Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

An Efficient Gannet Optimization Algorithm for Feature Selection based on Sensitivity and Specificity

Amir Mohammad Sharafaddini*[1] and Najme Mansouri†[2]

[1,2]Department of Computer Science, Shahid Bahonar University of kerman, Kerman, Iran.

ABSTRACT

The selection of features is a crucial step in the analysis of high dimensional data in machine learning and data mining. Gannet Optimization Algorithm (GOA) is a recently proposed metaheuristic algorithm that has not yet been investigated in terms of its capacity to solve feature selection problems. A new wrapper feature selection approach based on GOA is proposed to extract the best features. The GOA is a robust meta-heuristic algorithm that can deal with higher dimensions. A fitness function is used to account for the entropy of the sensitivity and specificity, as well as the accuracy of the classifier and the fraction of features selected. Additionally, four new algorithms are compared with the proposed algorithm in this paper. Based on the experimental results, fewer features can be obtained with a higher classification accuracy using the proposed algorithm.

Keyword: Wrapper approach, Optimization algorithm, Accuracy, Sensitivity.

ARTICLE INFO

Article history:
Research paper
Received 07, September 2022
Received in revised form 19, November 2022
Accepted 4, December 2022
Available online 30, December 2022

AMS subject Classification: 05C78.

*a.m.sharafaddini@gmail.com
†Corresponding author:N. Mansouri. Email: najme.mansouri@gmail.com

1 Introduction

Several fields, including social media, business, and scientific research, have experienced an enormous increase in data over the last two decades. Developed computer hardware and software, as well as online database technologies, allow us to collect and store large datasets from different sources more efficiently and effectively than ever before. Additionally, some applications have an extremely high level of dimensionality. "Big Data" can be described as a large amount of data that cannot be handled by traditional database software [18]. As a result, feature selection is an active research area and is widely applied to real-world problems, primarily classified problems, although other fields can also be applied, including regression. In addition, big data can be overfitted by the large number of feature spaces, leading to lagging performance for the unseen data items. The complexity of the classification model and the storage requirements increases significantly with high dimensional data. Data mining algorithms, as well as classification algorithms, perform best when they are closely related to the dataset's most valuable or important features [2]. It is necessary to employ methods that are effective at reducing the number of features in order to effectively reduce the dimensionality of data in order to cope with the advent of Big Data problems. For this problem, two basic approaches exist: feature extraction and feature selection. Instead of selecting only a subset of the original features, feature extraction methods combine existing and create new ones to reduce the dataset's dimensionality [22]. Figure 1 indicates an example of feature selection and feature extraction.

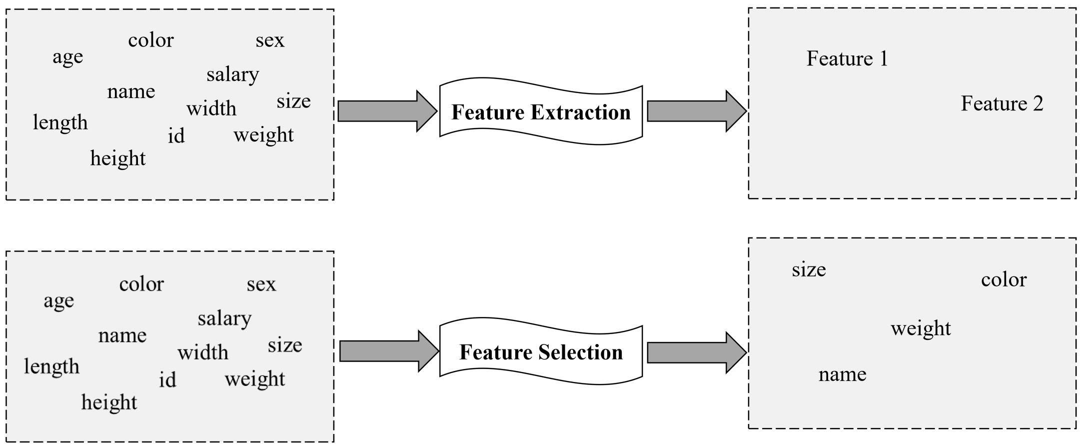


Figure 1: Example of feature selection and feature extraction.

The feature selection problem is an NP-hard problem with 2n states, where n is the number of features. It is becoming increasingly complex as n grows in many fields. An optimization problem may be designed to achieve trade-off solutions between two contradictory objectives, or it may be designed to aggregate these objectives into one optimization problem. Regardless of how many objectives are set, conventional statistical methods cannot be used with an exhaustive search approach in the era of big data. The random search approach is also inefficient because it selects a subset of features randomly. Meanwhile, metaheuristic methods are well-tested in this scenario for solving this class of problems. It is known that many metaheuristic algorithms (e.g., Particle Swarm Optimization (PSO), Grasshopper Optimization Algorithm (GOA) [21], and Genetic Algorithm (GA)) have been extensively studied for solving feature selection problems. Due to the No-Free-Lunch (NFL) theorem, there is no algorithm which can solve all types of optimization problems at once. Therefore, some optimization algorithms are better than others depending on the problem, but not always. Since then, a number of new and effective metaheuristic algorithms for solving engineering optimization problems have been developed. In 2022, Gannet Optimization Algorithm (GOA) was developed by simulating gannet behaviors during foraging [19]. Based on the unique behavior of gannets, a novel intelligent optimization algorithm has been developed. There are many benefits to using this method, including achieving early convergence and improving results on different benchmarks. In this paper, a GOA-based feature selection algorithm is presented. In most fitness function-based wrappers, accuracy and error rate of the classifier are combined with the percentage of features selected [4]. It is not acceptable to use accuracy as a metric when there is a problem of class imbalance [10]. As a result, specificity and sensitivity are used as appropriate metrics. This paper uses a fitness function that balances the sensitivity and specificity of the features as well as maximizing accuracy. Despite an imbalanced dataset, it improves performance of the classifier by balancing the true positive and true negative rates. Several contributions can be summarized in this article:

1. Propose a binary version of Gannet Optimization Algorithm (GOA).
2. Propose a new feature selection algorithm based on the predation behavior of gannets.
3. Consider the entropy of the sensitivity and specificity, as well as the accuracy of the classifier and the fraction of features selected.
4. Evaluate fitness value, classification accuracy, F-measure across multiple dimensions.
5. Experiments of the proposed method with four optimization methods on six benchmark datasets are conducted.

2 Background

2.1 Feature selection

The purpose of feature selection methods is to reduce the number of redundant and less informative features in order to speed up model training and improve accuracy, especially when dealing with high-dimensional datasets [17]. In order to build a model from the data, feature selection must connect to the algorithm being used to learn. Feature selection approaches can be classified into wrapper, embedded, and filter approaches according to their evaluation criteria.

