concepts, which are features, into probability values to be ready for the feature selection model. This procedure drastically decreases the number of features while preserving the informative features. RTF-C-IEF is a statistical weighting method to retrieve a term's significance within a document as the first stage of feature selection strategy for text mining. It was used for feature extraction instead of Bag of Word (BoW) and TF-IDF classical since RTF-C-IEF provides more accurate results [26].

A higher RTF-C-IEF feature score indicates more significance for that feature within the text's clinical context. The RTF-C-IEF formula is written as follows:

$$\boldsymbol{RTF-C-IEF} = \left(tf_{ij}\right)^{r_{tf}} \times \left(1 + \frac{t_x}{N}\right) \times e^{-\frac{dt\left(t_j\right)}{N}} \tag{3}$$

Where tf_{ij} is the term frequency, t_x represents the frequency count of the word x in the core corpus, N is the total of dataset, and $dt\left(t_j\right)$ corresponds to the frequency of documents that term t_j appears in the collection.

4.4. Improvements embedded into the standard FSA based feature selection

Prior to performing the classification, feature selection is a crucial step to choosing the

important features, eliminating the irrelevant ones, minimizing the feature dimensions, and shortening the computing time required to complete the classification [10,27,28]. To realize that, FSA [29] is implemented. FSA is a new algorithm that simulates the behavior of flamingos searching for the best possible solution within a given search region (where food is most plentiful). Since FS is a binary issue, the native optimizer needs to be tweaked so that FSA may optimize in a high-dimensional binary search space, thereby improving the algorithm's efficiency. Many significant steps in updating the FSA algorithm are detailed in this study. Introducing a new operator into the algorithm's structure is the most common method for enhancing FSA exploration as well as correcting the typical roaming behavior of swarm members. In the first step, transfer functions from S-shaped families are used to convert the FSA to binary. Secondly, A novel initialization modification (MIA) approach was incorporated into the standard FSA algorithm to obtain high-quality individuals in beginning and thus increase the likelihood of discovering the best solution, which may increase the optimization's performance. In the third stage, the Levy flight operator is added to each flamingo to boost its variability and the optimizer's capacity to probe further into underexplored portions of the search space. Finally, enhancing the exploitation by Local Search Algorithm (LSA). These promising improvements are discussed in this sub-section. The architecture of the suggested feature selection approach is depicted in Figure 4, and the pseudocode of IBFSA is presented in Algorithm 4.

4.4.1. Transformation function

Modeling the FS problem as a binary one, which can only take values 0 or 1 in the feature-subset selection issue. Thereby, FSA cannot be utilized to directly resolve a feature selection problem because the final solution it produces using Eqs (1) and (2) is made up of continuous values (real number domain). As a result, a transfer function (TF) must be used to convert the values from continuous to binary (0 or 1). TF specifies the rate at which the values of the decision variables change from 1 to 0 and back. That is, when choosing a TF to convert the continuous values into binary (0,1), the range of values the TF produces should fall within the range [0,1]. The S-shaped family of logistic transformation functions is perfect for mapping processes since it produces output in the [0,1] range. The purpose of this discovery is to identify features that have been omitted or elected. In this case, the flamingo stands for features set, and its binary values indicate whether or not that feature was chosen for inclusion in the final model, where 1 represents a selected feature and 0 means discard. An individual's value range is now mapped to [0,1] by the following function [10]:

$$\boldsymbol{TF}\left(\boldsymbol{x_i^d(t)}\right) = \frac{1}{1+e^{-2x_i^d(t)}} \tag{4}$$

Where x_i^d denotes the i^{th} flamingo location in the d^{th} dimension at the t^{th} iteration, x_i is computed by Eqs (1) and (2). In Eq (4), the output of the S-shaped function is still displayed continuously as illustrated in Figure 3. Thus, to obtain the binary value the i^{th} position is modified as follows:

$$\boldsymbol{x_i^d(t+1)} = \begin{cases} 0 & rand < TF\left(x_i^d(t)\right) \\ 1 & rand \geq TF\left(x_i^d(t)\right) \end{cases} \tag{5}$$

Where $x_i^d(t+1)$ represents the ith element in the X solution at dimension d in iteration $t+1$, and $rand \in [0,1]$.

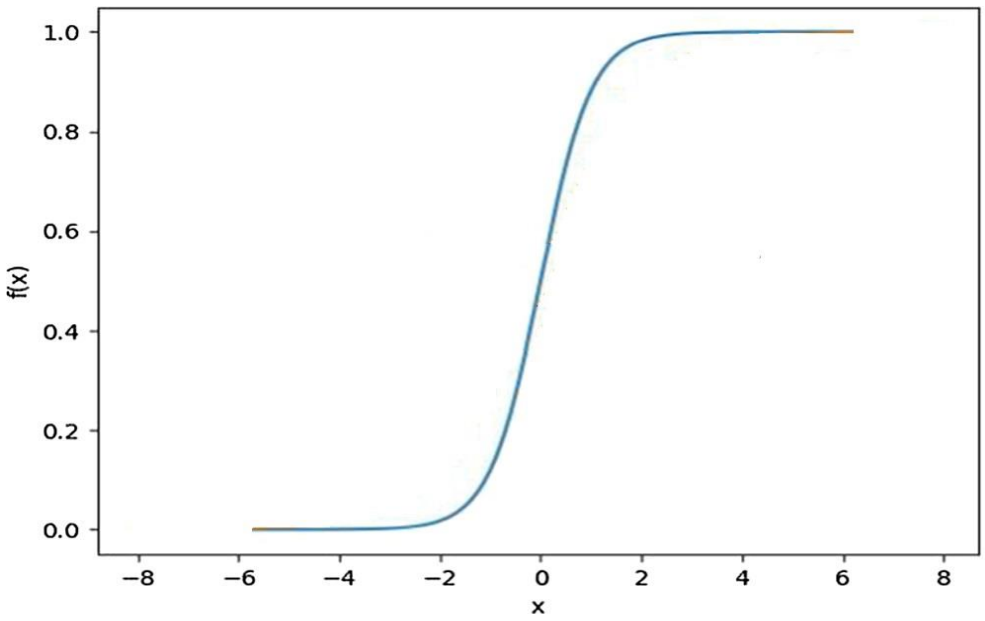

Figure 3. S-shaped function used in FSA algorithm.

4.4.2. Levy flight strategy

Figure 4 depicts Levy flight, a mathematical representation of a random motion that follows a heavy-tailed probability distribution [30]. Levy flight was recently introduced as a solution to optimization problems. It has since been incorporated into the design of many optimization algorithms to improve their performance in areas including speed of convergence, preventing premature convergence, leaping from local minima, and striking a balance between exploration and exploitation [8,9,30]. This research aims to improve the FS process used in the COVID-19 diagnosis from clinical texts by proposing for the first time that Levy flight be included in the FSA structure to enhance the performance of the FSA optimizer. An equation that represents the flamingo location update based on Levy's flying improvement is Eq (6). So, in order to increase the variety of search spaces, it has been planned that each upgraded flamingo would employ Levy flight once, resulting in a higher level of exploration.

$$\boldsymbol{x_{ij}^{t+1}} = \left(x_{ij}^t + \varepsilon_1 \times xb_j^t + levy(\beta) \oplus \left|G_1 \times xb_j^t + \varepsilon_2 \times x_{ij}^t\right|\right)/K \tag{6}$$

$$\boldsymbol{Levy(\beta)} \sim \boldsymbol{\mu} = t^{-1-\beta} \qquad 0 \le \beta \le 2 \tag{7}$$

$$\boldsymbol{levy(\beta)} \sim \frac{\phi \times \mu}{\left|V^{1/\beta}\right|} \tag{8}$$

$$\boldsymbol{\phi} = \left[\frac{\Gamma(1+\beta)\times\sin(\pi\times\beta/2)}{\Gamma\left(\left(\frac{1+\beta}{2}\right)\times\beta\times 2^{\frac{\beta-1}{2}}\right)}\right]^{\frac{1}{\beta}} \tag{9}$$

Where X_i^t indicates the i^{th} flamingo at iteration t, rand indicates a random number in the range [0, 1], $\oplus$ represents the dot product, and α represents the step control parameter. Levy flight, as previously mentioned, is a random walk where the leap size supports a Levy distribution as given in Eq (7). Using Eq (8), Levy is computed as random numbers; μ and ν are common random distributions. Eq (9) shows how to calculate φ, where Γ represents a typical Gamma function, and β = 1.5, mentioned in [31].

