mathematics

MDPI

Article

A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem

Ruba Abu Khurma [1], Ibrahim Aljarah [1,*], Ahmad Sharieh [1], Mohamed Abd Elaziz [2,3,4], Robertas Damaševičius [5,*] and Tomas Krilavičius [5]

[1] King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan; ruba_abukhurma@yahoo.jo (R.A.K.); sharieh@ju.edu.jo (A.S.)
[2] Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt
[3] Artificial Intelligence Research Center (AIRC), Ajman University, Ajman P.O. Box 346, United Arab Emirates
[4] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt; abd_el_aziz_m@yahoo.com
[5] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania; tomas.krilavicius@vdu.lt
* Correspondence: i.aljarah@ju.edu.jo (I.A.); robertas.damasevicius@vdu.lt (R.D.)

Abstract: This survey is an effort to provide a research repository and a useful reference for researchers to guide them when planning to develop new Nature-inspired Algorithms tailored to solve Feature Selection problems (NIAs-FS). We identified and performed a thorough literature review in three main streams of research lines: Feature selection problem, optimization algorithms, particularly, meta-heuristic algorithms, and modifications applied to NIAs to tackle the FS problem. We provide a detailed overview of 156 different articles about NIAs modifications for tackling FS. We support our discussions by analytical views, visualized statistics, applied examples, open-source software systems, and discuss open issues related to FS and NIAs. Finally, the survey summarizes the main foundations of NIAs-FS with approximately 34 different operators investigated. The most popular operator is chaotic maps. Hybridization is the most widely used modification technique. There are three types of hybridization: Integrating NIA with another NIA, integrating NIA with a classifier, and integrating NIA with a classifier. The most widely used hybridization is the one that integrates a classifier with the NIA. Microarray and medical applications are the dominated applications where most of the NIA-FS are modified and used. Despite the popularity of the NIAs-FS, there are still many areas that need further investigation.

Keywords: feature selection; evolutionary algorithms; nature inspired algorithms; meta-heuristic optimization; computational intelligence; soft computing

check for updates

Citation: Abu Khurma, R.; Aljarah, I.; Sharieh, A.; Abd Elaziz, M.; Damaševičius, R.; Krilavičius, T. A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem. *Mathematics* **2022**, *10*, 464. https://doi.org/10.3390/math10030464

Academic Editors: Freddy Gabbay and Ripon Kumar Chakrabortty

Received: 2 December 2021
Accepted: 21 January 2022
Published: 31 January 2022

1. Introduction

As data accumulate rapidly in databases and data warehouses, a dimensionality problem becomes the main challenge for machine learning tasks (e.g., classification or clustering) [1]. Many negative effects may result from scaling up the dimensionality of a data set. These include the existence of irrelevant and redundant features that may adversely affect the learning algorithm or cause data over-fit [2]. Thus, the development of effective data mining techniques becomes an urgent necessity in various fields such as medicine [3], bioinformatics [4], text mining [5], image processing [6], design of smart infrastructures and smart homes [7], financial estimation [8,9], coastal engineering [10], and sustainability [11]. Their significance depends on their ability to turn huge amounts of data into an acceptable form. This will simplify knowledge discovery and make huge data sets more understandable, analyzable, and predictable.

Feature Selection (FS) is a pre-processing data mining technique for dimensionality reduction [12]. In recent years, research in FS has been rapidly developed in line with the era of big data and huge data sets. This subject has attracted the attention of researchers who have become more interested in developing novel FS techniques and improving current technologies [13]. FS manages the dimensionality problem by finding the most representative feature subset. The essence of FS is to choose features that are highly correlated to the class concept (relevant features) and weakly correlated with each other (complementary features/not redundant) [14]. Removing irrelevant and redundant features from a data set will cause improvements in different directions. For the modeling process, it will promote the generalization process. This will improve the quality of the generated model, so it becomes less complicated and more understandable. As a result, the inductive learner will be more efficient. FS is categorized based on the evaluation strategy into filters and wrappers [15]. The main difference between them depends on integrating a learning algorithm in the evaluation stage. Wrappers use learning algorithms to evaluate the selected feature subset. Hence, wrappers are more accurate and more expensive. In contrast, filters do not rely on learning algorithms, but use some data proprieties for evaluation. Examples of filters include univariate and multivariate filters. The main difference between them is that univariate filters rank a single feature to evaluate its performance, while multivariate filters evaluate the entire feature subset, which includes a set of feature as a combination. The generation of a feature subset in multivariate filters depends on the search strategy and the staring point of generation such as: Forward selection, backward elimination, bidirectional selection, and heuristic feature subset selection. Forward selection starts with an empty feature subset and then adds features, backward selection starts with the whole feature subset and eliminates one or more features from the set, and bidirectional search starts from both sides from an empty feature subset and from the whole feature subset at the same time [16], F-statistic [17], and information gain [18].

FS is not only a variable shrinkage process, and the target is not just to perform arbitrary cardinality reduction for a data set. FS is a multi-objective optimization problem which searches for the (near) optimal subset of features in terms of certain evaluation criteria. The main target of the FS problem is to find trade-offs between various conflicting objectives [19]. FS tries to achieve the minimum number of selected features with maximum performance [20]. Relative to search space, FS is considered a combinatorial nondeterministic polynomial-time-hard (NP-hard) problem. The reason being that it has a large search space that needs exponential running time to traverse exhaustively all the generated subsets of features [12]. The 2^N run time complexity will grow exponentially with increasing the value of N which represents the number of dimensions (features/variables) in a data set. This means that the traditional brute force methods are too impractical to be applied and other advanced search methods should be used.

Meta-heuristic search techniques are promising alternative solutions. They observed superior performance in various optimization scenarios. Potentially, they have a great opportunity to be suitable solutions for the FS problem. Meta-heuristics includes Nature Inspired Algorithms (NIAs), which are further divided into two main subcategories, namely Swarm Intelligence (SI) and Evolutionary Algorithms (EA) [21]. Both categories simulate the public behavior and biological evolution of agents in nature, respectively. Examples on EAs are: Genetic Algorithms (GA) [22] and Differential Evolution (DE) [23]. The SI category includes other types of algorithms such as Particle Swarm Optimization (PSO) [24], Ant Colony Optimization (ACO) [25], the Artificial Bee Colony (ABC) algorithm [26], memetic algorithms [27], artificial ecosystem-based optimization [28], marine predators algorithm [29], polar bear optimization [30], and red fox optimization [31].

Despite the effectiveness of nature-inspired Algorithms (NIAs) in solving the FS problem, finding the optimal solution is still not guaranteed. The main challenges that affect meta-heuristic optimization are stagnation in local minima, premature convergence, parameter tuning, exploitation and exploration imbalance, the diversity problem, dynamicity, multi-objectivity, constraints, and uncertainty [32].

Several kinds of modifications were proposed in the literature to enhance the performance of NIAs in optimization. Examples of these modification techniques include a new operator, hybridization [33], updated mechanism, new initialization strategy, new fitness function, new encoding schemes, modified population structure, multi-objectives, state flipping [34,35], and parallelism [36]. Each modification addresses the weakness of the NIA algorithm in some issues without harming the essence of the algorithm and its logic. The research field of NIA-FS has witnessed considerable development. To show the expansion of the NIAs-FS models in the literature, Figure 1 illustrates the correspondence between the year and number of publications that combine modified NIAs with FS. In the first years, research was volatile, and there were also years of research disruptions. Since 2006, the number of publications has remarkably increased to reach its peak in 2018. Furthermore, the research in this area has become very effective in the last five years. An intensive search for surveys in this area found that there are very limited NIAs-FS surveys [20]. Some FS surveys did not refer to meta-heuristics at all, but focused on other issues such as data perspectives [19], supervised/unsupervised FS approaches [15], and other FS surveys were tailored to specific applications or limited to certain domains [37]. The analysis of FS surveys showed that either they briefly refer to the meta-heuristic FS or they do not refer to them at all. To our knowledge, there is no survey about a modified NIAs-FS. This finding was one of the main motivations for this work. Unlike the previous FS surveys, FS will not be discussed in isolation from other related issues. The main objective is to bridge the gap in FS surveys by providing a review of the important aspects and design issues of NIAs-based FS approaches. The main modification strategies that have been adopted to enhance NIA for solving FS problem are categorized and discussed.

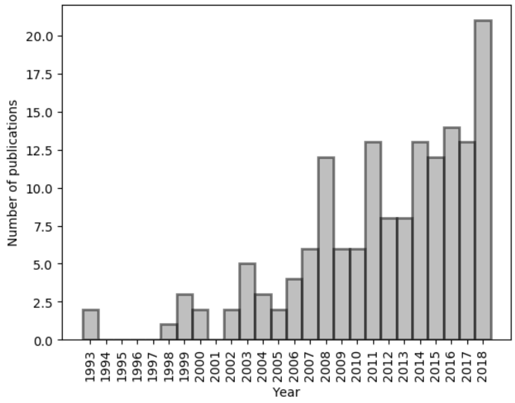


Figure 1. Development of research field regarding Nature Inspired Algorithms (NIA) modifications for tackling Feature Selection (FS).