- In contrast to learning algorithms, filters select the optimal feature subset based on the general characteristics of the data. A feature's score (subset) is determined by an evaluation criterion. A set of features is then selected based on the highest scores. Multivariate or univariate measures may be used in the evaluation. In contrast to multivariable measures, univariable measures focus on the relationship between each feature individually. This implies that multivariable measures have the capability of detecting redundant features and therefore are regarded as more general measures.

- It is the learner's responsibility to measure the goodness of the subsets of features proposed by the wrapper. By using wrappers to achieve better feature subsets, learning algorithms are more likely to perform better. However, wrappers are typically more computationally intensive than filters. Using search strategies, the wrappers obtain a subset of features. As a second step, a learning algorithm evaluates the quality of the selected feature subset. Until the stopping criterion is met, this procedure is repeated.

- In an embedded approach, feature selection is embedded into the process of learning algorithms. Typically, wrappers and filters are traded off. In this way, they take advantage of the characteristics of wrappers and filters. Firstly, they cooperate with the learning algorithm. As a result, they are more efficient than wrappers since they do not have to run the learning algorithm several times. Learning is often not better with embedded approaches than with wrappers. Figure 2 shows types of feature selection.

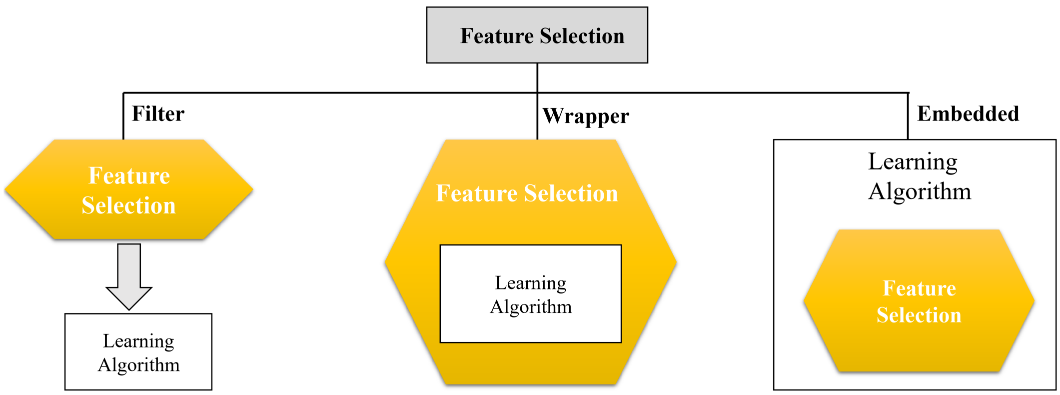


Figure 2: Types of feature selection [20].

As feature selection problems arise from a vast array of possible subsets, their complexity is derived from selecting the most relevant set of features. Combinatorial problems are introduced by feature selection and optimization techniques that cannot be solved easily. Thus, in search of better solutions for complex challenges, metaheuristics-based algorithms entered the picture and became more widely used in the literature. Algorithms based on metaheuristics are used to solve numerous kinds of optimization problems with the benefit of self-learning operators interacting with actors, enabling investigation of solutions in order to arrive at the best solution [3].

2.2 Metaheuristic algorithms

The method of solving the feature selection problem by using a metaheuristic algorithm is described in this section. The relevant features are obtained using binary vector representations. The designed algorithm represents a solution vector by (10101100 . . . .). In this case, 1 means a particular feature is selected in the subset, and 0 means the feature is not selected.

An overview of the main activities carried out by metaheuristic algorithms is shown in Fig. 3. The first step involves creating an initial population and calculating the fitness values. After that, the iterations begin. By exploring and exploiting metaheuristic operators, new candidate solutions are generated given a termination condition. During optimization, the same solutions should not be analyzed repeatedly. Recalculating the recombination operators of the metaheuristics is not necessary before each run, since it is possible that the same candidates will be generated repeatedly. Furthermore, due to their computationally expensive nature, faster versions of these algorithms, such as parallel or dynamic programming, can produce better results thanks to their increased number of fitness evaluations.

Figure 3 shows types of feature selection.

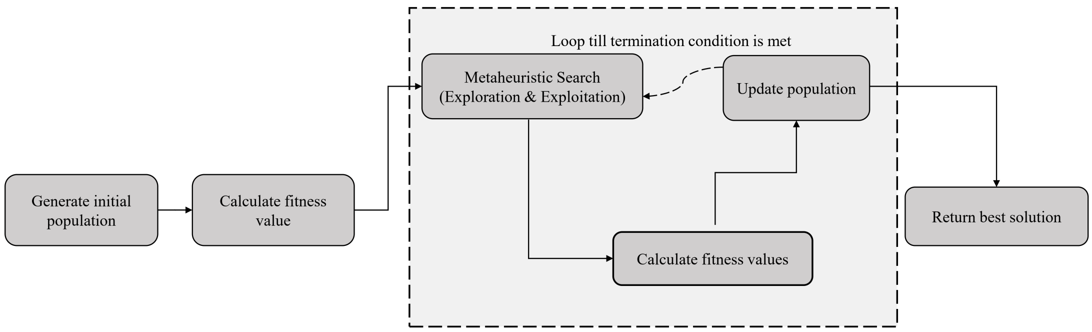


Figure 3: An overview of the main steps taken by metaheuristic for feature selection [6].

2.2.1 Gannet Optimization Algorithm (GOA)

Gannet Optimization Algorithm (GOA) is a nature-inspired metaheuristic algorithm developed in 2022. A variety of mathematical models are presented in the model to simulate gannets' unique behavior during predation. In the exploration phase, Gannets determine which area is best by diving in U-shaped and V-shaped patterns. The sudden rotation and random walk during the development process result in a better solution. During

the optimization process, the GOA produces random initial solutions. As each iteration progresses, each individual adjusts their position in accordance with the four formulas provided by GOA. A formula is chosen for position update during exploration equally, whereas a different formula is chosen for position update during exploitation based on the catching capability. Diverse methods and turning search processes are integral to gannet feeding. Consequently, the algorithm selects both exploration and exploitation phases equally in every iteration. Several optimizations are carried out to attain an optimal or near-optimal solution. Finally, the iteration ends when the results satisfy the end criteria. By designing the memory matrix appropriately, the algorithm can achieve better convergence speed. Furthermore, the iterative curve illustrates GOA's ability to explore and escape local optima more intuitively. Moreover, GOA has excellent performance advantages as the dimensions increase, so it can handle large-dimensional problems well. According to experimental results, GOA outperformed many existing algorithms on five engineering design problems. The GOA is divided into three phases:
Initialization phase: As a starting point, the GOA considers a set of random solutions called matrix M, followed by the optimal solution. A memory matrix is defined as an MX matrix. During the initialization phase, MX is assigned the values of the X matrix. In each iteration of evolution, the MX memory matrix will record the position change of each gannet individual. A memory matrix MX_i can be substituted for X_i if it performs better than the current solution X_i based on the fitness function. In other cases, the X matrix solution is used. If MX_i performs better than X_i based on the fitness function, then MX_i is used instead of X_i. Alternatively, the X matrix solution continues to be used.
Exploration phase: After locating their prey, gannets adjust their dive patterns to catch it based on the depth of the dive. There are two types of diving: deep, long U-shaped dives and shallow, short V-shaped dives [12]. U-shaped dives are calculated using Eq. 2, while V-shaped dives are calculated using Eq. 3.