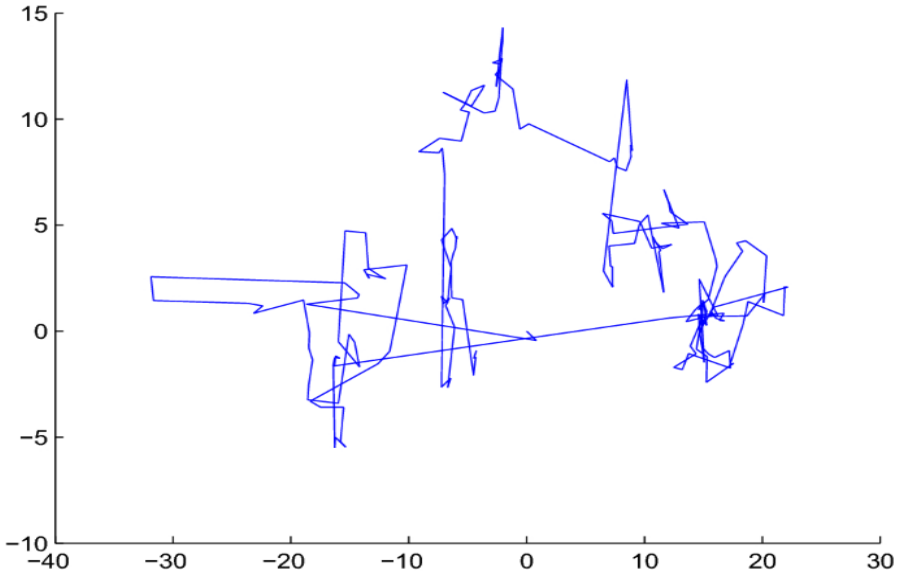

Figure 4. Simulated Levy flight.

4.4.3. Modified initialization approach (MIA)

Evolutionary algorithms rely heavily on the variety and convergence of their populations, and population initialization is a crucial aspect of this. This step's purpose is to offer an initial guess at potential solutions. These initially hypothesized solutions will then be iteratively enhanced throughout the optimization process until a stopping requirement is fulfilled. In most cases, having a high-quality initial population can help an algorithm converge more quickly and find the optimal solution. On the other hand, it is possible that an algorithm will not be able to locate the optimal solution if it has based on poor guesses [32,33]. In recent years, researches has shown that proper initialization approaches can improve the likelihood of locating global optimum solutions and decrease the variance of the final search outcomes [34]. In this paper, the performance of FSA is expanded to make it appropriate for the optimization problem by introducing a new initialization algorithm named MIA. Its basic idea is to create a population based on the initial population in a sporting way without any complex equation or making much change in the original FSA algorithm and its structure. Next, the better individuals will be selected out of the initial population, resulting in

the creation of a new initial population made up of outstanding individuals. Thus, the MIA managed to manage part of this algorithm and correctly cover the possible space. Additionally, the suggested initialization technique significantly impacts solution quality, finds the optimal solution with high precision, and has helped boost the likelihood of starting with a global optimum. The whole pseudo code of MIA is displayed as Algorithm2.

Algorithm 2: The proposed MIA algorithm

$\boldsymbol{X_{ij}}$ = *position of flamingos;* /* Randomly generate the positions of *N flamingo;*
$\boldsymbol{X_{bin}}$ = After Convert to binary_map ($\boldsymbol{X_{ij}}$);
$\boldsymbol{Fit_{old}}$ = Find all Fitness to *Population size(flamingos);*
D_{max} = *maximum of number of local iterations;*
M_{max} = *maximum of number of local iterations;*
***N** (Population size).*

for $d = 1\ To$ $\boldsymbol{D_{max}}$ **do**
  Find $\boldsymbol{X_{bin-best}}$ /* (Global optimal position)
  for i= $1\ To$ **N**
    $X_{new} \leftarrow (\boldsymbol{X_{bin-best}} + \boldsymbol{X_{ij}})$ * rand /* **Generate** a new position;
    $\boldsymbol{X_{bin-best}} \leftarrow$ binary_map (X_{new})
    Calculate the fitness function values of each Flamingo $\mathbf{F_i}$
      if $\boldsymbol{F_i} < \boldsymbol{Fit_{old}}$ then
        $\mathrm{Fit_{old}} \leftarrow \mathrm{F_i}$
        $\mathrm{X_{ij}} \leftarrow \mathrm{X_{new}}$
        $\mathrm{X_{bin}} \leftarrow \mathrm{X_{bin-best}}$
      end if
      for m= $1\ To$ $\boldsymbol{M_{max}}$ **do**
        $randfeat \leftarrow rand$ / * randomly selected features, $\in \{0,1\}$.
        Calculate the fitness function values of each Flamingo $\boldsymbol{F_m}$ $(randfeat)$
          if $\boldsymbol{F_m} < \boldsymbol{Fit_{old}}$ then
            $\mathrm{Fit_{old}} \leftarrow \mathrm{F_m}$
            $\mathrm{X_{bin}} \leftarrow$ randfeat
          end if
      end for
  end for
end for
Return $\mathrm{X_{bin}}$, X_{ij}, Fit_{old}

4.4.4. Improving based on local search algorithm (LSA)

The LSA algorithm was created and presented in Algorithm3 by [35] . In the original FSA, in each iteration of the migratory flamingo MP_b, LSA is called to enhance the local location obtained by the Eq (3). After the migratory flamingos have moved to their best position, LSA is again called to improve finding the best solution X_{ij}^{t+1} currently obtained by still removing any more potentially

pointless features. At first, LSA stores, in a variable $Temp$, the value of X_{best}^{t+1} produced at the end of each IBFSA iteration. To improve $Temp$, LSA runs iteratively LT times. At each iteration L_t of LSA, four features' $rand-feat$ are randomly selected from $Temp$. Every variable in the $rand-feat$ is reversed by LSA. Then, the value of fitness $f(Temp)$ of the new solution (the new *Temp*) is evaluated; if it is best than (X_{best}^{t+1}) , then X_{best}^{t+1} is set to $Temp$; otherwise, X_{best}^{t+1} and f_g are kept unaltered.

Algorithm 3: The proposed LSA algorithm

$\boldsymbol{LT}-$ *maximum of number of local iterations;*
X_{best}^{t+1} /* the best position so far at the end of IBFSA current iteration $t+1$;
$Temp \leftarrow X_{best}^{t+1}$
$Lt \leftarrow 0$;
While $Lt < LT$ **do**
  $rand-feat \leftarrow four\ feature\ randomly\ selected\ from\ Temp$;
  for feature $\alpha \in rand-feat$ **do** /* $\alpha \in \{0,1\}$
    $\alpha \leftarrow \neg\alpha$;
  end for
  $f(Temp) \leftarrow calcualte\ the\ fitness\ value\ of\ Temp$;
  if $f(Temp) < f\left(X_{best}^{t+1}\right)$ **then**
    $X_{best}^{t+1} \leftarrow Temp$
    $\boldsymbol{X_{best}} \leftarrow X_{best}^{t+1}$
    $f_g \leftarrow f(\boldsymbol{X_{best}})$
  end if
  $Lt \leftarrow Lt + 1$
end while
Return $\boldsymbol{X_{best}}$, f_g

In addition, in order not to lose the distinctive sites that the flamingo passes through in its journey during the search for the optimal global solution, we added a parameter to help it maintain its sites that have the best fitness value appropriate that it has currently reached, and this prevents the flamingo from moving away from the optimal position and moving to a worse position.