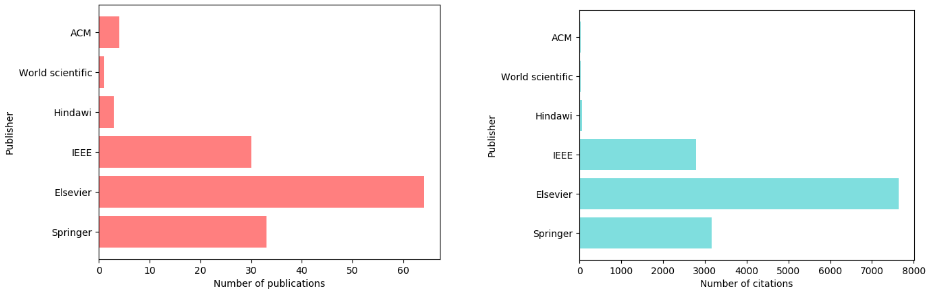
In this review, a set of research questions will be asked and answered:

1. What is the current status of modified NIAs-FS research?
2. What are the important aspects and design issues regarding building NIA for tackling FS?
3. What are the modifications that were applied on NIA for tackling FS and in what domains were they applied?
4. Are there current open-source software systems that apply a modified NIA-FS?

Based on the aforementioned research questions, we have constructed this review based on three primary issues:

- Theoretical aspects of modified NIAs-FS provide detailed coverage for three main subjects: Meta-heuristic optimization, the FS problem, and modifications on meta-heuristic to enhance meta-heuristics for FS;
- Applied aspects of modified NIAs-FS presents different applications of modified NIAs-FS;
- Technical aspects of modified NIAs-FS presents a new developed FS tool, named Evolopy-FS.

The review will refer to various well-regarded publishers such as ACM, Elsevier, Springer, IEEE, World scientific, Hindawi, and others. Figure 2a shows the number of publications for each NIA in main publishers regarding modifications for tackling FS. Figure 2b shows the number of citations for popular NIAs articles in the main publishers regarding modifications for tackling FS.



(**a**) (**b**)

Figure 2. Statistics of the number of publications and citations for papers on NIAs modifications for FS. (**a**) Statistics of publications on modified NIAs-FS; (**b**) Statistics of citations for papers about modified NIAs-FS.

A description of meta-heuristic optimization is presented in Section 3. Section 4 discusses the problem of feature space symmetry in datasets and the need for feature selection as a disentanglement of symmetry. Section 4 discusses the feature selection problem and its related issues. A review of different NIAs-FS modification techniques is presented in Section 5. Section 6 highlights the main applications on modified NIAs-FS. An assessment of NIA-FS is provided in Section 7. Finally, in Section 8, the outlook for the NIA-FS research field and possible future directions are discussed.

2. Feature Selection as a Task of Disentangling the Symmetry of Feature Space

The aim of supervised machine learning is to estimate a function f that fits well with the features of training data and allows to predict the outputs on previously unseen inputs. The number of samples required for training grows exponentially with the dimension of a feature space, which is known as the "curse of dimensionality" [38]. To approximate a Lipschitz-continuous function composed of Gaussian kernels placed in the quadrants of a d-dimensional unit hypercube (blue) with error ϵ, one requires $\mathcal{O}(1/\epsilon^d)$ samples [39].

Intuitively, a symmetry of an object is a transformation that leaves certain properties of the object invariant. For example, translation and rotation are symmetries of objects, which do not change their representations [40]. The geometric structure of the feature space imposes the structure on the class of functions f that we are trying to learn. One can have invariant functions that are unaffected by the action of the group, i.e., $f(\rho(g)x) = f(x)$ for any $g \in \mathcal{G}$ and x, here $\mathcal{G}$ is the symmetry group, g is the symmetrical transformation in the feature space, $\rho(g)$ is the group representation, and x is an input in the space of input signals $\mathcal{G}(\Omega)$ that acts as a point in the feature space. Such symmetrical transformations (e.g., translation, rotation, shifting) are commonly used for data (image) augmentation to increase the number of data instances for effective training of machine learning models.

The goal of feature selection is to eliminate uninformative and/or redundant features from the feature space, leaving only relevant (i.e., predictive) features [41]. Feature selection seeks to decrease M to M' and $M' << M$ for a dataset with N samples and M dimensions (or features). In other words, feature selection produces a disentangled representation [40] with respect to a particular decomposition of a feature space with some symmetry group, which may be useful for subsequent tasks, such as reduced complexity of training a machine learning classifier. Such disentanglement, in fact, is performed by a neural network as a part of the classification process by learning the weights of a network nodes [42], which produce asymmetric activations for separation of classes.

The redundant features are characterized by a high level of inter-correlation. Such correlated features result in the symmetrical distribution of instances in feature space.

Feature selection aims to reduce feature dimensionality by reducing the symmetry in feature space. The resulting distribution of classes in the lower-dimensional feature space should be as asymmetrical as possible to allow for easy separability of classes [43]. Furthermore, a strong correlation in features might result in numerous near-optimal feature subsets, making traditional feature selection approaches unstable and lowering the trust in selected features [44]. As many different feature space decompositions are possible, the problem of finding an optimal feature subspace in a high-dimensional feature space is known to be NP-hard [45]. In this paper, the nature-inspired meta-heuristic optimization algorithms are studied for solving the feature selection problem.

3. Meta-Heuristic Optimization

Meta-heuristic algorithms are characterized by flexibility, simplicity, low cost in computations, and they are derivation-free methods. The principle of meta-heuristics is reasonability vs completeness. In other words, it gives up completeness for providing approximated solutions for complex unsolved problems. Meta-heuristics are further categorized based on the number of candidate solutions encountered during the optimization process into the trajectory and population.

3.1. Trajectory-Based Optimization

A trajectory algorithm begins with one random solution and it tries to optimize the solution until a stop condition is satisfied. The computation overhead is reduced significantly because only one solution is being improved and evaluated during the optimization process. Equation (1) expresses the number of function evaluations needed in trajectory algorithms where T is the number of iterations:

$$\#Evaluations(in\ trajectory\ based) = 1 \times T. \tag{1}$$

Trajectory algorithms are local search techniques. They depend on making a few changes in the components of the current solution to find a better one. A potential solution is picked, and its neighboring solutions are checked if they are better. Local search implies searching within a limited region (exploitation). This process suffers from a potential entrapment in local minima because of the diversity weakness and a lack of information exchange. Examples of trajectory algorithms are Simulated Annealing (SA) [46] and Tabu search (TS) [47].

3.2. Population-Based Optimization

A population algorithm begins with a set of randomly generated solutions and tries to enhance them during the optimization process. Each candidate solution fluctuates outward or converges toward the best solution following a certain mathematical framework. The predominance of these algorithms is because of their simplicity and flexibility. Simplicity means that they are built upon simple methodologies and are evolved from simple concepts. They can be adopted to deal with real-world problems without structural modifications. All that is required is an accurate representation of the problem and the structure of the optimizer is left untouched. Population algorithms are more efficient in mitigating local minima compared with trajectory algorithms because more individuals and more information are shared between them. However, multiplicity in solutions increases the computation burden because more evaluations are required. The number of calls for a fitness function is driven by the number of individuals and the number of iterations. Equation (2) identifies the number of function evaluations in population algorithms where N is the number of individuals and T is the number of iterations:

$$\#Evaluations(in\ population\ based) = N \times T. \tag{2}$$

A population algorithm begins with the initialization step where a set of candidate solutions are generated. The solution is a candidate or possible solution if it satisfies the constraints of the problem. The next step is the evaluation of individuals. The evaluation is carried out using a specified fitness function and in terms of predefined evaluation criteria. The fitness function is called for each individual so that each individual gets a fitness value. After evaluating the individuals, the update process refines and improves current solutions. This requires updating the positions of individuals in the search space. This iterative process of evaluating and updating individuals continues until a predefined criterion is satisfied and the global optimal solution is best approximated.

Population-based algorithms compromise of NIAs that are the result of the union of nature with different scientific fields including physics, biology, mathematics, and engineering. Computer science utilized these relations between science and nature and turned it into a well-defined discipline for optimizing different challenging problems. NIAs are categorized based on the source of inspiration into EA- and SI-based algorithms [21].

3.2.1. Evolution-Based Optimization (EA)

This category includes different computational systems that share in their emulation for the biological evolution. EAs model the natural cellular processes such as reproduction, mutation, recombination, and selection.

EAs typically designed by generating a population of possible solutions $\overrightarrow{\mathrm{I}_1}, \overrightarrow{\mathrm{I}_2}, \overrightarrow{\mathrm{I}_3} \ldots \overrightarrow{\mathrm{I}_{\mathrm{n}-1}}$, $\overrightarrow{\mathrm{I}_\mathrm{n}}$ called chromosomes. Each chromosome is split into smaller units called genes. The length of the chromosome (#genes) determines the dimensionality of a problem. The relation between gene, chromosome, and population can be expressed as $gene \subset chromosome \subset population$. Most of the current evolutionary frameworks implement the chromosome and population as 1-d array (vector) and 2-d array, respectively. Equation (3) identifies the individual I_i with a length d and Equation (4) identifies a population P where each individual represents a row in a matrix. Each solution is evaluated by a certain object function $\overrightarrow{\mathrm{O}_1}, \overrightarrow{\mathrm{O}_2}, \overrightarrow{\mathrm{O}_3} \ldots \overrightarrow{\mathrm{O}_{\mathrm{n}-1}}$, $\overrightarrow{\mathrm{O}_\mathrm{n}}$ to determine its quality and decide if it is fitted or unfitted. The highest evaluated solution (best individual) is preserved at each iteration. The unfitted solutions (worst individuals) are candidates to be replaced by newly-generated offsprings. This allows the average fitness value to increase dramatically throughout iterations. Common EA examples are GA [22] and DE [23]. GA undoubtedly is the most widespread and typical example of EAs:

$$I_i = \begin{bmatrix} x_i^1 & x_i^2 & x_i^3 & \cdots & x_i^{d-1} & x_i^d \end{bmatrix} \tag{3}$$

$$P = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & \cdots & x_1^{d-1} & x_1^d \\ x_2^1 & x_2^2 & x_2^3 & \cdots & x_2^{d-1} & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^1 & x_n^2 & x_n^3 & \cdots & x_n^{d-1} & x_n^d \end{bmatrix}. \tag{4}$$

3.2.2. Swarm-Based Optimization (SI)

SI algorithms have a common behavior that is very similar to the social behavior of creators. The Swarm system comprises an abundant number of agents that are distributed in the environment to achieve a global target. Intelligence can be seen in the actions of agents to coexist. The main characteristics of swarm systems are adaptability, self-organization, distributed control, scalability, and flexibility [20]. The most common SI examples are Particle Swarm Optimization (PSO) [24] and Ant Colony Optimization (ACO) [25]. A PSO source of inspiration are flocks of birds that search for food. The search procedure is guided by two main factors: Pbest and gbest. Pbest represents the best experience that was gained by the previous particle itself. Gbest represents the best individual in the whole swarm. Particles also have a position and velocity that are both updated in each iteration.