$$\mathrm{T} = 1 - \frac{It}{Tmax\_iter} \tag{1}$$

$$\mathrm{a} = 2 \times \cos(2 \times \pi \times r_1) \times t \tag{2}$$

$$\mathrm{b} = 2 \times V(2 \times \pi \times r_2) \times t \tag{3}$$

$$\mathrm{V(x)} = \begin{cases} \frac{-1}{\pi} \times x + 1, x \in (0, \pi) \\ \frac{-1}{\pi} \times x - 1, x \in (\pi, 2\pi) \end{cases} \tag{4}$$

Where $Tmax_iter$ is the maximum number of iterations, It is the number of iterations currently being performed. In both r_1 and r_2, the number is random between 0 and 1. Next, the position should be updated using these two dive strategies. In order to determine which dive strategy Gannet will choose at a random rate when predating, a random number q is defined for this process. Eq. 5 explains how to update the position.

$$MX_i(t+1) = \begin{cases} X_i(t) + u1 + u2, q \geq 0.5 & ,(a) \\ X_i(t) + v1 + v2, q < 0.5 & ,(b) \end{cases} \tag{5}$$

$$\mathrm{u2} = A \times (X_i(t) - X_r(t)) \tag{6}$$

$$\mathrm{v2} = B \times (X_i(t) - X_m(t)) \tag{7}$$

$$\mathrm{A} = (2 \times r_3 - 1) \times a \tag{8}$$

$$\mathrm{B} = (2 \times r_4 - 1) \times b \tag{9}$$

There are two random numbers r_3 and r_4 between 0 and 1, $u1$ is a random number between $-a$ and a, and $v1$ is a random number between $-b$ and b. It is assumed that $X_i(t)$ is i-th member of the current population and that $X_r(t)$ is a randomly selected member of the current population. The following equation can be used to determine $X_m(t)$, where $X_m(t)$ represents average position in the present population.

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \tag{10}$$

Exploitation phase: The capture capacity is defined by Eq. (11). Gannets are able to catch fish if they possess sufficient energy, i.e., if their capture capacity is high. It is likely that the gannet's energy will decrease as the time goes on, and the bird will not be able to capture its prey.

$$\text{Capturability } = \frac{1}{R * t2} \tag{11}$$

$$t2 = 1 + \frac{It}{Tmax_iter} \tag{12}$$

$$R = \frac{M * vel^2}{L} \tag{13}$$

$$L = 0.2 + (2 - 0.2) * r_5 \tag{14}$$

A random number between 0 and 1 is assigned to r_5 and the weight of the gannet (M) is 2.5 kg. Considering no resistance in the water, $Vel = 1.5$ m/s represents the gannet's underwater speed. Whenever the gannet's ability to catch prey is within its range, the position is updated by turning suddenly. The gannet may also use a Levy movement when it cannot catch this flexible fish, as shown in Eq. 15.

$$MX_i(t+1) = \begin{cases} t * \text{ delta } * (X_i(t) - X_{\text{best }}(t)) + X_i(t), \text{ Capturability } \geq c, (a) \\ X_{\text{best }}(t) - (X_i(t) - X_{\text{best }}(t)) * P * t, \text{ Capturability } < c, (b) \end{cases} \tag{15}$$

$$\text{delta } = \text{ Capturability } * |X_i(t) - X_{\text{best }}(t)| \tag{16}$$

$$P = leavy(Dim) \tag{17}$$

After several experiments, the constant c was determined to be 0.2. A Levy flight function [15] is determined by Eq. 18 by taking $XBest(t)$ and $Levy()$ from the current population.

$$\text{Levy(Dim)} = 0.01 \times \frac{\mu \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{18}$$

$$\sigma = \frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}\Big)^{\frac{1}{\beta}}} \tag{19}$$

Based on this assumption, μ is a random values between 0 and 1, and β is set to a constant value of 1.5.
There are three main processes that contribute to the complexity of the GOA: initialization, fitness function calculation, and updating the gannet positions. GOA determines each individual's fitness value according to the number of individuals N in the population. Thus, the complexity is $O(N)$. In this analysis, fitness functions are not included because they are calculated according to each specific problem. Using the last process, the GOA complexity is $O(TN) + O(TND)$. Where, T is the maximum number of iterations and D is the problem's dimensionality. GOA's computational complexity is then determined by $O(N(TD + 1))$.

2.3 Support Vector Machine (SVM)

In machine learning, a support vector machine is based on supervised learning. By determining support vectors from labeled training data samples, SVMs use statistical learning theory to classify data. In SVMs, the main goal is to find the best hyperplane for categorizing new data points. SVMs are binary classifiers that classify multidimensional data by creating hyperplanes using some nearest training data points for each class and maximizing their margins [14]. The support vectors are only derived from a subset of training data points. SVM technique has several advantages such as:

- In cases where we don't know what the data is, SVMs are very good.
- High-dimensional spaces are more suitable for SVM.
- When the number of dimensions exceeds the number of samples, SVM is effective.
- SVM is a relatively memory-efficient algorithm.
- In practice, SVMs are more generalized, so over-fitting is less likely.
- Complex problems can be solved using an appropriate kernel function.

3 Related Works

A number of metaheuristic approaches have been developed to solve the feature selection problem recently that demonstrate superior performance to traditional feature selection approaches. An optimization approach based on Particle Swarm Optimization (PSO) and Gray Wolf Optimization (GWO) was proposed by Dhal and Azad [5] as a binary version of the hybrid two-phase multi-objective FS approach. An initial global search will be performed, followed by a second local search. For global search, the PSO property is employed. A modified version of PSO and GWO is used to perform local search in the proposed method. Incorporating Law of Motion into the hybrid approach enhances its effectiveness. A minimum classification error rate must be achieved and the number of features selected must be minimized. A high-dimensional gene expression study is conducted to assess the proposed methodology's performance. A comparison of the proposed approach with other metaheuristics, statistics, and multi-objective FS approaches shows that it is more efficient and effective than other approaches.