4.5.Binary FSA for FS problem

After the flamingo is converted into a binary vector with the same number of rows and columns of the dataset in TF. The fitness function of the IBFSA is used to quantify each flamingo's level of fitness by combining two seemingly opposing goals. These goals are the number of chosen features and the accuracy. The FS problem seeks to maximize classification accuracy (minimize error rate) with a minimum of specified features. Then, the model performance was optimized with the SVM technique, and the optimal set of features for detecting COVID-19 was determined by identifying the

best flamingo. IBFSA uses the following fitness function to evaluate the solutions and achieve an equilibrium between the two main goals:

$$\boldsymbol{Fit_{FS}} = \alpha \times E + \beta \times \frac{d}{D} \tag{10}$$

Where E is the classifier's error rate, d is the number of features used to make a decision, and D is the total number of features. In addition, the values of α and β are the weights employed to strike a balance between these two goals.

Algorithm 4: The proposed IBFSA based on MIA, TF, Levy flight and RSA

Input:

$M - maximum\ number\ of\ iterations$

$N - total\ number\ of\ flamingo$

$MP_b - number\ of\ migrating\ flamingo$

Output:

$X_{best} - Global\ optimal\ position$

$f_{best} - Fitness\ of\ global\ optimal\ position$

Start
Initialize a swarm of N $flamingo$s and its relevant parameters;
Apply **MIA** to X_{ij} using Algorithm (2);
$t \leftarrow 1$;
While $t < M$ **do**
    R← $rand(1)$; /* randomly assign the alarm value in [0,1] */;
    $MP_r \leftarrow R \times P \times (1 - MP_b)$
    $MP_o \leftarrow MP_b$
    $MP_t \leftarrow P - MP_b - MP_r$
    Re-rank the whole swarm in ascending order based on the fitness values f(x) (see Eq (10))
    for flamingo migration $i = 1,2, \dots, MP_b$ **do**
        for j= $1,2, \dots, n$ **do /* n is dim size**
            Update the flamingo's position using Eq (2);
            $X_{ij}^{t} \leftarrow call\ function\ the\ boundar\ detection(lb, ub)$
            Apply binary conversion, using Eq (5);
            Calculate the fitness degree of all flamingo using Eq (10);
            Find the current new location X_i^{t+1};
            Apply **LSA** to $(X_{ij})_{\text{bin}}$ using Algorithm 3;
            Find the current global optimum position X_{best};
            Apply **LSA** to X_{best} using Algorithm 3;
        end for
    end for
    for flamingo foraging $i = 1 + MP_o\ to\ MP_o + MP_r$ **do**
        for j= $1,2, \dots, n$ **do**
            Update the flamingo's position using Eq (1);
            Levy flight is used to update the position of each flamingo;

$X_{ij}^{t} \leftarrow call\ function\ the\ boundar\ detection(lb, ub)$;
Apply binary conversion, using Eq (5);
Calculate the fitness degree of all flamingo using Eq (10);
Find the current new location X_i^{t+1};
end for
end for
for $i = 1 + MP_o + MP_r\ to\ P$ **do**
for j= $1,2,\ldots,n$ **do**
Update the flamingo's position using Eq (2);
$X_{ij}^{t} \leftarrow call\ function\ the\ boundar\ detection(lb, ub)$
Apply binary conversion, using Eq (5);
Calculate the fitness degree of all flamingo using Eq (10);
Find the current new location X_i^{t+1};
end for
end for
for $i = 1,2,\ldots,P$ **do**
Find the current global optimum position X_{best}^{t+1};
end for
$X_{best} \leftarrow X_{best}^{t+1}$;
$f_g \leftarrow f(X_{best})$;
$t \leftarrow t + 1$
end while
end
Return X_{best}, f_g /*X_{best} is the best solution obtained by the algorithm*/

4.6. Classifier and evaluation

The proposed method is a wrapper-based approach. Hence a learning algorithm should be part of the assessment process. In this research, SVMs are used as classifiers in the fitness evaluation process [36,37,38] because they are so efficient, mainly when dealing with data sets that only have two classes. In addition, the other classifiers are utilized in all other cases. Each dataset was divided at random into 20% for testing and 80% for training. Multiple metrics, including precision, sensitivity, F-measure, Macro-F1, and Macro-Recall, are used to assess the results of our tests and verify the efficacy of the suggested method. Are defined as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad , \quad \text{Recall} = \frac{TP}{TP+FN} \tag{11}$$

$$\text{F1\_score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+ Recall}} \tag{12}$$

$$MacroF = \frac{1}{T}\sum_{j=1}^{T} F_j \tag{13}$$

$$MacroR = \frac{1}{T}\sum_{j=1}^{T} R_j \tag{14}$$

Where T denotes the total number of categorized classes and, F_j, R_j are F, R values in the j^{th} category of class. In order to increase the statistically significant of the empirical results, we independently test each optimization technique 20 times across all datasets. For each assessment, the following metrics are calculated and used: average classification accuracy, features election ratio, average fitness, and standard deviation (STD) and adopted as follows:

$$\boldsymbol{\mu_{feat}} = \frac{1}{20}\sum_{k=1}^{20}\frac{d_*^k}{D} \tag{15}$$

$$\boldsymbol{\mu_{fit}} = \frac{1}{20}\sum_{k=1}^{20} f_*^k \tag{16}$$

$$\mathbf{SD} = \sqrt{\frac{1}{19}}\sum_{k=1}^{20}\left(Y_*^k - \mu_Y\right) \tag{17}$$

5. Results and analysis

This section offers a comprehensive empirical examination of the IBFSA optimization algorithm's behavior based on several improvements. Two datasets of patient medical records from COVID-19 are utilized for experiments. Table 1 details the specifics of these data collections.

5.1. Parameter tuning

It is well-known that it is challenging for a metaheuristics method to achieve optimal performance across all possible optimization situations, especially when employing the same parameter settings. Therefore, to obtain optimal performance, it is preferable to fine-tune the critical parameters for each optimization issue independently. Parameters must be established when the IBFSA has been defined, and its procedure explained (the number of flamingos, the number of iterations, and the number of runs). The iterations provide the flamingos the chance to achieve the best intensity during one generation. When the number of iterations is repeated multiple times, the runs get their best intensity. Although the runs take more time, they ensure that the solution produced is optimal. Keep in mind that only a subset (80%) of the COVID-19 datasets is used in the experiments for parameter setup. At the same time, the remaining data is held for assessment and validation at the end (testing data). To prevent random bias, each combination is separately run 20 times, and the average results are then shown. In addition, the state-of-the-art wrapper approaches, such as BPSO, BGWO, BWOA, BMFO and BFFA, were compared to the suggested method. All algorithms have been built with the same computer platform and settings for all algorithm parameters to ensure that comparisons are fairness. Table 3 displays how finely tuned the parameters got.