3.3. Challenges of Meta-Heuristic Optimization

Despite the efficiency of meta-heuristics in tackling challenging optimization problems, some obstacles impact their performance. These include dynamicity, multi-objectivity, constraint, and uncertainty. For multi-objectivity, there are multiple conflicting objectives to be optimized until trade-offs (Pareto optimal set) are achieved. The search space is quite more complex. The optimization problem becomes highly challenging when the number of objectives becomes larger than four [48]. Many objective fields have emerged to deal with these cases. Constraints of real problems create gaps in the search space by dividing it into feasible and infeasible regions. Feasible regions satisfy the constraints while infeasible regions violate these constraints [49]. Accordingly, the optimization algorithm should follow certain mechanisms to become closer to the promising region and avoid the infeasible region until an optimal solution is found. The other main issue of meta-heuristic optimization is uncertainties. For example, the global solution frequently changes its position in the search space, which requires more attention from the optimization algorithm. Some operators are used for registering the history and memorizing the locations of the global optima all the time. Other severe challenges are related to the problem search space, such as the existence of many holes or valleys that lead to stagnation in local minima, discontinuities in a search space, the location of global optima that comes onto the boundary of a search space (the boundary of constraints), and the isolation of global optima [32].

Population algorithms are characterized by two conflicting milestones that are called exploration (diversification) and exploitation (intensification) [32]. In exploration, the candidate solutions churn and change violently, which leads one to examine more regions and to find diverse solutions. Exploitation changes gently and causes a less sudden stir for the candidate solutions. GA realizes these processes through crossover and mutation operators. Crossover intermixes a combination of solutions while mutation squeezes certain regions and searches locally. PSO configures the inertia weight operator by large values for more explorations and selects small values for more exploitation.

The main challenges of exploration and exploitation include: Firstly, since they have conflicting purposes, increasing any of the causes decreasing the other. Secondly, a transition between these two milestones is not defined because the search spaces of the optimization problems are usually unknown. Thirdly, performing pure exploration causes less accuracy in approximating an optimal solution because different regions are being explored without a focus on a certain promising region. Performing pure exploitation gives rise to entrap in local optima. Fulfilling a balance between exploration and exploitation may produce better results and increases the chance of being close to the optimal solution. Recently, this idea has become an active research problem. Several types of research have tried to attain balance by integrating several random and adaptive operators in the structure of the algorithms.

4. Feature Selection

This section introduces FS in two parts: The dimensionality problem and the FS system based on the NIA search strategies.

4.1. Dimensionality Problem

Due to the incremental growth of information and the abundance of data, data sets have increased in both data samples (number of instances) and dimensions (number of features). As a result of the increased dimensionality, different negative effects were embedded in data mining tasks. One of these problems is called the curse of dimensionality, which describes the status of data as it becomes sparser in large dimensionality space [12]. This raises the need for more instances for the training of the classifier, which increases the learning time. Learning algorithms were designed to build their models based on rules inferred from a small number of dimensions. Learning algorithms cannot generalize well in a large dimensionality space. High dimensionality implies the existence of noisy features, such as redundant and irrelevant features that mask the informative features and mislead the classifier and cause data to overfit. An overfitting [2] problem occurs when a classifier overtrained on the data and learned all examples, including outliers. Considering that noise and random fluctuations as related concepts will cause building complex models;

logically, learning from relevant features allows the classifier to be more accurate. Another negative effect of increasing dimensions is the increased demand for specialized devices such as large memory storage and high-speed processors, which increases cost.

4.2. FS Preliminaries

Features are defined as measurable properties of the observation under study. The complexity of the problem is determined by its features. In real-world applications, the discovery of relevant features is a big challenge. In 1997, the first papers about relevance and feature selection were published [14]. Feature relevance can be formalized as follows. Let $1 \leq i \leq n$, E_i be the domain of feature x_i, $X = x_1, x_2, \cdots x_n$ be the set of all features. $E = E_1 \times E_2 \cdots \times E_n$ is the instance space from which instances derive their values. Each instance can be represented as a point in space and the distribution of these data points has a probability P. If we consider the class (label) space to be T, then we can define an objective function c as a relation that maps an instance S to a specified label/class in labels space T as: c: $E \rightarrow T$. Arguably, a data set with $|S|$ number of instances is the result of sampling $|S|$ times from E with a probability P and get label from T. An x_i in X is a relevant feature with respect to class concept if there exist two instances (A and B) in E, which only differ in their assignment to xi (all their feature values are the same except those for feature xi) and $c(A) \neq c(B)$. In contrast, a variable with no correlation or weak correlation with the target concept is called an irrelevant feature. Other types of noisy features are redundant features. These are features that are highly related and connected with other features and add nothing new regarding the classification decision.

In the literature, FS was defined in different ways, which are all close in meaning and intuition [15]. FS is a searching process that tries to find the subset of features which is the best one to describe the data. According to relevance discovery, FS aims to determine the most meaningful subset of features, which has the largest relevance and minimum redundancy. Even though those features are fewer than the original features, but they carry the maximum discriminate information. Classically, FS selects a subset of M features from a set of N features where $M < N$ and the value of an evaluation function is optimized over all subsets of size M. The essence of FS is to select or discard features intelligibly in such a way, the resulting class distribution is as close to the class distribution with the complete set of features. In another meaning, FS is not a technique for only reducing data set cardinality, but it should find a trade-off and a balance between different conflicting objectives. As a multi-objective optimization problem, there are two primary objectives to be optimized. These objectives are the performance and the number of selected features. These are conflicting objectives because the optimization algorithms require getting the maximum performance and the minimum number of selected features.

Typically, the standard process of FS consists of four primary stages of subset generation, subset evaluation, stopping criterion, and results validation [15].

Regarding the subset generation and search procedure, FS is considered an NP-hard problem. When the number of features equals n, the search space comprises 2^n subsets of features. Using brute search methods such as a huge search space needs an exponential running time to traverse all the candidate subsets of features.

Concerning subset evaluation, there are different methods to assess the goodness of a feature subset such as filters and wrappers. A stop criterion is a condition that halts the FS process and prevents the infinite loop. For example, the search completion (all feature subsets have been examined), the learning performance reached its highest limit, the subset of features with a specified size is obtained, the pre-defined number of iterations is reached, the occurrence of conversion situation in which results become stable, and no further enhancement is achieved. A direct way to validate the obtained results is based on prior knowledge from a domain. Unfortunately, this features knowledge is usually unavailable so other methods have to be used instead. FS could be validated by comparing the system performance using the whole subset of features with its performance using the selected features. FS has many advantages that positively affect the data mining task, including improving the quality of the generated model, speeding the learning time of the classifier, enhancing the ease of reading the data set, and reducing the need for more hardware resources.

4.3. NIAs for Feature Selection

Two important points should be focused on: The representation of a solution and the evaluation for it. Normally, a feature subset is represented by a binary vector. The dimensionality of the problem is equal to the number of features in the data set. If the gene value is set to 1, this indicates that the feature is selected, otherwise, it is not selected. The quality of a feature subset is evaluated based on two contradictory objectives: The classification accuracy (minimum error rate) and the minimal number of selected features simultaneously. These two criteria are represented in one fitness function that is shown in Equation (5), where $\alpha\gamma_R(D)$ is the error rate of the classification produced by a classifier, $|R|$ is the number of selected features in the reduced data set, and $|C|$ is the number of features in the original data set, and $\alpha \in [0,1]$, $\beta = (1-\alpha)$ are two parameters for representing the significance of classification and length of feature subset according to recommendations:

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|}. \tag{5}$$

5. NIAs FS Modifications

This section highlights the main modification techniques applied in the literature to enhance the NIAs as wrappers FS. By referring to 156 articles in the domain of modified NIAs-FS, it can be noticed that the modification techniques can be classified into nine categories as depicted in Figure 3: New operators, hybridization, update mechanism, modified population structure, different encoding scheme, new initialization, new fitness function, multi-objective, and parallelism.