According to Fang and Liang [7], the Nonlinear Binary Grasshopper Whale Optimization Algorithm for Feature Selection (NL-BGWOA) is a hybrid algorithm that can maximize the number of grasshoppers and whales. This method proposes a combined position updating strategy that optimizes the diversity of searching within the target domain by combining whale and grasshopper population changes. By expressing the datasets in the iteration with fewer features, the proposed method ensures the goodness of feature subsets and can increase the efficiency of the optimization algorithm. A study of the FS problem of data with high dimensions reveals that NL-BGWOA has a comprehensive advantage.

A method designed by Halim et al. [9] preserves the majority of unique data information with a minimum number of features. GA-based Feature Selection (GbFS) is a method for selecting features with Genetic Algorithms (GAs) that increases the classifiers' accuracy. Besides providing parameter optimization for GA-based feature selection, they present a novel fitness function for GA-based feature selection. The fitness values were assigned to the GA individuals, allowing for the selection of the chromosomes with the best features. From the IDS datasets, this approach selected the most appropriate features.

According to Got et al. [8], whale optimization algorithm (WOA) is used to select hybrid features for filter-wrappers in a hybrid filter-wrapper approach. With WOA, it is possible to locate promising regions in the feature space. As a result, it combines the advantages of filtering and wrapping into a single system for improved performance. Optimizing involves considering two objective functions. First, the mutual information (MI) is used to identify non-dominated subsets of features by estimating relevance and redundancy. In the second objective, a learning classifier is used as a wrapper fitness function in order to estimate classification accuracy. As a result of the experimental results, we are able to conclude that the proposed algorithm is more efficient both in terms of the number of features and classification performance when compared to the selected approaches.

Using a global search strategy, Abd Elaziz et al. [1] developed a more effective atomic orbital search algorithm. Arithmetic Optimization Algorithm (AOA) operators are employed to produce a promising candidate solution. As the initial population increases, Opposite-Based Learning (OBL) will enhance convergence towards the optimal solution. Additionally, a dynamic photon rate is used for simultaneous exploration and exploitation. To increase classification accuracy by finding relevant features, Sequential Backward Selection (SBS) is used as a final step in the process. In comparison with the other performance measures, the proposed algorithm performed better. A summary of five algorithms is presented in Table 1.

Table 1: A summary of related works

Paper	Year	Algorithm(s)	Compared Methods	Classifier	Objective Function(s)	Disadvantage(s)
[5]	2021	PSO + GWO	GA – BPSO – BGWO - MOPSO – MOFA – Rc-BBFA -NSGA-II	KNN – DT – RF – NB – SVM	Accuracy – Selected features	The algorithm involves more parameters.
[7]	2022	NL-BGWOA	PSO (BPSO) – WOA (BWOA) – GOA (BGOA) – NL-BGWOA	5-KNN	Not reported	As for classified datasets with fewer features, there is still a need to improve the accuracy and fitness of the classification algorithm.
[9]	2021	GA	Recursive feature elimination, Sequential feature selector, Correlation based feature selection	SVM – KNN – Xg-Boost	Accuracy – computed correlation matrix	Suitable for smaller datasets, but it hasn't been tested for high-dimensional datasets.
[8]	2021	FW-GPAWOA	FW – BDE – BPSO – BGWO – WOA – JDE	KNN	filter function (selected feature)– wrapper function (accuracy)	Increased number of classes causes high running time and poor performance.
[1]	2022	f IAOS-SBS	SCA – BPSO – SSA – DE – HGSO – SCAWDO	KNN (k=5)	Number of selected feature – number of total feature - error of classification	Incomplete fitness function.

4 Proposed Feature Selection Algorithm

A description of the proposed GOA-based Feature Selection algorithm (GOAFS) is provided in this section. Figure 4 indicates the general framework of the proposed algorithm.

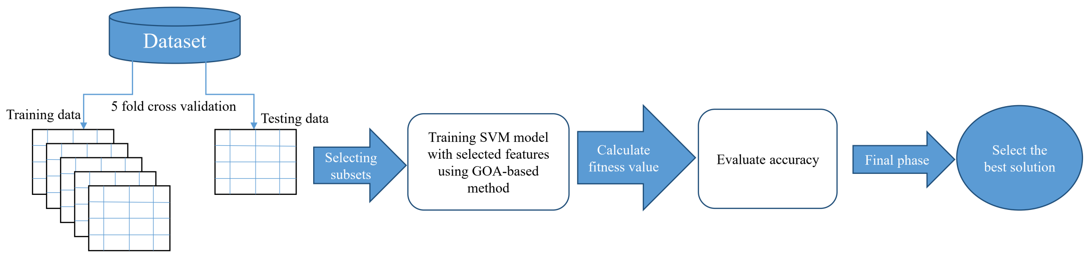


Figure 4: General framework of the proposed method.

4.1 Solution representation

The solution to a problem needs to be represented in the metaheuristic algorithm. Based on the original dataset, a one-dimensional vector containing N elements is produced; N represents the number of features. Each cell in the vector has a value of 1 or 0. A value of "0" indicates that no feature is selected. otherwise, a value of "1" indicates that the feature is selected [13]. Figure 5 indicates the binary representation. In this paper, the

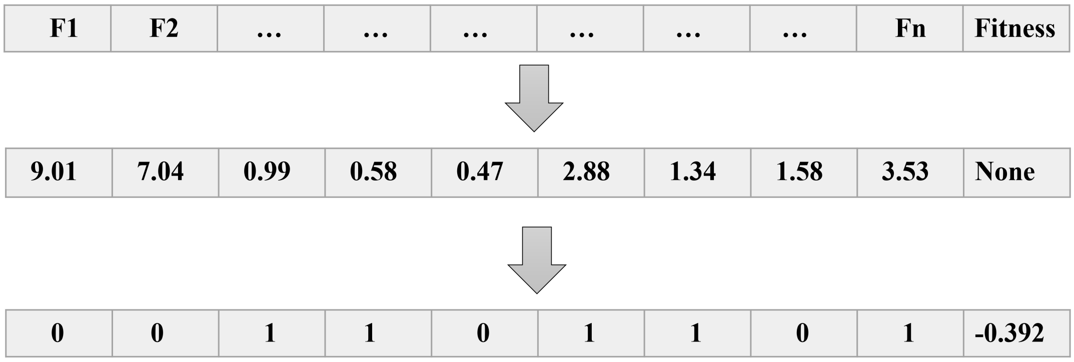


Figure 5: Binary representation.

solutions are converted to binary using the sigmoid function:

$$\operatorname{sigmoid}\left(x_i^d\right)=\frac{1}{1+\exp\left(-x_i^d\right)} \tag{20}$$

$$\operatorname{binary}\left(x_i^d\right)=\begin{cases} 1 \text{ if sigmoid } \left(x_i^d\right)>r_d \\ 0 \text{ else}\end{cases} \tag{21}$$

Which r_d is a random number between 0 and 1.