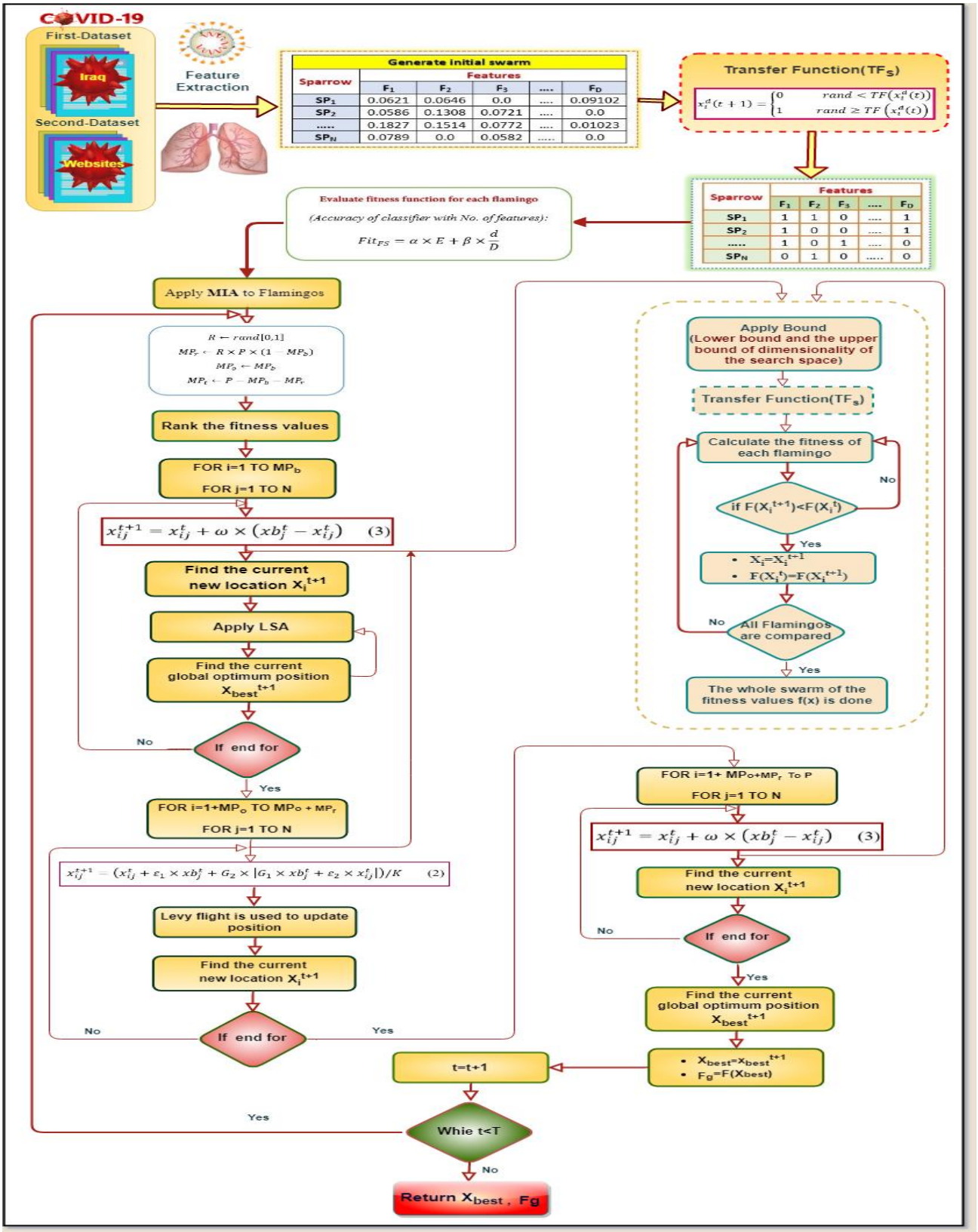

Figure 5. The sequential steps of IBFSA-FS method.

Table 3. Parameter settings for IBFSA.

IBFSA Parameters	Description	Setting
N	Run Time	20
Pop. Size(N)	Number of flamingo search agents	50
$Iter_{max}$	Maximum number of iterations	500
Dim	Dimension	Number of features
β	Significance of the feature subset	0.01
α	Importance of classification accuracy	0.99
$\boldsymbol{MP_b}$	Proportion of migrating flamingo	0.1

5.2. Experiment

Here, we show the results we got from applying our method to test datasets associated with Covid-19, measuring how well our system did at classifying the data. In two stages, experiments are conducted. In stage one, the term weighting schema's impact is investigated on datasets to categories Covid-19 patients as we look for the best performance by including it in the suggested strategy. In the second stage, the proposed IBFSA is compared to numerous alternative wrapper FS methods to demonstrate the proposed method's efficacy. The IBFSA result, which consists of clinical texts with decreased feature sizes, is used as input for classifiers to categorize the patients into the appropriate classes. Take note, the phase of feature selection was separated from the phase of categorization. SVM with a linear kernel function as baseline classifier, Random Forest, the logistic recursion Nave Bayes classifier, and the multi-layer perceptron are all used to assess the quality of the feature subsets. These experiments are based on two key metrics: 1) The total number of features chosen; 2) Secondly, the accuracy of the classification. Measures such as best fitness value, worst fitness value, mean fitness value, STD for the average fitness values, the average number of the elected features, average accuracy score, and maximum accuracy value obtained are used to evaluate IBFSA performance on the FS issue in this section. For ease of understanding, the optimal results of a particular method are presented in bold.

Table 4. Number of the extracted features from pre-processing.

Dataset	Number of features
DS1 of Covid-19	377
DS2 of Covid-19	2367

Table 5. Fitness values from various algorithms on DS1.

Algorithm	Best	Worst	SD	Mean
PSO	11.9508	13.3517	3.6424	12.9468
WOA	13.1452	14.6777	3.7754	13.7407
MFO	12.8370	13.7504	2.1992	13.2715
GWO	15.1563	16.8318	4.8170	16.1638
FFA	13.8441	14.8428	2.7810	14.3461
IBFSA	13.2032	18.6477	13.1204	15.2640

Table 6. Fitness values from various algorithms on DS2.

Algorithm	Best	Worst	SD	Mean
PSO	4.6866	5.4455	2.067	5.0539
WOA	4.8834	5.9688	2.5784	5.6351
MFO	4.7724	5.5376	2.6126	5.3095
GWO	6.8914	9.0156	5.8036	8.0924
FFA	4.9955	6.1279	3.5989	5.7708
IBFSA	2.3806	5.3688	8.9300	3.9802

Table 7. Number of selected features from various algorithms on DS1.

Algorithm	Best	Worst	SD	Selection Ratio	Removal Ratio
PSO	267	302	8.5006	73.5941	26.4058
WOA	181	324	29.0923	79.1909	20.809
MFO	270	304	10.8204	75.557	24.4429
GWO	175	208	8.8317	50.1326	49.8673
FFA	197	225	8.5230	56.3129	43.6870
IBFSA	54	86	7.6461	17.9310	82.0689

Table 8. Number of selected features from various algorithms on DS2.

Algorithm	Best	Worst	SD	Selection Ratio	Removal Ratio
PSO	1681	1773	2.5576	72.858	27.1419
WOA	1156	1951	28.1438	72.3595	27.6404
MFO	1669	1830	3.8661	74.4592	25.5407
GWO	1128	1245	2.7217	49.8183	50.1816
FFA	1299	1377	1.9534	56.2251	43.7748
IBFSA	225	312	2.2832	11.2568	88.7431

Table 4 displays the total number of features extracted during pre-processing before the feature selection procedure. Tables 7 and 8 display the total number of features chosen from the datasets generated using various techniques. The tables show that, on average, the number of features is picked by using IBFSA better than any other technique tested (for both DS1 and DS2) from 20 iterations. Keep in mind that the accuracy and the number of selected features is tradeoffs. Thus, it may be challenging to get the best results in both of these objectives for any dataset. In light of this, we can conclude that the proposed IBFSA outperforms other algorithms in terms of feature selections in the chosen datasets, as shown in Figures 6 and 7.

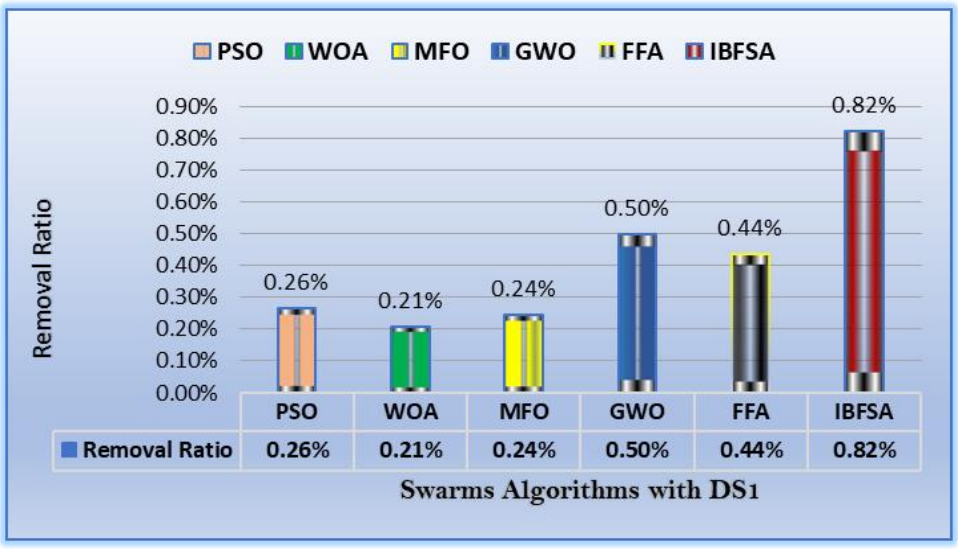

Figure 6. Average features removal ratio from DS1.