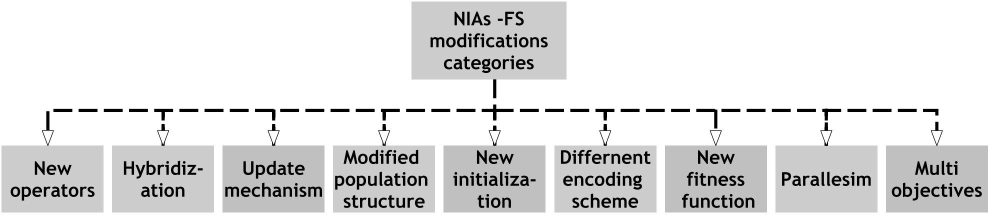


Figure 3. NIAs FS modifications categories

5.1. New Operators

This modification depends on integrating a new operator in the original NIA structure to achieve certain targets, such as improving the algorithm performance, increasing the diversity among the population, enhancing the exploitation and exploration processes, facilitating the sharing of information between population's individuals, repositioning of the worst individuals in the population, and performing a search along various vectors in search space [36]. In literature, several operators have been used to enhance NIAs wrappers. Some of these operators are discussed next.

5.1.1. Chaotic Maps

The denotation of chaos means a state of disorder. In mathematics, it is a formula that describes a dynamic system with time dependence. The chaotic system has a high level sensitivity to its initial conditions. This behavior implies that even a simple modification in the initial conditions will lead to big changes in the outcomes. Although the chaotic system is deterministic and does not incorporate any randomness but the results are not always predictable [50].

Chuang in [51] used two kinds of chaotic maps and integrated them with Binary Particle Optimization (BPSO), namely logistic maps and tent maps. Equation (6) describes how the logistic map is written in mathematics (general formula), where X_n is a number between 0 and 1 which represents the ratio of the current population size to the maximum population size and μ is a constant value between 0 and 4. Equation (7) describes how Chuang exploited Equation (6) to modify the inertia weight value where w is the inertia value between (0,1) and t is the number of iteration. The same thing was followed to apply the tent map chaotic map. Equation (8) is the general mathematical formula and Equation (9) is the modified version of inertia weight using a tent map. Using large values for inertia weight facilitates more exploration while selecting small values facilitates more exploitation. Hence, chaos theory could be used for balancing the two types of search in the search space. Besides, the study contributed that Chaos Binary Particle Swarm Optimization (CBPSO) with a tent map achieved a higher classification accuracy than CBPSO with a logistic map:

$$X_{n+1} = \mu\ X_n(1 - X_n) \tag{6}$$

$$w(t+1) = 4.0 * w(t)(1 - w(t)) \tag{7}$$

$$f(x) = X_{n+1} = \begin{cases} \mu\ X_n, & \text{if } X_n < 0.5 \\ 1-\ \mu\ X_n, & \text{otherwise} \end{cases} \tag{8}$$

$$w(t+1) = \begin{cases} w(t)/0.7, & \text{if } w(t) < 0.7 \\ 10/3w(t)(1 - w(t)), & \text{otherwise.} \end{cases} \tag{9}$$

In the same year, Chuang presented another model for FS [52]. The proposed model was a filter-wrapper approach based on using a correlation-based filter (CFS) and Taguchi chaotic BPSO (TCBPSO). In [53], chaotic was applied with BPSO for FS in text clustering. Ahmad in [54] used chaotic maps as modifications for the SSA algorithm. He replaced the C3 random parameter with chaotic sequences, namely logistic map, piecewise map, and tent map. It was clear the impact of chaotic maps in improving the SSA. In the same year, the influence of chaotic operators on SSA was investigated in [55]. The experiments proved that the logistic map achieved a better performance for the SSA algorithm over nine chaotic maps. The chaotic multiverse optimization (MVO) FS model was proposed in [56] to cope with some limitations of MVO. Tent, logistic, singer, sinusoidal, and piecewise chaotic maps were used. The results showed that the logistic chaotic maps were the best, which increased the MVO performance more than other maps. Sayed in [57] developed a new wrapper FS approach based on the Whale Optimization Algorithm (WOA) and chaotic theory named CWOA. He used 10 chaotic maps namely chebyshev, circle, guass/mouse, iterative, logistic, piecewise, sine, singer, sinusoidal, and tent. The results showed that a circle chaotic maps was the best among other chaotic. In [3], a model based on chaotic Moth Flame Optimization (CMFO) and Kernel Extreme Learning Machine (KELM) was proposed. In [58], Sayed developed a new FS system composed of the Crow Search Algorithm (CSA) algorithm and chaos theory to enhance the performance and convergence speed of CSA. Lately in [59], a Binary Black Hole optimization Algorithm (BBHA) has been modified by embedding new chaotic maps embedded with the movement of stars in the BBHA. This model was called CBBA and uses 10 chaotic maps. The results of three chemical data sets demonstrated that CBBA outperformed the BBHA in terms of the number of selected features, classification performance, and computational time.

5.1.2. Rough Set

Rough Set (RS) was first described by Zdzislaw Pawlak at the beginning of the 1980s [60]. This is a mathematical concept related to topological operations. In mathematics, RS is a theory that tries to find two approximate sets for the original conventional set (crisp set). The first RS gives the lower approximation for the crisp set which compromises the elements that surely belong to the target subset. The second RS gives the upper approximation of the crisp set which compromises the elements that possibly belong to the target subset. The pair of rough sets are themselves either crisp sets or fuzzy sets. Rather than belonging or not belonging in relation to the elements as in crisp sets, the fuzzy sets depend on the membership function for gradual assessments of the elements. Unlike the fuzzy sets, RSs depend on finding the positive region, not the membership function for dealing with uncertainties and vagueness. The RS has many advantages, including the approximation of concepts, reduction of spaces, discovering the equivalence relations, and finding the minimal sets of data in vague and uncertain domains. In FS, the RS tries to define the attribute dependency. Zainal in [61] proposed the RS-PSO model for a better representation of data. Another RS-PSO-FS model was proposed in [62] based on Relative Reduct (PSO-RR) and PSO-based Quick Reduct (PSO-QR). Both tools depend on the dependency measure for comparing sets of attributes. In [63], the authors proposed a model for FS in nominal data sets based on BCS and Rough Sets. Another CS model was introduced in [64] by incorporating the RS with different classifiers. In [65], a new model was developed based on two incremental techniques (QuickReduct and CEBARKCC). Quick reduct and CEBARKCC are two filtering methods where the former one is a rough set-based filter that simulates the forward generation method and the latter is a conditional entropy-based method. These two methods were integrated with the Ant Lion Optimization (ALO) algorithm to improve the initial population quality. The RS-FA model was developed in [66]. Hassanien in [67] developed a new system based on rough set and MFO. Lately, in [68], a hybrid model called BPSOFPA composed of Flower Pollination Algorithm (FPA) and PSO was also developed. BPSOFPA was integrated with the RS approach for the FS problem. Ropiak in [69] integrate RSs with deep learning as rough mereological granular computing.

5.1.3. Selection Operators

Inspired from Darwin's theory [70], which explained the evolution and changes in species through the natural selection mechanism, the genetic algorithm incorporated selection operators to select some individuals from the population for later breeding. A conventional strategy to implement the selection is using the fitness values of the solutions. In other methods, these fitness values are normalized by finding the summation of them then divide the fitness of each individual by this summation. Another method sorts all individuals in the population according to their fitness values in descending order. The selection mechanism was applied in other studies by finding the accumulated fitness for each individual so that the final individual fitness value is one [71]. All such methods become computationally expensive and may negatively impact the performance of GA when the population becomes larger. Other methods of selection which are widely implemented with GA are Tournament Selection (TS) and Roulette Wheel Selection (RWS). The stochastic nature of these methods makes them simpler in implementation and better in performance than the aforementioned methods. TS is the most applied selection operator with GA because of its simplicity. It selects randomly a set of solutions from the population then the best one is used for breeding the successive generation. In RWS, the mechanism differs in that no agent in the population is discarded. The RWS strategy depends on creating something like a roulette where all fitness scores of the individuals are represented as areas or sectors on this roulette. The individual with a large fitness value well reserve a large sector on the roulette, which shows a larger probability for selection. Individuals with small fitness scores will reserve small areas on the roulette. In RWS, the final selection for the agent is done by rotating the roulette and the selected individual is the one where the point stayed when the roulette had stopped. Mafarja in [46] developed a new model that combines TS with the WOA optimizer to enhance the exploration of the search. One year later, Mafarja presented in [72] an FS model based on the Grasshopper Optimisation Algorithm (GOA) algorithm with RWS and TS. Mafarja in the same year developed a new wrapper FS model based on WOA along with studying the effect of TS and RWS [73]. In [26], the selection operators were incorporated to improve the ABC optimizer. In [74], the method compromised of a DE optimizer and RWS structure for the selection of the Wavelet Packet Transform.