4.2 Objective function

The proposed approach uses the Support Vector Machine (SVM) classifier, which is efficient at determining the classification effectiveness of a subset of features. Feature selection aims to reduce the number of selected features as well as increase classification accuracy in order to achieve great classification performance. The fitness function must take into account both contradictory goals simultaneously [23].
In most fitness function-based wrapper approaches, the efficiency of the classifier is measured by its accuracy or error rate. A classifier's accuracy is derived as the percentage of correct classifications as shown in Eq. 25. FP, TP, TN, and FN correspond to false positives, true positives, true negatives, and false negatives, respectively. When dealing with class imbalance problems, accuracy is not an effective metric. Thus, specificity and sensitivity should be considered as appropriate metrics. Sensitivity and specificity can be determined Eqs. 22 and 23, respectively.

$$\text{Sensitivity: } \frac{TP}{TP+FN} \tag{22}$$

$$\text{Specificity: } \frac{TN}{TN+FP} \tag{23}$$

According to Eq. 24, a hybrid wrapper fitness function is used.

$$\psi=\gamma\left(\frac{F_S}{F_T}\right)+\beta\left(p\log_2(p)+q\log_2(q)\right)+\alpha\left(1-A_c\right) \tag{24}$$

In this case, p denotes the sensitivity fraction when compared to the sum of specificity and sensitivity, and q denotes the specificity fraction when compared to the sum of specificity and sensitivity. Features selected are FS, features total is FT. Accuracy (Ac) is given by Eq. 25.

$$\text{Accuracy: } \frac{TP+TN}{TP+FP+TN+FN} \tag{25}$$

In addition, $p + q = 1$. According to Fig. 6, the term $p\ \log_2(p) + q\ \log_2(q)$ has the minimum value when sensitivity equals specificity. As a consequence, it attempts to balance specificity with sensitivity. In addition, the fitness function tries to minimize features used, maximize accuracy, and balance sensitivity and specificity. In this way, so long as there is a balance between true positives and true negatives of the dataset, the classifier will perform well, even when there is an imbalance in the dataset. In the fitness function, each parameter is weighed according to its contribution to the function. The small number of features might lead to a subset getting selected even if accuracy and entropy are low. Due to this, accuracy and entropy should weigh more heavily than features selected based on specificity and sensitivity. In this paper α=0.01,γ=0.495,β=0.495 [16].

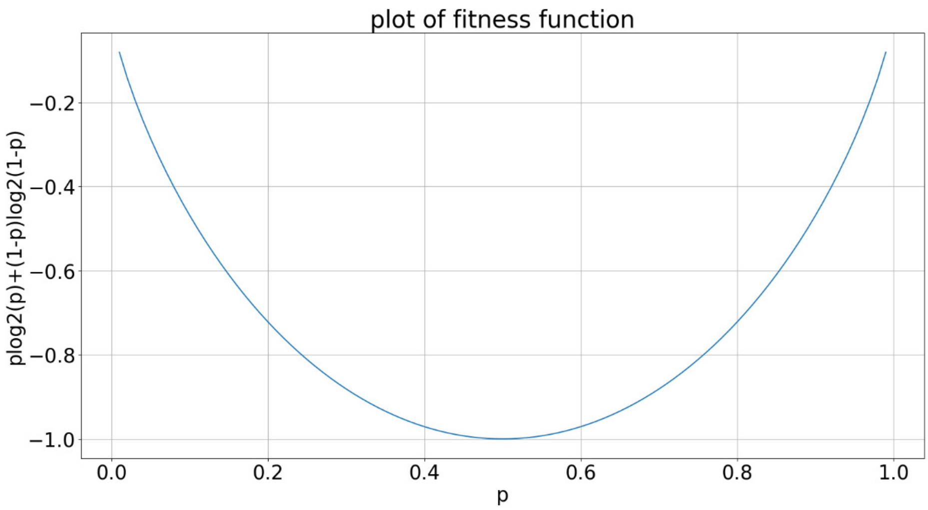


Figure 6: . Plot of function $p \times \log_2(p)+\ (1-p) \times \log_2(1-p)$ indicating the minimum value at p = 0.5 [16]

4.3 GOA based Feature Selection (GOA-FS)

Feature subsets in GOA-FS are represented by gannet positions. The original set may have N features, where N is the number of features in each subset. Based on the proposed fitness function, each solution is evaluated in accordance with three objectives: accuracy of SVM classification, specificity, sensitivity of founded solutions, and number of selected features.

Algorithms create population matrices by randomly selecting solutions (subsets). After that, the fitness function is applied to each population matrix solution. Next, the population memory matrix needs to be calculated. Iterations are performed in GOA's main loop. A population memory matrix solution is updated in each iteration by either Eq. 5 or Eq. 15 based on random numbers. The fitness function determines each individual's performance. Population memory matrix individual should be replaced with population matrix individual if fitness value of i-th individual in population matrix is lower than fitness value of i-th individual in population memory matrix. As a stopping criteria, the process is usually stopped after a maximum number of iterations. GOAFS pseudo-code is presented in Algorithm 1.

Algorithm 1. Pseudo-code of the GOA-based feature selection.

```
Input: N: population size; D: problem dimension; Epoch: maximum number of iterations;
Output: The location of Gannet and its fitness value;
Initialize the population matrix X randomly;
Calculate the fitness value of X;
Generate the population memory matrix;
While stopping condition is not met do
  If rand1 > 0.5 then
    If rand2 > 0.5 then
      for MX_i do
        Update the location Gannet using Eq. (5a);
      end for
    else
      for MX_i do
        Update the location Gannet using Eq. (5b);
      end for
    end if
  If rand3 > 0.5 then
      for MX_i do
        Update the location Gannet using Eq. (15a);
      end for
    else
      for MX_i do
        Update the location Gannet using Eq. (15b);
      end for
    end if
  end if
  for MX_i do
    Convert each individual to binary version using sigmoid function;
    Calculate the fitness value of MX_i ;
    If the value of MX_i is better than the value of X_i , replace X_i with MX_i;
  end for
end while
```

5 Experimental results

5.1 Experimental setup

A detailed experimental analysis of the proposed feature selection method is presented in this section. For the purpose of demonstrating the effectiveness of our method, we run experiments on six public datasets obtained from GitHub and the UCI machine learning repository. Table 2 describes the datasets used in detail. The interval [0-1] is used to normalize all continuous features. A quantitative comparison is conducted between GOAFS and four metaheuristics proposed recently, namely Beluga Whale Optimizer (BWO) [25], Genetic Algorithm (GA), Dandelion Optimizer (DO) [24] and Geometric Octal Zones

Table 2: Information about the datasets.