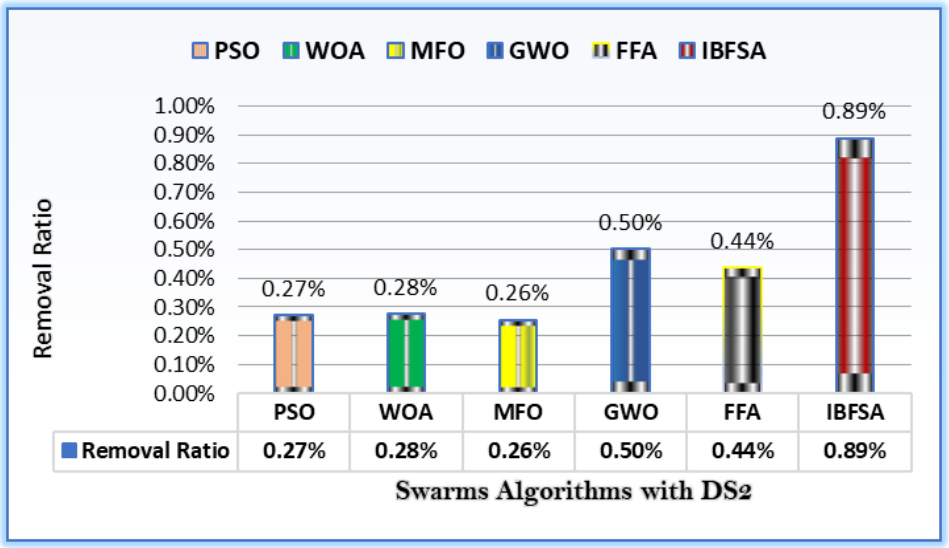

Figure 7. Average features removal ratio from DS2.

The boxplots for both datasets are seen in Figures 8 and 9 to measure the number of features selected and algorithms performance. It should be noted that the boxplots reflect outcomes of classification and the number of FS, and are displayed after each method has been executed 20 times. These figures allow us to visually see the minimum, median, and maximum values of the data. As shown in these figures, IBFSA has higher boxplots than the other approaches in both datasets.

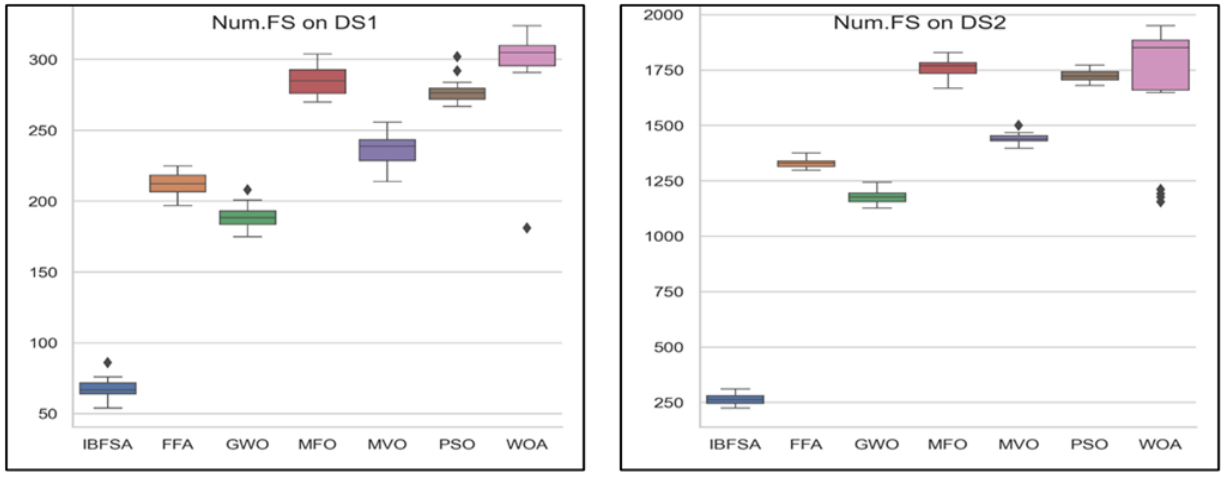

Figure 8. Boxplots of IBFSA compared with other algorithms in number of FS for both datasets.

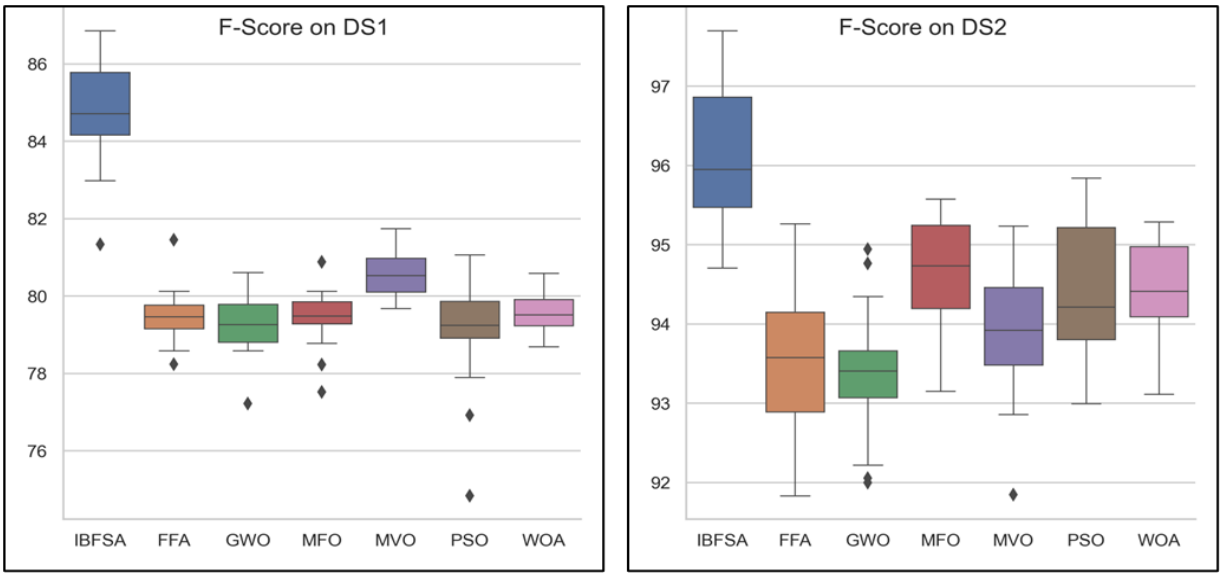

Figure 9. Boxplots of IBFSA compared with other performance of algorithms by F-score of SVM classifier for both datasets.

Table 9. Comparison results of Classification performance obtained by LR algorithm with DS1.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	79.5247	76.5082	78.9809	75.9353	85.3741	82.4489	81.5780	79.0423
WOA	78.7934	77.0658	77.9874	76.0964	85.034	83.6054	81.0450	79.6699
MFO	77.3309	76.4533	76.3975	75.4018	84.6939	83.4183	79.8700	79.2028
GWO	77.6965	75.6307	77.3885	73.7182	88.7755	85.0850	80.5030	78.9576
FFA	79.7075	76.1791	79.4212	74.6162	87.4150	84.4387	81.6520	79.2132
IBFSA	83.6996	80.1190	87.3494	80.8904	88.9262	83.6409	85.3377	81.8920

Table 10. Comparison results of Classification performance obtained by RF algorithm with DS1.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	78.9762	77.1755	77.2871	75.3667	85.3741	83.1632	80.3226	79.0642
WOA	79.5247	77.989	77.7429	75.7595	87.0748	83.8775	80.9135	79.6005
MFO	79.159	77.5502	76.8519	75.0723	86.3946	84.3027	80.5825	79.4137
GWO	77.5137	76.1152	75.3943	73.6087	89.1156	83.4524	80.4992	78.1753
FFA	79.8903	77.0292	79.0323	74.6154	86.7347	83.7925	81.1258	78.9211
IBFSA	80.9524	78.7912	79.6774	76.0975	94.2953	89.2953	84.2579	82.1315