5.1.4. Sigmoidal Function

A sigmoid function is a mathematical function that falls under the S-shaped family and is considered a special case of a more general function called a logistic function, which has the mathematical formula defined by Equation (10), where e is the natural logarithm base (Euler's number), $x0$ is the sigmoid's midpoint, L is the sigmoid's maximum value, and k is the logistic growth rate of the curve [75]. The sigmoid function formula is defined by Equation (11):

$$f(x) = L/(1 + \exp^{-k(x-x0)}) \tag{10}$$

$$S(x) = 1/(1 + \exp(-x)) = \exp(x)/(\exp(x) + 1). \tag{11}$$

The sigmoid function has some special characteristics including the monotonic behavior, which means that the function is defined on all real numbers but the output of the function is increasing either from 0 to 1 or from -1 to 1. Moreover, the sigmoid function is differentiable and has a bell-shaped first derivative where the derivative at each point is a non negative value. There are several variations of the sigmoid function such as hyperbolic tangent, arctangent function, and algebraic functions which are respectively defined by Equations (12)–(14). The sigmoid function is widely applied as the activation function of a Neural Network (NN). Other useful usage of the sigmoid function is that it is used as a discretization method to convert a continuous space into a binary one, such an application is a feature selection application:

$$f(x) = tanh(x) = (e^x - e^{-x})/(e^x + e^{-x}) \tag{12}$$

$$f(x) = arctan(x) \tag{13}$$

$$f(x) = x/\sqrt{(1 + x^2)}. \tag{14}$$

For solving the FS problem, Aneesh developed a modified BPSO called Accelerated BPSO (ABPSO). The strategy for accelerating the particles was using a new velocity update function based on a sigmoidal function [76]. In [6], the sigmoidal function was used with BGWO in solving FS. In [77], different transfer functions that map continuous solutions to binary ones were applied in combination with the CS algorithm. The CS-sigmoid and CS-hyperbolic tangent was performed on five data sets. In [78], the effect of different transfer functions on the Bat optimization (BA) algorithm was studied. Sigmoid and hyperbolic tangent functions were used to analyze their influence on FS. The results proved that the sigmoid function was better than the hyperbolic function in feature reduction for almost all data sets. Mafarja, in [79], presented new versions of the Grasshopper Optimization Algorithm (GOA) based on sigmoid and V-shaped TFs in the context of FS.

5.1.5. Transfer Functions

Transfer functions (TFs) are mathematical formulas that play a significant role in mapping a continuous search space to discrete search space. The discrete search space could be viewed as a hyper-cube in which solutions move in different directions within its boundaries by flipping their bit values. TFs are one of the most efficient ways that could be utilized to covert continuous meta-heuristic algorithms into their corresponding binary versions [80]. The mathematical formulations of these TFs can be found in [80]. The update procedure in a binary meta-heuristic algorithm is switching solutions elements between 0 and 1 based on certain mapping formula TFs that links the original continuous update procedure with a new binary update procedure. TFs in a close meaning define the probability of updating each element (gene/feature) in a solution to be either selected 1 or not selected 0.

Equations (15) and (16) define the general update formulas of a solution using S-TFs and V-TFs, respectively, where $X_i^d(t+1)$ represents the *ith* element (gene/feature value) in the X solution (feature subset) at dimension d (feature number/index) in iteration $t+1$, rand $\in [0,1]$, which was generated using a random probability distribution:

$$X_i^d(t+1) = \begin{cases} 0, & \text{if } rand < S\_TF(X_i^d(t+1)) \\ 1, & \text{if } rand \geqslant S\_TF(X_i^d(t+1)) \end{cases} \tag{15}$$

$$X_{t+1} = \begin{cases} X_t, & \text{if } rand < V\_TF(X_{t+1}) \\ \neg X_t, & \text{if } rand \geqslant V\_TF(X_{t+1}). \end{cases} \tag{16}$$

These can be reformulated to preserve the concepts of searching using any specific meta-heuristic algorithm. As an example, PSO was converted by Kennedy and Eberhart [81] from a real algorithm to a binary algorithm. The PSO binary conversion started by employing a sigmoid function to convert the velocity values into probability values bounded in the interval [0,1] as in Equation (17), where $T(v_i^d(t))$ indicates the velocity of particle i at dimension d in iteration t. In the next step, the computed probabilities are used to update the position vector using Equation (18). To preserve the PSO continuous searching method and keep the concepts of pbest/gbest, the TF gives a high probability for switching gene values for those genes having high-velocity values since they are far away from the best solution. Small probability is given for genes having small velocity values since they are considered close to the best solution [80]:

$$T(v_i^d(t)) = 1/(1+e^{-v_i^d(t)}) \tag{17}$$

$$X_i^d(t+1) = \begin{cases} 0, & \text{if } rand < TF(v_i^d(t+1)) \\ 1, & \text{if } rand \geqslant TF(v_i^d(t+1)). \end{cases} \tag{18}$$

In the literature, there were several studies that adopted TFs operators with FS problem. Mirjalili in [80] improved the performance of BPSO by using TFs, S-shaped, and V-shaped transfer. The results of V-TFs improved the performance of BPSO more than S-TFs. In [82], a new wrapper was developed by modifying the Salp Swarm Algorithm (SSA) using TFs. The proposed approach achieved significant superiority over other competitive approaches in 90% of the data sets. Mafarja in [83] presented a new wrapper FS method based on a modified Dragonfly Algorithm (DA) using time-varying S-shaped and V-shaped TFs. Recently, in the context of Internet of Things (IoT) attack, a new wrapper-based approach using the WOA was developed. The augmented WOA used both V-shaped and S-shaped transfer functions.

5.1.6. Crossover

In living things, the chromosomal crossover is a recombination process that occurs between non-sister chromatids to exchange the genetic material during recombination (sexual reproduction). This process ends in the production of new recombinant chromosomes. Faraway from the biological chromosomal crossover in the genetic algorithm and evolutionary computation, this process was inspired to exchange information between solutions in the population and generating new offsprings in the next generation. In the genetic algorithm, recombination (crossover) is defined as a stochastic operator that enforces the diversity in the population by exchanging (swapping) the bits after a random cutting point (crossover point) between the parents' vectors (selected individuals) to produce new children (offsprings). Equation (19) shows how a crossover operator is used to combine solutions where $\bowtie$ is an operator that performs the crossover scheme on the two binary solutions X_i and X_{i-1}. In a binary space, the crossover can be realized by exchanging the binary bits of two solutions to obtain an intermediate solution. Equation (20) shows that the crossover mechanism switches between two input vector with the same probability, where Xd is the value of the dth dimension in the yielded vector after applying the crossover operator on X_i and X_{i-1}:

$$X_i^{t+1} = \bowtie(X_i, X_{i-1}) \tag{19}$$

$$X^d = \begin{cases} X_1^d, & \text{if } rand \geqslant .5 \\ X_2^d, & otherwise. \end{cases} \tag{20}$$

In [84], a crossover operator was applied in combination with the sigmoid function to modify a Binary Grey Wolf Optimizer (BGWO). The BGWO1 approach was used to convert the Continuous version of GWO (CGWO) into the binary version. The first steps toward the three best solutions are converted into binary, then a random crossover is applied among them to find the updated position. The results of the approach positively affected the performance of GWO. In [82], the crossover operator was applied to improve the Salp Swarm Algorithm (SSA) optimizer in solving the FS problem. The crossover job was to increase the diversity of the model and improve the exploration process of the search space. In [73], the study incorporated many modifications strategies with WOA. Solving the limitations of the WOA represented by local minima and slow convergence was the priority. The crossover was used for achieving this target. Mafarja in [79] applied multiple operators with GOA. The combination operator together with the mutation was applied in his approach to BGOA-M for achieving more exploration.

5.1.7. Mutation

In the organism, the mutation is an error that occurs during DNA replication (meiosis). The error specifically results from a permanent deletion, insertion, or alternation on the DNA segment (nucleotide sequence of the genome). Even though this is a small genome error, it causes abnormal changes in the characteristics of an organism. Evolutionary and genetic algorithms inspired the same idea to make changes and increase the diversity in the population. The advantages of mutation come from preventing solutions becoming similar and thus ensuring the evolution does not stop. Mutation operators alter one or more gene values (a bit in chromosome vector) which causes the solution to be changed from its previous state. Besides diversity, the mutation could contribute to mitigating the local minima problem. Equation (21) identifies the mutation process where $Xi(t+1)^d$ is the ith element at the dth dimension in the X_i solution,

$$X_i^d(t+1) = \begin{cases} 0, & \text{if } rand \geqslant .5 \\ 1, & otherwise. \end{cases} \tag{21}$$

In [85], a Particle Swarm Optimization (PSO) applied mutation to a solution was conducted after it was updated. A probability commonly $1/n$ indicates one bit of the solution will be muted (flipped). The model proved the effectiveness of the suggested modified PSO-FS model. In [53], the authors developed a hybrid intelligent algorithm that combined mutation with the BPSO and other operators to solve FS in the text clustering. The model attained a higher clustering accuracy and improved the convergence speed of BPSO. In [79], the mutation operator was applied with the GOA optimizer. The BGOA-M approach achieved superiority in comparison with other approaches compared. In [86], an Improved Harris Hawks Optimization (IHHO) was proposed based on elite opposite-based learning, mutation neighborhood search, and rollback strategies to increase the search performance.

5.1.8. Levy Flight

Levy flight has its source from chaos theory. It describes a random walk that follows a heavy-tailed probability distribution. This probability distribution represents the step-lengths that take place either on a discrete grid or continuous space. In mathematics, according to a central limit theorem, the steps from the original point of a random walk follow a stable distribution which could be modeled using equations of Levy flights. Investigators in nature found that Levy flights can describe the animals hunting patterns especially when the prey is sparsely distributed and not easily detected as opposed to Brownian motion, which can only approximate the prey place when the hunting is near an abundant and predictable prey [87]. In [64], a novel Cuckoo Search (CS) algorithm was developed using the Levy flight with the rough sets. He applied his idea by integrating the Levy flight random probability distribution in the equation that generates new solutions as shown in Equation (22) where $\oplus$ denotes the entry-wise multiplication, α is the step size, $\alpha > 0$, and $Levy(\lambda)$ is the Lévy distribution which is described in Equation (23). In [88], Levy flight was used in combination with transfer functions to enhance the performance of the MFO algorithm and increase diversity:

$$X_i^{t+1} = X_i^t + \alpha \oplus Levy(\lambda) \tag{22}$$

$$Levy \sim u = t^{-\lambda} \qquad 0 < \lambda < 3. \tag{23}$$

5.1.9. Other Operators

A local search operator was incorporated with GA to mitigate the weakness of standard GA in fine-tuning near the local minima [89]. In [90], local search was used to improve the BPSO. A new local search and gbest resetting strategy called PSO-LSRG was proposed in [24] to facilitate the exploitation. A Uniform Combination (UC) operator was used in [80] to improve the performance of BPSO. Later, UC was adopted in [91] to balance the exploitation and exploration of bones PSO. The DE evolutionary operator was used in [5] to solve the local optima in standard WOA. The DE evolutionary includes mutation, crossover, and selection operators. Boolean algebra (and operator) was used in BPSO [92]. The bacterial evolutionary algorithm and PSO algorithm, both with a plain and a memetic variant complemented with gradient-based local search and fuzzy logic numbers were used in [93] for solving various resource allocation problems.