Datasets	**Classes**	**Instances**	**Features**	**Classification accuracy before applying feature selection**
BreastEW	2	568	30	0.88
HeartEW	2	270	13	0.66
Vote	2	300	16	0.93
Divorce	2	170	54	0.97
Iris	2	150	4	0.78
Ionosphere	2	351	34	0.94

Distances Estimation (GOZDE) [11]. These algorithms are summarized as follows.

- Gannet Optimization Algorithm (GOA): The GOA is a mathematical model for exploring and exploiting various unique behaviors of gannets during foraging. With sudden turns and random walks, GOA's diving patterns determine the area within the search space in which better solutions can be found based on its U- and V-shaped patterns.

- Genetic Algorithm (GA): Natural selection, the mechanism driving biological evolution, is used to solve both constrained and unconstrained optimization problems. A genetic algorithm continually modifies an individual solution. The genetic algorithm selects each parent and produces children for the next generation based on the current population. Generations of a population evolve toward an optimal solution.

- Dandelion Optimizer (DO): Three stages are involved in the simulation of long-distance flight of a dandelion seed relying on wind. When seeds are rising, eddies from above or local drifts, depending on weather conditions, cause them to spiral upward. As they descend in global space, flying seeds continuously adjust their directions. As seeds land on the ground, they grow in randomly selected positions.

- Beluga Whale Optimization (BWO): Each phase of exploration consists of three phases, corresponding to pair swim, prey, and whale fall behaviors. The balance factor and whale fall probability are self-adaptive factors that control BWO's exploration and exploitation abilities. In order to enhance global convergence, Levy flights are also introduced during the exploitation phase.

- Geometric Octal Zones Distance Estimation (GOZDE): Based on the distance between zones and median values, a search scheme is used to share information between zones. There are eight zones in the search space, which are combinations of different search strategies. The whole population represents the eight zones.

Each algorithm's parameter settings and common settings are presented in Table 3.
In order to validate the performance of the proposed GOAFS against competitors, it was independently evaluated 10 times for each dataset. As a result, the following key performance measures were adopted for the FS problem:

Table 3: Setup of parameters for all algorithms.

Algorithms	**Parameters**
All algorithms	Population size N= 100 Maximum number of iterations T = 100 Dimensionality D = Number of features in the datasets Number of runs for each method= 10
GOA	Weight of the gannet M = 2.5 Gannet speed Vel = 1.5 c = 0.2 Constant in Levy flight function = 1.5
GA	Mutation rate= 0.6 Cross over rate= 0.4
BWO	Constant in Levy flight function β =1.5
DO	S= 0.01 and β = 1.5 in Levy function
GOZDE	No constants

Fitness value: By executing the algorithm 10 times independent of each other, the classification error rate can be reduced and the number of selected features can be minimized, as described in Eq. 24. If the value is lower, the solution is more optimal.
F-measure: Based on statistical analysis, F-measures are used to determine the accuracy of classification tests:

$$2 \times \frac{\text{presicion} \ \times \ \text{recall}}{\text{presicion} \ + \ \text{recall}} \tag{26}$$

A precision is calculated by dividing the number of true positives by the total number of positive results, which includes incorrectly identified results. Recall represents the number of positive results divided by the total number of samples which should have been positive.
Classification accuracy: In this metric, the correct classification rate is estimated. To evaluate the accuracy of the model, we used cross validation accuracy. Using cross-validation is one of the most accurate ways of measuring the impact of a machine learning algorithm on real data when using a classification algorithm for prediction. With K-fold cross validation, original data is randomly divided into K parts, with one part being used as test data and the other as training data. From the results of K repeated experiments, the average error is calculated. This study used K=5 for *K-Fold*.
Size of selected features: An indicator of how many features have been selected.

5.2 Evaluation of the proposed algorithm

The box plots in Fig. 7 show the classification performance of each algorithm. In four of six datasets with the highest median, GOA has higher boxplots as shown in Fig. 7. The number of selected features is also one of the goals that we have, compared to other optimization methods.
GOAFS performs well in BreasetEW, divorce, HeartEW, and vote datasets, as shown in Fig. 7. In order to effectively explore the search space, GOA calculates the gannet's U-shaped and V-shaped dive patterns. Among all selected datasets, GOA was ranked first for number of selected features, followed second by BWO, third by GA and fourth both by DO and GOZDE; GOA performed well in both low and high dimensions of feature selection. The following table compares GOA-based feature selection algorithm