Table 11. Comparison results of Classification performance obtained by MLP algorithm with DS1.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	79.8903	75.6307	80.0654	76.9761	88.7755	78.1292	81.6667	77.3871
WOA	79.159	76.8098	79.4118	75.5138	90.1361	84.4217	81.2102	79.6115
MFO	77.5137	76.0603	78.5156	75.1362	90.8163	83.1632	80.3709	78.8295
GWO	77.8793	75.4936	78.5441	73.9967	89.1156	84.3367	80.4314	78.6784
FFA	79.3419	75.5393	78.6184	74.8233	90.4762	82.6700	81.5057	78.3134
IBFSA	79.7075	77.5686	81.2287	76.6963	92.8571	83.8946	82.0350	80.0531

Table 12. Comparison results of Classification performance obtained by NM algorithm with DS1.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	76.782	73.7842	72.8814	70.4028	90.8163	88.5204	80.7284	78.4110
WOA	76.416	74.6618	73.5043	71.6316	89.4558	87.5680	80.0000	78.7926
MFO	76.051	74.6343	72.5762	71.4047	89.7959	88.0952	80.0000	78.8734
GWO	76.416	73.6380	71.6535	69.5449	93.5374	90.7993	80.8889	78.7455
FFA	76.234	74.4698	71.5847	70.5987	92.5170	90.0000	80.7122	79.1192
IBFSA	76.5996	74.0859	72.3118	69.6321	93.8776	92.3129	80.7808	79.3395

Table 13. Comparison results of Classification performance obtained by SVM algorithm with DS1.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	79.1590	76.5082	80.2768	75.8037	88.4354	82.9251	81.0631	79.1439
WOA	78.2450	76.9652	77.7070	76.0817	85.7143	83.3843	80.5873	79.5568
MFO	77.8793	76.5996	76.8750	75.2367	87.0748	84.1836	80.8847	79.4516
GWO	77.1481	75.5210	76.0125	72.7785	89.7959	87.1088	80.6107	79.2690
FFA	79.5247	76.0146	79.3548	73.6135	88.4354	86.4285	81.4570	79.4856
IBFSA	84.9817	82.0330	83.1288	79.0333	96.3087	91.4933	86.8590	84.7629

Tables 9 and 10 show that when comparing LR and RF performance, IBFSA performs best in terms of accuracy, precision, and F-measure index. However, there is no significant difference in average recall values between IBFSA and others. In the MLP classifier, Table 11 shows that the IBFSA has the best mean performance measured by the F-measure index. On the other hand, show Table 12 that compared to the performance of other models, the combination of Naive Bayes and IBFSA can categorize the texts with higher sensitivity. Moreover, in Table 13, we see that the SVM with IBFSA has a superior efficacy and outperforms all other algorithms regarding classifier performance, see Figure 10.

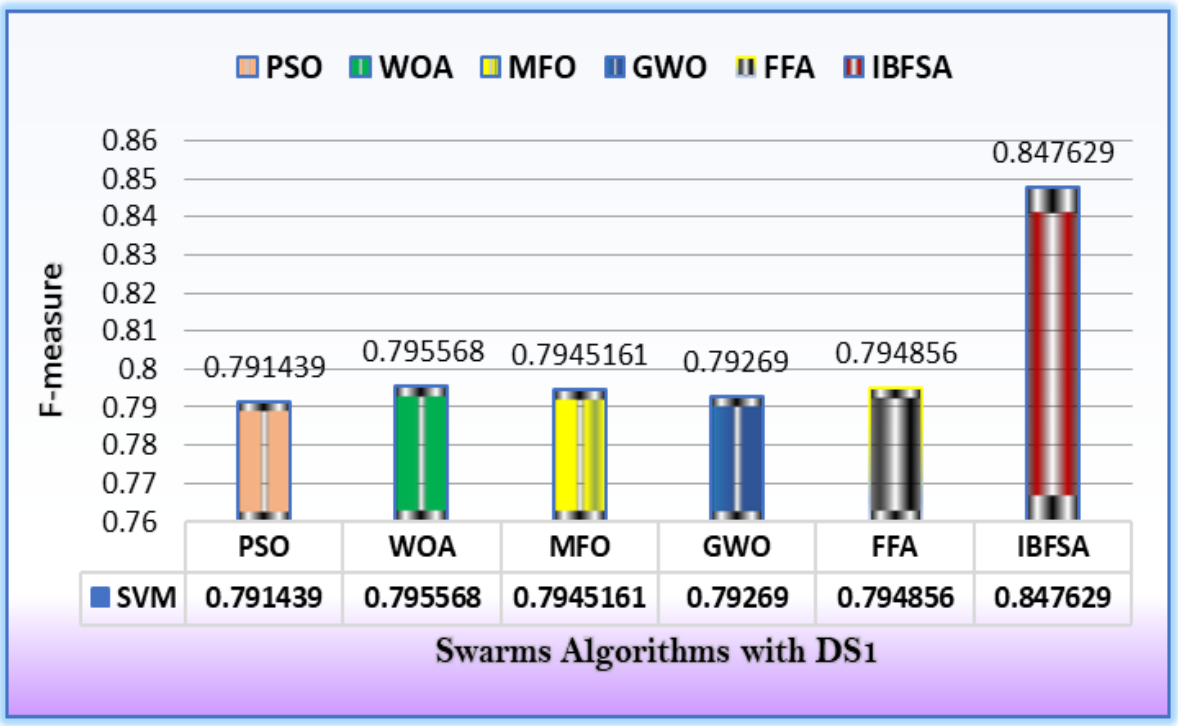

Figure 10. Average classification F-measure of IBFSA on DS1 compared with other algorithms by SVM Classifier.

Table 14. Comparison results of Classification performance obtained by LR algorithm with DS2.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	93.8849	91.6187	94.9721	92.7515	96.0674	94.2977	95.2646	93.5134
WOA	93.1655	91.8345	94.3820	92.8152	96.6292	94.5786	94.7075	93.6820
MFO	93.5252	92.3201	94.9153	93.2023	96.6292	94.9438	95.0276	94.0601
GWO	94.2446	90.9172	96.0227	92.9304	96.0674	92.8932	95.4802	92.9027
FFA	93.5252	91.4388	93.8889	92.7945	96.6292	93.9325	95.0276	93.3521
IBFSA	93.1655	90.4676	95.4286	92.5492	94.9438	92.5842	94.6176	92.5574

Table 15. Comparison results of Classification performance obtained by RF algorithm with DS2.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	94.2446	92.2302	93.5484	90.621	97.7528	95.9269	95.6044	93.1861
WOA	93.5252	92.5000	93.7500	91.1323	97.7528	96.0393	94.7368	93.5073
MFO	93.8849	92.6978	93.4066	91.1925	97.191	96.3202	95.0549	93.6793
GWO	93.1655	91.5287	94.7977	91.9839	97.7528	93.3988	94.7658	92.6582
FFA	92.8058	91.7985	94.3503	91.8734	96.6292	94.3258	94.1176	93.0739
IBFSA	92.8058	91.5468	94.8276	93.3384	93.2584	91.9663	93.7853	92.6419

Table 16. Comparison results of Classification performance obtained by MLP algorithm with DS2.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	92.8058	90.4496	94.7977	92.8646	96.0674	92.1909	94.3820	92.5100
WOA	92.4460	90.7014	94.3503	92.4501	97.1910	93.1460	94.0845	92.7617
MFO	92.4460	90.7554	95.8824	93.2032	95.5056	92.3595	94.1828	92.7467
GWO	92.8058	89.4964	95.4023	92.5473	96.6292	90.9550	94.5055	91.7095
FFA	92.8058	90.5755	94.6429	92.4290	97.1910	92.9213	94.5355	92.6571
IBFSA	92.4460	90.1798	95.3757	92.9065	94.3820	91.7135	94.0171	92.2804

Table 17. Comparison results of Classification performance obtained by NM algorithm with DS2.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	91.3669	88.4172	88.8889	86.6196	98.8764	96.9101	93.617	91.4698
WOA	89.5683	87.8057	88.601	85.9397	98.8764	96.882	92.2667	91.0656
MFO	92.446	88.7589	91.9786	87.2246	98.8764	96.6572	94.2466	91.6853
GWO	88.1295	85.1978	86.4322	82.5656	98.8764	97.528	91.2467	89.4133
FFA	90.2878	85.9712	88.3249	83.5081	98.8764	97.4157	92.8	89.9095
IBFSA	83.0935	77.7338	79.638	74.7827	100	98.5674	88.2206	85.0241

Table 18. Comparison results of Classification performance obtained by SVM algorithm with DS2.