A catfish strategy was applied in [94] to improve the performance of BPSO based on introducing new particles into the search space when there is no improvement in the searching process. For example, when the gbest is unchanged over a consecutive number of iterations. The catfish particles replace the particles with the worst fitness and initialize a new search from the extreme positions of the search space. Feature subset ranking was introduced in [95]. The idea was to compute the significance of each feature according to its classification accuracy and compute the accuracy for some combinations of these ranks, then the BPSO wrapper approach was used to search on the top-ranked features subsets instead of the whole features.

A Gaussian operator was introduced in [96] and the idea was that FS is highly influenced by features interaction. The highly relevant features with a class label may have high interactions with other features which makes them redundant. On the other hand, irrelevant features concerning a class label may have small interactions with other features. As feature interaction is a challenge to classification and FS, a statistical clustering method based on Gaussian distribution was adopted. It groups homogeneous features based on the interactions between features then the PSO algorithm selects one feature from each cluster. Threshold was adopted in [97]. The idea was to set a nonzero value for a threshold based on the number of trails BPSO were run. The significance of a particular dimension is measured based on the frequency of appearance for that dimension in the gbest vector in all runs. The final gbest after thresholding will contain the most recurrent features.

Zhang in [91], used the Gaussian sampling to compute the positions of particles which is based on pbest and gbest instead of velocity. Another operator was incorporated, called reinforced memory. Reinforced memory is based on the idea of enhancing the probability of survival for outstanding genes. These are the important features with high fitness value in the current iteration. Consequently, the update of the local leaders (pbest) of each particle will avoid the gene degradation and preserve it in the next iteration. Hamming distance was used in [98] to replace the Euclidean distance in BPSO. Particularly, it was used to measure the distance between two binary vectors based on the Exclusively-OR (XOR) operator and count the number of ones in the resulting vector. In [99], a new model called Hybrid Particle Swarm Optimization Local Search (HPSO-LS) was proposed based on using local search with correlation information. The correlation information was used to guide the local search in PSO. This was carried out by including the most dissimilar features (low correlated) as a feature subset in the newly generated particles. Consequently, similar features (highly correlated) have less chance to be selected as a feature subset. Moreover, HPSO-LS used a specific subset size determination scheme to allow PSO to search within the abounded region and find a smaller number of features.

Binary quantum was used in [100] to modify and improve the PSO. The idea was to perform a sampling around the personal best and compute the mean best of the sampled points then introduce this value in the BQPSO. For any bit position of the mean best, it will be equal to 1 if 1 appears more often than 0 in all the corresponding bit positions of all pbests. On the other hand, if the 1 and 0 have the same frequencies, then each element of the mbest is set randomly either to 0 or 1. A re-initialization strategy was applied on PSO-mGA in [101]. The idea was to use a small population (3–6 chromosomes) with a reinitialization strategy to achieve convergence. A non replaceable memory operator was added to keep the original swarm and remains intact with it during the optimization process. This will help in increasing the diversity of a swarm. Moreover, the nonreplaceable memory was used for maintaining a secondary swarm with a leader and followers. Zhang in [102] developed a new wrapper-based approach by utilizing the Firefly Algorithm (FA), Return-cost, Pareto dominance-based, and adaptive movement operator. A return-cost indicator was used to compute attractiveness. The firefly is cloned based on the return cost instead of the distance so that the firefly with a big return and small cost has a great chance to be cloned. A pareto dominance-based operator was added. Pareto dominance is commonly used in multi-objective optimization. It is a selection strategy used to search for the attractive one of a firefly based on the cost and return. Adaptive jump was used in place of the fixed uniform jump. It requires a change in the jump probability based on a linear function concerning the number of iterations to allow for more exploration.

In [103], a greedy search was used to enhance the local search. Three modified versions of the Lion Algorithm (LA) (Lion M1, Lion M2, and Lion M1+M2) were proposed to improve the local search. Mafarja, in [72], applied a new methodology based on BGOA and Evolutionary Population Dynamics operator (EPD). EPD depends on making a local change in the population instead of external force. This idea comes from the theory of Self-organized Criticality (SOC). Hancer, in [26], developed a new version of the DisABC algorithm for FS by introducing a DE-based neighborhood mechanism into the similarity-based search of DisABC. DE evolutionary operators were also used in [5] for solving the problem of local optima in native WOA. These include mutation, crossover, and selection operators. Khushaba in [74] developed a new modified FS method called DEFS using a repair mechanism. The repair mechanism was based on feature distribution measures and the RWS structure. A new model was developed in [104] based on GA and m-features (OR operator). The OR operator performed a search space reduction and improved GA performance and convergence. Zeng in [105] developed a novel GA with a new population structure and a new operator called dynamic neighboring. Dynamic neighboring is a new selection strategy that was used to boost the capabilities of GA for the FS problem. In [106], Guo proposed a new repair operator that allowed GA to transform feature subsets from arbitrary combinations to valid combinations that conform to the feature model constraints and domain-specific objective function.

5.2. Hybridization

Hybridization means the integration of over one algorithm to build a powerful predictive framework that combines the power of the integrated algorithms. The expectation of combining the complementary features of different optimization strategies is to achieve a better performance compared with implementing them separately as pure paradigms. There are several categories of NIAs hybridization techniques that were investigated in the literature such as combining NIA with other NIA or combining NIA with other algorithmic components from different areas of optimizations, such as with tree search, dynamic programming, and constraint programming [107].

5.2.1. NIA-NIA Hybridization

In mimetic models, a single solution algorithm is embedded in the population's structure algorithm to enhance the local search and exploitation of the search space. These algorithms are implemented in two search stages. In the first stage, the algorithm captures a global view of the search space. In the second stage, the algorithm focuses on the most promising area to perform a successive process of local search. As exploration/exploitation balance is guaranteed using these models and the premature conversion is avoided. In [4], Zawbaa developed a novel hybrid GWO-ALO system that exploits the GWO global search ability and Ant Lion Optimization algorithm (ALO) local search performance. In [65], Mafarja developed a hybrid model based on BALO and hill-climbing techniques called HBALO. A new hybrid algorithm was presented in [108] by combining the Clonal Selection Algorithm (CSA) with the Flower Pollination Algorithm (FPA). CSA was good in exploitation, while FPA was good in exploration via Levy flight. In [109], the Mine Blast Algorithm (MBA) was used to support the exploration phase. MBA was integrated with simulated annealing to optimize a local search in the exploitation phase to get closer to the optimal solutions. Ibrahim in [110] designed a hybrid SSA-PSO model. He integrated the update strategy of PSO into the structure of SSA so that the update for the current population was done by using either the SSA or PSO depending on the quality of the fitness function. PSO-mGA (micro Genetic Algorithm) model was presented in [101]. The ACO-DE model was developed in [23]. A novel SA-MFO model was presented by Sayed in [111]. The use of SA was to make the conversion rate slower, to reach to the global optima, and escape the local minima. A new MFO-based hybrid model was developed in [112] by combining MFO and Levy FA (LFA) algorithms. The other target of NIA-NIA hybridization is to refine the best solutions by implementing the NIAs sequentially as a pipeline where the operators of the first algorithm applied first then the operators of the other integrated algorithms are applied sequentially. These models often suffer from being slow in the search process. This hybridization strategy was applied in [113] to develop the PSO-GA model. In [46], the WOA-SA model was developed. In WOASA-1 (Low-Level Team-work Hybrid (LTH)) SA was used as an operator in WOA to enhance the exploitation. In WOASA-2 (High-Level Relay Hybrid (HRH)) SA was used after WOA to enhance the final solution. In 2020 [114], SA was hybridized with the HHO algorithm and AND and OR bitwise operations. SA was used to flee the HHO optimizer from local minima in the feature search space. A new hybrid binary version of the Bat Algorithm (BA) is suggested to solve feature selection problems. In [115], BA was hybridized with an enhanced version of the DE algorithm to reach the global solution. Hybridizing different NIAs to perform parallel exploration for the search space was also a primary target for other studies. Each algorithm generates its initial population and iteratively explores and evaluates the feature subsets. Using this strategy increases the speed of the search process. ACO-GA is an example of these hybrid models [116,117]. Recently, in [118], an enhanced hybrid approach using GWO and WOA was proposed to alleviate the drawbacks of both algorithms.