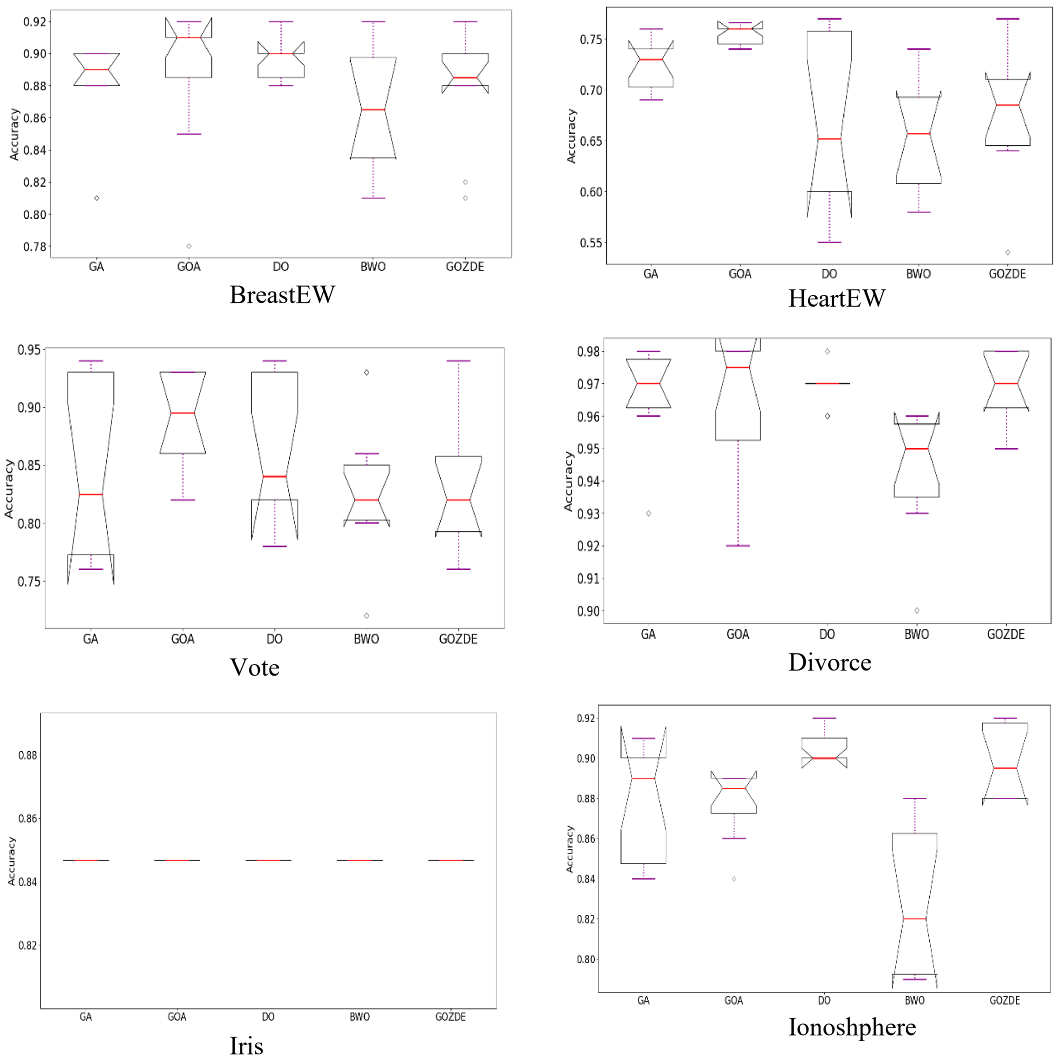


Figure 7: Box plots of classification accuracy on different datasets.

with other competitive methods in terms of classification accuracy (*Best*), number of selected features (N) and fitness value (*Fit*). With reference to the results obtained in Table 4.

Based on Table 4, it can be seen that DO and GOZDE perform almost equally in most datasets, but DO has better performance than GOZDE. GOZDE can need more analysis to produce a better configuration and first zone, because the parameter vector needs to be more detailed. For observing the effects of changing the reference area with any of the other zones, it can also tend to be stuck in local minima. Fitness values for GOA and BWO range from -0.43 to -0.47 in all datasets, but GOA performed better than BWO. Like GOA, BWO shows good performance by ensuring a good balance between exploration and exploitation phases. The GOA has a lower classification accuracy in the ionosphere than GA, DO, and GOZDE, but it has selected a smaller number of features.

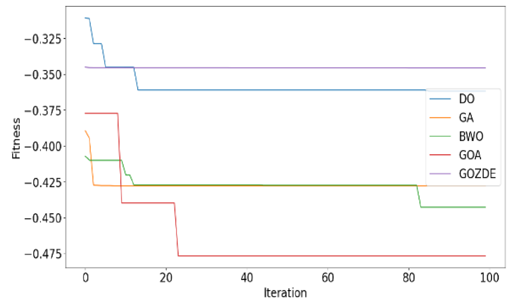
Table 4: Comparison between the proposed approaches with different methods.

Datasets	Measure	GOA	BWO	GA	DO	GOZDE
BreastEW	Best	0.92	0.92	0.90	0.92	0.92
	N	2	3	4	8	9
	Fit	-0.476	-0.442	-0.427	-0.361	-0.345
	f	0.91	0.91	0.88	0.91	0.91
HeartEW	Best	0.76	0.74	0.76	0.77	0.77
	N	1	1	1	3	4
	Fit	-0.454	-0.451	-0.453	-0.461	-0.414
	f	0.76	0.75	0.76	0.77	0.77
Vote	Best	0.93	0.93	0.94	0.94	0.94
	N	1	1	1	3	3
	Fit	-0.463	-0.461	-0.461	-0.432	-0.4309
	f	0.94	0.92	0.92	0.94	0.94
Divorce	Best	0.98	0.99	0.98	0.98	0.98
	N	6	7	12	19	18
	Fit	-0.48	-0.43	-0.38	-0.32	-0.32
	f	0.74	0.95	0.89	0.89	0.89
Iris	Best	0.84	0.84	0.84	0.84	0.84
	N	1	1	1	1	1
	Fit	-0.36	-0.36	-0.36	-0.36	-0.36
	f	0.67	0.67	0.67	0.67	0.67
Ionosphere	Best	0.89	0.88	0.91	0.92	0.92
	N	3	4	8	11	10
	Fit	-0.449	-0.448	-0.377	-0.348	-0.348
	f	0.79	0.80	0.82	0.83	0.83

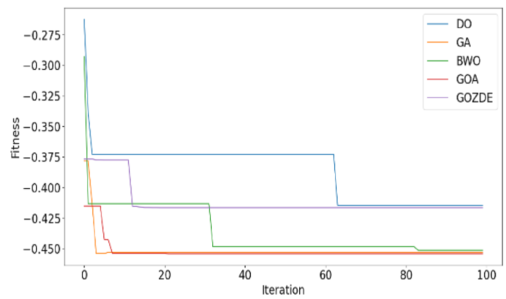
According to Table 4, the suggested GOAFS performed equally well on most datasets while reducing the number of genes per dataset while achieving high classification accuracy. According to Fig.8, different methods with different number of iterations have different fitness values for the given data. From Fig. 8, it is obvious that GOA-based feature selection achieves the highest feature reduction rate and accuracy improvement rate among the five presented methods. Overall, the proposed approach has a faster convergence time than other optimization algorithms in the feature selection problem, and it is better at balancing exploration and exploitation as well as escaping local optima. Furthermore, it is not complicated to implement, and only a few parameters need to be adjusted. Furthermore, it combines the strengths of both metaheuristic algorithms to achieve the desired result efficiently and effectively.

6 Conclusion

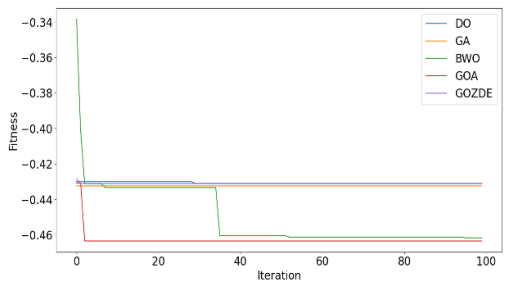
In classification tasks, feature subset selection enhances general classifier abilities, simplifies learning models, and reduces computational costs. A new wrapper-based approach to feature selection is proposed in this work. Since Gannet Optimization Algorithm (GOA) can process large-dimensional problems efficiently, it is used for selecting the feature subsets. By using the entropy of sensitivity and specificity, the fitness function achieves a balance between the True Positive Rate and the True Negative Rate. The proposed approach was evaluated using six well-known datasets. Comparisons were made with four standard feature selection methods (DO, GA, BWO, and GOZDE). In experiments, the proposed algorithm successfully selects a few prominent features while maintaining a rea-