Algorithm	Accuracy		Precision		Recall		F-measure	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
PSO	94.6043	92.7338	95.4802	93.4127	97.191	95.3932	95.8449	94.3868
WOA	93.8849	92.7877	94.4444	93.2464	97.191	95.6741	95.2909	94.4405
MFO	94.2446	93.0395	95	93.7031	97.191	95.5617	95.5801	94.6191
GWO	93.5252	91.5287	95.9064	93.3714	96.6292	93.4269	94.9438	93.3876
FFA	93.8849	91.6906	94.9721	93.0589	96.0674	94.0449	95.2646	93.5463
IBFSA	97.1119	95.0541	97.1098	94.1285	99.4186	98.1613	97.7011	96.0932

Classifiers results from machine learning's second dataset are displayed in Tables 14–18. As it can be seen from Tables 14–16 the classifiers achieved a promising performance compared to all methods, however, comparatively there is a marginal difference in accuracy between the classifiers. It is noteworthy, Table 17 shows that a NB classifier trained with IBFSA can prove superior efficacy compared to its other peers, achieving average classification sensitivity of 98.25% and a maximum sensitivity among the 20 runs is 100%. While, Table 18 shows that the IBFSA has the best accurate performance of all of the rivals regarding the SVM classifier, see Figure 11.

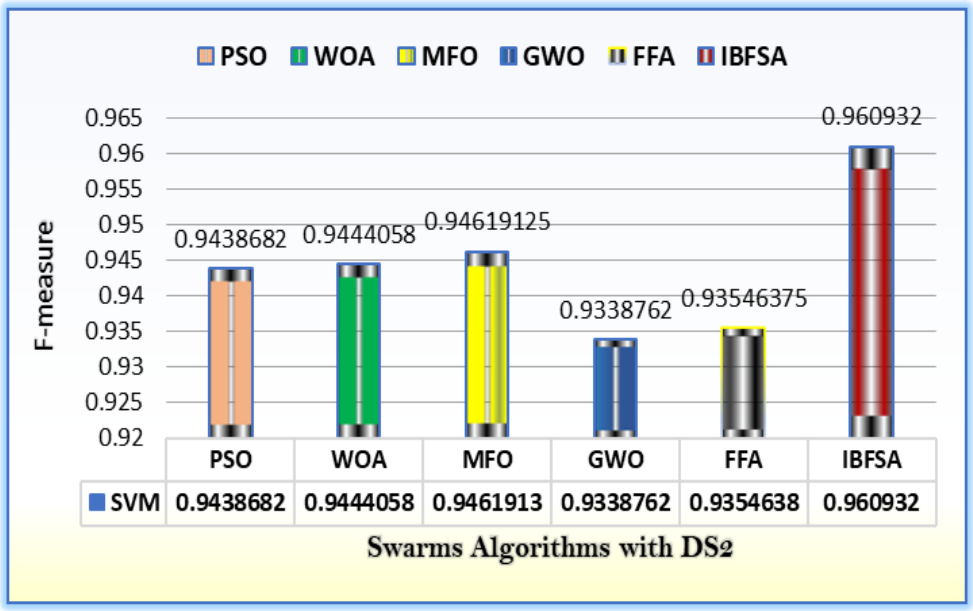

Figure 11. Average classification F-measure of IBFSA on DS2 compared with other algorithms by SVM Classifier.

As per results in Tables 13 and 18, it can be seen that the optimizer IBFSA with SVM classifier has demonstrated a greater classification accuracy in comparison to the other variations using LR, RF, MLP and NB classifiers in handling all selected datasets. One of the causes is that the SVM classifier uses over-fitting protection and does not depend primarily on the total number of processed features. So, it has better potential than previously studied classifiers in dealing with bigger text feature spaces. As seen in the results, when dealing with a sparsely of samples, the SVM can demonstrate a steadier efficacy compared to other models. On these particular datasets, the IBFSA algorithm achieves better results than any other competing approaches in terms of feature selection accuracy. The inclusion of new, more efficient components that improve the algorithm's balance between its exploratory and exploitative capacities is one possible explanation for the algorithm's improved performance.

6. Conclusions and future work

A new diagnostic model for COVID-19 has been developed that will effectively increase the final prediction accuracy. The suggested approach includes two primary stages. The first stage is utilizing RTF-C-IEF to determine the feature's importance. Next, the modified flamingo search

algorithm is then used to choose a collection of pertinent and non-redundant features in the second phase. Finally, the SVM-based classifier is used to predict COVID-19 using the features elected of clinical text. Our experiments were conducted on two sets of data, the first was collected from hospitals in the south of Iraq, and the second was from several sources on websites. In IBFSA, we presented four ways to boost both the global and local search capabilities of the algorithm. In addition, the continuous approach has been adapted to the binary feature selection problem using the binary transformation method. We have compared the suggested technique to state-of-the-art feature selection swarming methods such as PSO, MFO, GWO and FFA. Experiments reveal that the suggested technique is more effective in decreasing sub-features by more than 88% and with an accuracy superior to other methods. As a result, it can be concluded that the suggested approach is a powerful feature selection for COVID-19 patients' classification. Moreover, IBFSA reports that feature selection has decreased the number of diagnostic mistakes for COVID-19 patients. In this way, feature selection helps machine learning zero in on the most relevant information, lessening the likelihood of an incorrect diagnosis while attempting to distinguish between infected and uninfected individuals. In our future work, we'll take into account expanding and diversifying the test datasets to better assess the suggested methodology.

Data Availability

The original data (first dataset) used and/or processed during current study is part of the health records of a group of hospitals in southern Iraq. Therefore, data (DS1) is not available to the general public. May be made available from the corresponding author upon reasonable request.

Acknowledgments

The authors are so pleased to introduce their deep acknowledgment and great thanks to the staff of the hospitals and healthcare providers which supported the clinical data for this study, especially hospitals in Iraq.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. C. Li, C. Zhao, J. Bao, B. Tang, Y. Wang, B. Gu, Laboratory diagnosis of coronavirus disease-2019 (COVID-19), *Clin. Chim. Acta*., **510** (2020), 35–46. https://doi.org/10.1016/j.cca.2020.06.045
2. Y. Guo, Q. Cao, Z. Hong, Y. Tan, S. Chen, H. Jin, et al., The origin, transmission and clinical therapies on coronavirus disease 2019 (COVID-19) outbreak- A n update on the status, *Mil. Med. Res*., **7** (2020), 1–10. https://doi.org/10.1186/s40779-020-00240-0
3. M. Rostami, M. Oussalah, A novel explainable COVID-19 diagnosis method by integration of feature selection with random forest, *Inform. Med. Unlocked*, **30** (2022), 100941. https://doi.org/10.1016/j.imu.2022.100941