Another target for NIA-NIA hybridization is to enhance the initialization of the search using different NIAs. In these models, one algorithm is used to generate the initial solutions. Then the other combined algorithm is used to update these solutions. An example of these models is GA-IGWO presented in [119]. In [120], the hybridization of two Immune Firefly Algorithms (IFA1 and IFA2) was proposed. In IFA1, the FFA and Artificial Immune System (AIS) are used simultaneously to increase the global search of fireflies and select the best feature subset. IFA2 was used to study the influence the initial population on the searching progress of the AIS algorithm.

5.2.2. NIA-Classifier Hybridization

Hybridizing different classifiers such as SVM, Artificial Neural Network (ANN), aided Radial Basis Function (RBF), Optimum Path Forest (OPF), bagging, and Bayesian statistical with NIA for evaluating the solutions. Since classifiers have different capabilities regarding the training speed, computation complexity, and generalization capability; many studies investigated their influence when used in the wrappers framework. Other studies tried to make simultaneous FS and parameter optimization to enhance the performance of a classifier. NIA in these hybrid models works as a tuner to optimize the training parameters set up and select the optimal feature subset. In [121], a new wrapper approach was built to perform parallel FS and optimization for SVM parameters by exploiting the merits of MVO. Another hybrid model was presented in [122] for optimizing the SVM parameters simultaneously with selecting the best feature subsets using a GOA optimizer.

5.2.3. NIA-Filter (Wrapper-Filter) Hybridization

The filter-wrapper hybrid model is applied in two ways. First, a filter is applied to eliminate redundant and irrelevant features, minimize the dimensionality, and produce a reduced data set that is ready to be used by a wrapper. The second way to apply the filter-wrapper model is to use the filter in the structure of a wrapper to evaluate the generated features subsets. In [123], the Information gain and correlation-based were integrated with BPSO in models called IG-IBPSO and CB-IBPSO, respectively to solve FS. In [17], a MSPSO-F-score was developed. A mutual information filter was integrated with PSO and presented as a model called MI-PSO in [124]. PSO-MI and PSO-Intropy were developed in [125]. CS-MI was developed in [126]. BALO with QuickReduct and CEBARKCC filtering approaches were developed in [65]. In [5], IWOA-IG was developed. The ACO-MI model was presented in [127]. ACO with the multivariate filter was presented in [16]. The GA-MI model was presented in [128], GA-IG in [18,129], and GA-entropy in [130]. In [131], Relief-f was used with DE to rank the most significant features. Lately, in [132], an Embedded Chaotic Whale Survival Algorithm (ECWSA) has been proposed as a wrapper process and a filter method. In [133], an efficient hybrid model based on a combining filter and evolutionary wrapper approach was proposed for sentiment analysis of various topics on Twitter. The classification system was based on a SVM classifier and two FS methods using the ReliefF and MVO algorithms. Authors in [134] proposed a filter wrapper approach using a Sequential Floating Forward Search (SFFS) to acquire features for activity recognition. The model was validated using a benchmark dataset with a multiclass Support Vector Machine (SVM). The results show that the system is affected even with limited hardware resources.

5.3. Update Mechanism

The update modification aims to achieve a balance in exploration/exploitation processes. The update strategy is performed by either enhancing the update process of individuals or dynamically control the NIA parameters. A new variant of ACO was presented in [25,135]. The update strategy used performance and the number of selected features as heuristic information for ACO with no need for prior information about features. In [136], the gbest was updated based on some conditions. This strategy determines when to reset the gbest based on several epochs (iterations) in which the value of the gbest did not change. The same strategy was applied in [24,90]. Martinez, in [137], claimed that the initialization procedure and the update of all particles are not beneficial in high dimensional space. Hence, only a small subset of particles is randomly selected to be updated. The update for a particle is carried out by filling it with active features from the current particle, local best, and global best. This strategy was applied to the original PSO to get a new variant called CuPSO.

In [138], a new rule to update particle's positions was proposed. Instead of the original rule in BPSO that lies in giving equal probabilities to either selecting or not selecting a feature. $P(x_i^d(t) = 0) = P(x_i^d(t) = 1) = 0.5$ where $x_i^d(t)$ is the gene in the d dimension of the position vector at iteration t. The new rule was introduced to increase the probability of $x_i^d(t+1) = 0$ and reduce the probability of $x_i^d(t+1) = 1$. The idea in [139] is that pbest is usually updated based on the fitness value. However, if the new position has the same fitness value as the current pbest, then the pbest will not be updated even if the new solution corresponds to a smaller feature subset. This is a limitation of PSO. The proposed PSO was to update pbest and gbest into two stages where the priority is given first for the classification accuracy. Next, if the new particle position has the same performance as the current pbest but the number of features is smaller, then in this case, pbest will be updated and replaced by the new position.

In [96], the objective was to update PSO based on a clustering approach. The new GPSO uses Gaussian distribution. The idea was to group homogeneous features based on interactions between features, then PSO is used to select one representative feature from each cluster. Mafarja in [140] proposed five update strategies for the inertia weight (w) parameter. Linear, non-linear, coefficient, decreasing, oscillating, and logarithm were applied. His idea was based on applying an exploration operator more than exploitation at the beginning of the search then search those regions carefully to find the global optima. The conclusion was that the gradual decrease for the inertia weight (w) either linearly or non linearly improves BPSO. Mafarja in [141] studied the influence of the inertia weight (w) parameter on the performance of BPSO. He suggested the adaptive change for the exploration and exploitation by using a rank-based for updating the inertia weight (w) parameter. The same author presented in [83] the time-varying update strategy to improve the performance of the DA optimizer. In [142], Aljarah applied several asynchronous update strategies to solve the FS problem. An adaptive update strategy based on a descending linear function was used to update the SSA c1 parameter.

Recently in [143], a Binary DA (BDA) was proposed with new mechanisms to update its main coefficients. The main target is to apply the survival-of-the-fittest principle using different functions such as linear, quadratic, and sinusoidal. Three variants of BDA were introduced and compared with the standard DA. The new variants are linear-BDA, quadratic-BDA, and sinusoidal-BDA. Recently, in [144], a time-varying number of leaders and followers in a binary SSA (TVBSSA) with Random Weight Network (RWN) was proposed. In 2020, the CSA algorithm was enhanced in [145] using three enhancement strategies to solve the FS problem: Adaptive awareness probability to balance exploration and exploitation, dynamic local neighborhood to improve local search, and proposing a global search strategy to increase the global exploration of the crow.

In [146], an enhanced Binary Global Harmony Search algorithm, called IBGHS, was proposed to solve FS problems. An improved step is proposed to enhance the global search ability. In [147], a new update strategy based on ranking of the individuals was proposed. Each moth in the MFO algorithm is given a rank based on its fitness value. Therefore, a moth with a small fitness value will have a high rank so that there will be a great change in its position. On the other hand, a moth with a high fitness value will have a small rank so that there will be a small change in its position. This adaptive update strategy enhanced the performance of the optimizer. In [148], a time varying flame strategy was proposed to enhance the MFO algorithm. The number of flames represents the number of the best solution that decreases gradually across iterations. Different mathematical formulas were experimented with to decide the best formula that ensures exploitation around the best solution in the late stages.

5.4. Modified Population Structure

Zeng in [105] developed a novel GA with a dynamic chain-like agent population structure. CAGA aimed to enhance the population structure and diversity. This was better than the lattice-like agent population structure where agents do genetic operations just with neighboring agents. In [101], Mistry used a new population structure for PSO-mGA. He used a small-population secondary swarm strategy. A secondary swarm performs a collaborative role to avoid stagnation and overcome premature convergence.

5.5. Different Encoding Scheme

Galbally in [149], tried to minimize the verification error rate in the online signature system. Different encoding schemes were used, including binary and integer coding. GA with binary coding was used to search the complete search space. On the other hand, GA with integer coding was used for searching a subset of the search space. GA with an optimized descriptor weight or/and optimal descriptor subset was developed in [150] over MPEG-7. There were three different encoding schemes: A real-coded chromosome for weight optimization, binary-coded chromosome for the selection of optimal feature descriptor subset, and bi-coded chromosomes for simultaneous weight optimization and optimal feature descriptor selection. A new ensemble classifier was proposed in [151]. It was based on AdaBoost learning and parallel GA. A hybrid model parallel-GA-AdaBoost with different encoding schemes BGAFS and BCGAFS was proposed.

5.6. New Initialization

In [53], authors developed a hybrid model based on BPSO to solve the FS problem. A new initialization strategy called Opposition-based Learning (OBL) was proposed. The OBL strategy was used to enhance the initialization of particles and enforce diversity among solutions by considering the solution as well as its opposite solution simultaneously. OBL was used also to generate the opposite position of the gbest particle to get rid of the stagnation case. A novel framework based on IGWO and Kernel Extreme Learning Machine (KELM) was developed in [119]. In the GA-IGWO-KELM model, GA was applied first to generate high quality and diversified initial positions, then GWO was used to update the positions of the individuals in the discrete search space. Tubishat, in [5] developed a hybrid model called IWOA-SVM-IG. The OBL strategy was applied for increasing the level of diversity in the initial solutions generated by standard WOA. In [152], a quasi-oppositional learning-based Multi-Verse Optimization (MVO) algorithm was used to improve the initial setting up of solutions.