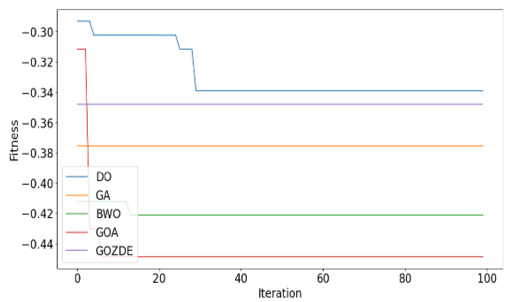
BreastEW



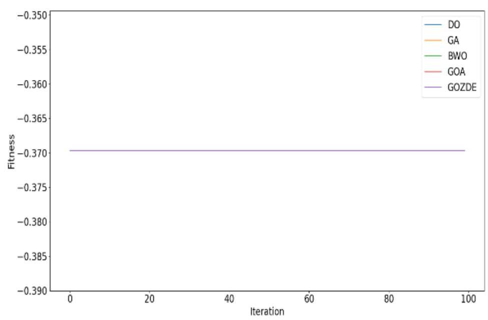
HeartEW



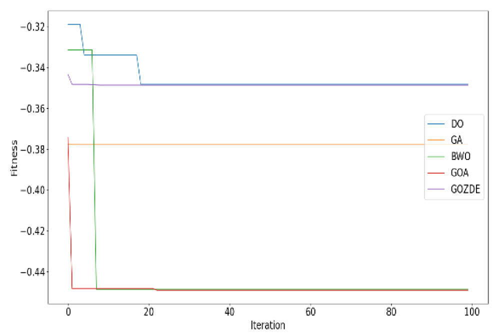
Vote



Divorce



Iris



Ionosphere

Figure 8: Performance of methods on six datasets for one hundred iterations.

sonable level of classification accuracy. A more complex data set should be tested in the future to determine the effectiveness of the proposed hybrid approach. Further research will be conducted on the effectiveness of the proposed method using different chaotic maps.

References

[1] Abd Elaziz, M., Ouadfel, S., Abd El Latif, AA. and Ibrahim, RA., Feature Selection Based on Modified Bio inspired Atomic Orbital Search Using Arithmetic Optimization and Opposite Based Learning . Cognitive Computation, (2022).

[2] Alishahi, M., Moghtadaiee, V. and Navidanc, H., Add noise to remove noise: Local differential privacy for feature selection, Computers & Security, (2022), 102934.

[3] Blum, C. and Roli, A., Metaheuristics in combinatorial optimization: overview and conceptual comparison, ACM Comput Surveys, 35, (2003), 268-308.

[4] Chakraborty, B. and Kawamura, A., A new penalty-based wrapper fitness function for feature subset selection with evolutionary algorithms, Journal of Information and Telecommunication, 2, (2018), 163–180.

[5] Dhal, P. and Azad, C., A multi-objective feature selection method using Newton's law based PSO with GWO. Applied Soft Computing, 107, (2021), 107394.

[6] Dokeroglu, T., Deniz, A. and Kiziloz, H.E., A comprehensive survey on recent metaheuristics for feature selection, Neurocomputing, 494, (2022), 269-296.

[7] Fang, L. and Liang, X., A Novel Method Based on Nonlinear Binary Grasshopper Whale Optimization Algorithm for Feature Selection. Journal of Bionic Engineering, (2022).

[8] Got, A., Moussaoui, A. and Zouache, D., Hybrid filter-wrapper feature selection using whale optimization algorithm: A multi-objective approach, Expert Systems with Applications, 183, (2021), 115312.

[9] Halim, Z., Nadeem, M., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., Ahmad, I. and Hanif, M., An effective genetic algorithm-based feature selection method for intrusion detection systems, Computers & Security, 110, (2021), 102448.

[10] Han, J., Kamber, M. and Pei, J., Data mining: concepts and techniques, (2011).

[11] Kuyu, Y.Ç. and Vatansever, F., GOZDE: A novel metaheuristic algorithm for global optimization, Future Generation Computer Systems, 136, (2022), 128-152.

[12] Machovsky-Capuska, G.E., Hunting Between the Air and the Water: The Australasian Gannet (Morus Serrator): a Thesis Presented in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Ecology at Massey University, Auckland, New Zealand (Ph.D. thesis), Notornis, 59, (2012), 66-70.

[13] Mafarja, M. and Mirjalili, S., Whale optimization approaches for wrapper feature selection, Applied Soft Computing, 62, (2018), 441-453.

[14] Manghnani, T. and Thirumaran, T., Computational CBGSA–SVM model for network based intrusion detection system, International Conference on Applications and Techniques in Information Security, (2019).

[15] Mantegna, R.N., Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes, Physical Review E, 49, (1994), 4677–4683.

[16] Naik, A.K., Kuppili, V., Edla, D.R, Efficient feature selection using one-pass generalized classifier neural network and binary bat algorithm with a novel fitness function, Soft Computing, 24, (2020), 4575-4587.

[17] Nematzadeh, H., García-Nieto, J., Navas-Delgado, I. and Aldana-Montes, J., Automatic frequency-based feature selection using discrete weighted evolution strategy, Applied Soft Computing, 130, (2022), 109699.

[18] Nguyen, B.H., Xue, B. and Zhang, M., A survey on swarm intelligence approaches to feature selection in data mining, Swarm and Evolutionary Computation, 54, (2020), 100663.

[19] Pan, J.S, Zhang, L.G., Wang, R.B., Snášel, V. and Chu, S.C., Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems, Mathematics and Computers in Simulation, 202, (2022), 343-373.

[20] Parmezan, A.R.S., Lee, H.D., Spolaôr, N. and Wu, F.C., Automatic recommendation of feature selection algorithms based on dataset characteristics, Expert Systems with Applications, 185, (2021), 115589.

[21] Saremi, S., Mirjalili, S. and Lewis, A., Grasshopper Optimisation Algorithm: Theory and application, Advances in Engineering Software, 105, (2022), 30-47.

[22] Souza, F., Premebida, C. and Araújo, R., High-order conditional mutual information maximization for dealing with high-order dependencies in feature selection, Pattern Recognition, 131, (2022), 108895.

[23] Sun, L., Si, S., Zhao, J., Xu, I., Lin, Y. and Lv, Z., Feature selection using binary monarch butterfly optimization, Applied Intelligence, (2022).

[24] Zhao, S., Zhang, T., Ma, S. and Chen, M., Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications, Engineering Applications of Artificial Intelligence, 114, (2022), 105075.

[25] Zhong, C., Li, G. and Meng, Z., Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, Knowledge-Based Systems, 251, (2022), 109215.