4. X. Luo, P. Gandhi, S. S. KH, A deep language model for symptom extraction from clinical text and its application to extract COVID-19 symptoms from social media, *IEEE J. Biomed. Heal Inform.*, **26** (2022), 1737–1748. https://doi.org/10.1109/JBHI.2021.3123192
5. G. Saranya, A. Pravin, Feature selection techniques for disease diagnosis system: A survey, in *Artificial Intelligence Techniques for Advanced Computing Applications*, Springer, Singapore, **130** (2021), 249–258. https://doi.org/10.1007/978-981-15-5329-5_24
6. J. T. Pintas, L. A. F. Fernandes, A. C. B. Garcia, Feature selection methods for text classification: A systematic literature review, *Artif. Intell. Rev.,* **54** (2021), 6149–6200. https://doi.org/10.1007/s10462-021-09970-6
7. L. M. Abualigah, A. T. Khader, E. S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, *J. Comput. Sci*., **25** (2018), 456–466. https://doi.org/10.1016/j.jocs.2017.07.018
8. D. A. Elmanakhly, M. Saleh, E. A. Rashed, M. Abdel-Basset, BinHOA : Efficient binary horse herd optimization method for feature selection : Analysis and validations, *IEEE Access*., **10** (2022), 26795–26816. https://doi.org/10.1109/ACCESS.2022.3156593
9. R. Abu Khurmaa, I. Aljarah, A. Sharieh, An intelligent feature selection approach based on moth flame optimization for medical diagnosis, *Neural Comput. Appl.*, **33** (2021), 7165–7204. https://doi.org/10.1007/s00521-020-05483-5
10. P. H. Prastyo, R. Hidayat, I. Ardiyanto, Enhancing sentiment classification performance using hybrid query expansion ranking and binary particle swarm optimization with adaptive inertia weights, *ICT Express*., **8** (2021), 189–197. https://doi.org/10.1016/j.icte.2021.04.009
11. B. Ji, X. Lu, G. Sun, W. Zhang, J. Li, Y. Xiao, Bio-Inspired feature selection : An improved binary particle swarm optimization approach, *IEEE Access*., **8** (2020), 85989–86002. https://doi.org/10.1109/ACCESS.2020.2992752
12. H. K. H. Chantar, M. M. Mafarja, H. I. Alsawalqah, A. A. Heidari, I. Aljarah, H. Faris, Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification, *Neural Comput. Appl*., **32** (2020), 12201–12220. https://doi.org/10.1007/s00521-019-04368-6
13. M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, L. Abualigah. Binary aquila optimizer for selecting effective features from medical data: A COVID-19 case study, *Math. MDPI*., **10** (2022), 1–24. https://doi.org/10.3390/math10111929
14. J. Piri, P. Mohapatra, B. Acharya, F. S. Gharehchopogh, V. C. Gerogiannis, A. Kanavos, et al., Feature selection using artificial gorilla troop optimization for biomedical data: A case analysis with COVID-19 data, *Mathematics*, **10** (2022), 1–31. https://doi.org/10.3390/math10152742
15. W. Tuerxun, X. Chang, G. Hongyu, J. Zhijie, Z. Huajian, Fault diagnosis of wind turbines based on a support vector machine optimized by the sparrow search algorithm, *IEEE Power Energy Soc. Sect*., **9** (2021), 69307–69315. https://doi.org/10.1109/ACCESS.2021.3075547
16. C. A. Flores, R. L. Figueroa, J. E. Pezoa, FREGEX: A feature extraction method for biomedical text classification using regular expressions, in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (2019), 6085–6088. https://doi.org/10.1109/EMBC.2019.8857471
17. W. M. Shaban, A. H. Rabie, A. I. Saleh, M. A. Abo-Elsoud, Accurate detection of COVID-19 patients based on distance biased Naïve Bayes (DBNB) classification strategy, *Pattern Recognit*., **119** (2021), 108110–108110. https://doi.org/10.1016/j.patcog.2021.108110

18. A. Singh, K. K. Singh, M. Greguš, I. Izonin, CNGOD-An improved convolution neural network with grasshopper optimization for detection of COVID-19, *Math. Biosci. Eng.*, **9** (2022), 12518–12531. https://doi.org/ 10.3934/mbe.2022584
19. Z. M. Fadhil, R. A. Jaleel, Multiple efficient data mining algorithms with genetic selection for prediction of SARS-CoV2, in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, (2022). https://doi.org/10.1109/ICACITE53722.2022.9823757
20. I. M. El-Hasnony, M. Elhoseny, Z. Tarek, A hybrid feature selection model based on butterfly optimization algorithm: COVID-19 as a case study, *Expert Syst.*, **39** (2022), e12786. https://doi.org/10.1111/exsy.12786
21. M. A. k. alsaeedi, S. Kurnaz, Feature selection for diagnose coronavirus (COVID-19) disease by neural network and Caledonian crow learning algorithm, *Appl Nanosci.*, (2022), 1–16. https://doi.org/10.1007/s13204-021-02159-x
22. T. Bezdan, M. Zivkovic, N. Bacanin, A. Chhabra, M. Suresh, Feature selection by hybrid brain storm optimization algorithm for COVID-19 classification, *J. Comput. Biol*., **29** (2022), 515–529. https://doi.org/10.1089/cmb.2021.0256
23. Z. Wang, J. Liu, Flamingo search algorithm and its application to path planning problem, in *2021 4th Flamingo search algorithm and its application to path planning problem*, (2021), 567–573. https://doi.org/10.1145/3488933.3489011
24. A. Onan, M. A. Toçoğlu, A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification, *IEEE Access*, **9** (2021), 7701–7722. https://doi.org/10.1109/ACCESS.2021.3049734
25. M. Neumann, D. King, I. Beltagy W. Ammar, ScispaCy: Fast and robust models for biomedical natural language processing, in *Proceedings of the 18th BioNLP Workshop and Shared Task*, (2019), 319–327. https:// doi.org/10.18653/v1/W19-5034
26. A. Y. Mahdi, S. S. Yuhaniz, Automatic diagnosis of COVID-19 patients from unstructured data based on a novel weighting scheme, *C. Mater. Contin*., **74** (2022), 1375–1392. https://doi.org/10.32604/cmc.2023.032671
27. T. Parlar, S. A. Özel, F. Song, A new feature selection method for sentiment analysis, *Human-centric Comput. Inf. Sci*., **8** (2018), 1–19. https:// doi.org/10.1515/jisys-2018-0171
28. S. L. Marie-Sainte , N. Alalyani, Firefly algorithm based feature selection for arabic text classification, *J. King Saud Univ. Comput. Inf. Sci.*, **32** (2020), 320–328, https://doi.org/ 10.1016/j.jksuci.2018.06.004
29. W. Zhiheng, L. Jianhua, Flamingo search algorithm: A new swarm intelligence optimization algorithm, *IEEE Access*., **9** (2021), 88564–88582. https://doi.org/10.1109/ACCESS.2021.3090512
30. M. Abd El Aziz, A. Hassanien, Modified cuckoo search algorithm with rough sets for feature selection, *Neural Comput. Appl*., **29** (2018), 925–934. https://doi.org/10.1007/s00521-016-2473-7
31. Z. Li, Y. Zhou, S. Zhang, J. Song, Lévy-Flight Moth-Flame algorithm for function optimization and engineering design problems, *Math. Probl. Eng*., (2016), 1–22. https://doi.org/10.1155/2016/1423930
32. P. A. Digehsara, S. N. Chegini, A. Bagheri, M. P. Roknsaraei, An improved particle swarm optimization based on the reinforcement of the population initialization phase by scrambled Halton sequence, *Cogent. Eng*., **7** (2020), 1–29. https://doi.org/10.1080/23311916.2020.1737383

33. B. Kazimipour, X. Li, A. K. Qin, A review of population initialization techniques for evolutionary algorithms, *2014 IEEE Congr. Evol. Comput.*, (2014), 2585–2592. https://doi.org/10.1109/CEC.2014.6900618
34. W. H. Bangyal, A. Hameed, W. Alosaimi, H. Alyami, A new initialization approach in particle swarm optimization for global optimization problems, *Comput. Intell. Neurosci.*, **2021** (2021), 1–17. https:// doi.org/10.1155/2021/6628889
35. A. G. Gad, K. M. Sallam, R. K. Chakrabortty, M. J. Ryan, A. A. Abohany, An improved binary sparrow search algorithm for feature selection in data classification, *Neural Comput. Appl.*, **34** (2022), 15705–15752. https://doi.org/10.1007/s00521-022-07546-1
36. P.H. Prastyo, A.S. Sumi, A.W. Dian, A. E Permanasari, Tweets responding to the Indonesian government's handling of COVID-19: Sentiment analysis using SVM with Normalized Poly Kernel, *J. Inf. Syst. Eng. Bus. Intell.*, **6** (2020), 112–122. https://doi.org/10.20473/jisebi.6.2.112-122
37. K. Kowsari, K. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, D. E. Brown, Text classification algorithms : A survey, *Inf. J.*, **10** (2019), 1–68. https://doi.org/10.3390/info10040150
38. M. Qaraad, S. Amjad, I. I. M. Manhrawy, H. Fathi, B. A. Hassan, P. E. Kafrawy, A hybrid feature selection optimization model for high dimension data classification, *IEEE Access.*, **9** (2021), 42884–42895. https://doi.org/10.1109/ACCESS.2021.3065341