5.7. New Fitness Function

Chakraborty [153], proposed the PSO algorithm where the fitness evaluation of each particle is based on ambiguity. The new fuzzy evaluation function was used to measure the fuzziness of a fuzzy set. The best feature was represented with minimum intraclass ambiguity as well as maximum interclass ambiguity. In [154], GA was proposed with Fisher's Linear Discriminant function in a model called GA-FLD. The new evaluation function estimates the probability distribution of the class in the N-dimensional feature space. It uses also the cardinality of the feature subset using covariance matrices which is an extension of FLD. This method was used to measure the statistical proprieties of the feature subset. Authors in [53] developed BPSO with a new fitness function based on dynamic inertia weight. High inertia weights are assigned to particles with low fitness values to facilitate more exploration of the search space. Low inertia weights are assigned to particles with high fitness to facilitate more exploitation. In [6], GWO was modified using several fitness functions. The fitness functions were accuracy, Hausdorff distance, Jeffries–Matusita (JM) distance, the weighted sum of the accuracy and Hausdorff, and the weighted sum of the accuracy and JM. In [155], different fitness functions were used to enhance the performance of the MFO algorithm. The best fitness function was the one that was applied across two-stages. The first stage optimizes the classification performance only while the second stage takes into consideration the number of genes. The results show that the proposed fitness functions can achieve better classification results compared with the fitness function that takes into account only the classification performance.

5.8. Multi Objective

Zio [156], developed a system for nuclear plants based on GA to select among the several measured plant parameters. The first approach applied was single objective GA with fuzzy k-Nearest Neighbor classifier (KNN) then multi-objective approaches were applied. Mandal, in [157] developed a prediction system based on a multi-objective PSO that satisfies the Pareto front and makes a trade-off between the non-dominated solutions based on different objectives. The proposed multi-objective PSO FS algorithm performed a dual-task where the first objective was maximizing the mutual information between a feature and class label (relevance) and the second objective was minimizing the mutual information among the features (redundancy). A Dynamic Locality Multi-Objective SSA for FS was proposed in [158]. In [159], a multi-objective FS method was proposed based on bacterial foraging optimization. In [160], a multi-objective PSO modified by Levy Flight was proposed for intrusion detection in Internet of Things (IoT). RWS mechanism was used to remove redundant features and information exchange mechanisms to avoid local minima. A systematic review of the multi-objective FS problem that covered the related studies in the period (2012, 2019) was introduced in [161].

5.9. Parallelism

In [162], Punch applied a wrapper FS based on GA to biological datasets. 5KNN was modified to work on weighted features (multiplied by weights according to their importance). The new approach was applied to a parallel distributed machine (Sparc and HP). A new ensemble classifier was proposed in [151]. It was based on AdaBoost learning and parallel GA. A parallel version of GA was applied on 16 processors with a master-slave paradigm and KNN was used as a base classifier. Ghamisi in [163] applied the parallelism strategy on PSO. Darwinian PSO (DPSO) was based on running many PSO algorithms simultaneously. Each algorithm runs as a different swarm on the same problem. A natural selection process was applied by rewarding the swarm that got better results and extending its particles' life so that new descendants were spawned. On the other hand, the swarm with suboptimal results (stagnate) was punched so its search area was discarded and its life was reduced by deleting its particles.

6. NIAs FS Applications

This section provides an extensive discussion on the use of modified NIA algorithms in different applications.

6.1. Microarray Gene Expression Classification

In [164], a hybrid model of GA and SVM was developed to perform FS and kernel parameter optimization. GA-SVM is a recommended approach for FS especially when the kernel parameters are optimized and the number of selected features is not known beforehand. Huang, in [128], developed a new GA-based wrapper approach. He adopted two stages of optimizations. The outer optimization stage (global search) applied a fitness function based on mutual information between actual classes and predicted classes. The second stage (the inner optimization) implements a local search (filter manner) based on feature ranking. A gene selection approach based on ACO was developed in [165]. A high-dimensional multi-class cancer gene expression (GCM) and colon cancer data sets were used. The comparisons were conducted with several rank-based models. The simulated results proved the validity of the proposed ACO approach for FS in high dimensional data sets.

A reliable FS technique was developed in [136] for selecting relevant features in the gene expression data set. The proposed methodology was IBPSO-KNN. The results of the accuracy increased by 2.85% compared with other methods in the literature. Yang [92], presented a new modified model for BPSO and applied it over six multi-category cancer-related human gene expression data set. Yang [18] developed a hybrid filter wrapper method for FS in microarray data sets using GA and IG. The ranking of features was performed using a decision tree. Experiments showed that the IG-GA algorithm simplified the number of gene expression levels and either achieved higher accuracy or used fewer features compared to other methods. A hybrid filter-wrapper model based on Information Gain (IG), Correlation-based (CFS), and IBPSO was proposed in [123]. Kabir [166] developed a new hybrid model based on GA, NN, MI, and local search operators. A new PSO model that has the capability of discovering biomarkers from microarray data was designed in [137].

Chuang [52] developed a hybrid model for FS and classification of large-dimensional microarray data sets. Mohammad [138] developed a diagnostic medical model based on IBPSO to find the least possible number of discriminative genes. One year later, Kabir developed an ACO-based FS model in [167]. The ACOFS target was to select the salient features with the smallest size. The model combined the ACO, neural network, filter, and included an update for the rules-based on subset size determination scheme. In [24], the PSO variant was superior to other methods in terms of performance, a number of features, and cost.

A new filter-based approach based on the CS optimizer, mutual Information filter (MI), entropy, and Artificial Neural Network (ANN) classifier was proposed in [126]. The entropy and mutual information were applied in the fitness function to calculate the relevance and redundancy for the feature subsets. Banka developed a new modified version of the PSO algorithm in [98]. Three benchmark data sets were used for colon cancer, defused B-cell lymphoma, and leukemia. The model achieved a minimal number of features and a higher classification accuracy. In [100], a model for cancer gene selection and cancer classification was developed based on BQPSO and SVM with LOOCV. Five DNA microarray data sets were used. Experiments showed better results for BQPSO/SVM compared with BPSO/SVM and GA/SVM in terms of accuracy, robustness, and the number of genes selected. Zawbaa [4], handled the complexity of the FS problem in data sets with large dimensionality and few numbers of instances by developing a novel hybrid system called GWO-ALO. A total of 27 different microarray and image processing data sets were used. Some of the data sets were very complex with 50,000 features and less than 200 instances. The experiments showed promising results when compared with GA and PSO. Ibrahim, in [168], developed a novel wrapper approach based on combining SVM with the GOA optimizer, and then he applied the hybrid model on three biomedical data sets from Iraqi cancer patients and UCI.

6.2. Facial Expression Recognition

A new modified ACO-based FS approach without a need for prior knowledge about features was presented in [25]. The experiments were applied to an ORL gray-scale face image database. The same author proposed after one year another ACO-based FS approach [135], which showed superior performance compared with GA-based and other ACO FS approaches. Aneesh, in [76], proposed a new face recognition technology using a modified version of BPSO, called Accelerated BPSO (ABPSO). ORL database images taken at the AT&T Laboratories and Cropped Yale B database-4 were used in the experiments. A biometric technique for Face Recognition (FR) based on BPSO was developed in [97]. Seven benchmark databases, namely, Cambridge ORL, UMIST, Extended YaleB, CMUPIE, Color FERET, FEI, and HP were used in the experiments.

Zhang [112] developed a facial recognition system based on the MFO-LFA-SA hybrid model to avoid premature stagnation and to guide the search procedure towards global optima. MFO logarithmic spiral search behavior increased the exploitation power meanwhile the LFA used the attractiveness function for more exploration in the search space. The SA empowered the exploitation around the most promising solution. Experiments used frontal-view images extracted from CK+JAFFE, MMI, and BU-3DFE. MFO-LFA FS outperformed other facial expression recognition models. Mistry [101] incorporated several update mechanisms in one model including the hybridization of a PSO- and mGA- (micro Genetic Algorithm), modified population structure, new velocity update strategy, diversity maintenance strategy, and a sub-dimension-based regional facial feature search strategy. Cross-domain images from the extended Cohn Kanade and MMI benchmark databases were used in the experiments besides multiple classifiers including NN with back-propagation, a multi-class SVM, and ensemble classifiers.

In [169], a system for Facial Emotion Recognition (FER) was developed based on GWO-NN. The hybridization was used to tune the weights with less training error, then it classified the emotions from the selected features. The proposed FER system was evaluated using the JAFFE and Cohn–Kanade database and the results showed higher accuracy compared with conventional methods.

6.3. Medical Applications

A new recognition system for skin tumor diagnosis was developed by handels in [170]. A GA algorithm was used to extract the most suitable features from 2D images that characterize the structure of the skin surface. NN with back-propagation was used as a learning paradigm that was trained using the selected feature sets. Different network topologies and parameter settings were investigated for optimization purposes and GA was compared with heuristic greedy algorithms. The GA skin tumor achieved the highest classification performance of 97.7%. An optimized mass detection system for digitized mammograms was developed by Zheng [171]. A GA-BBN hybrid model was used to classify positive and negative regions for masses depicted in digitized mammograms. The results showed that GA achieved the same ratio of feature reduction in comparison with the exhaustive search but reduced the total computation time by a factor of 65. In [113], a hybrid PSO-GA FS system was developed to improve the cancer classification performance and reduce the cost of medical diagnoses. Chakraborty [153] proposed a modified version of PSO using a new fuzzy evaluation.

In [172], different hybridization models were developed using the GA algorithm with different neural classifiers to get the best feature subset while preserving accuracy. A comparison was conducted between GA-KNN, GA-BP-NN, GA-RBF-NN, and GA-LQV-NN. The results showed that GA with neural classifiers were more robust and effective. In [173], Babaoglu investigated the effectiveness of both BPSO and GA as FS models for determining the existence of Coronary Artery Disease (CAD). BPSO-SVM and GA-SVM were applied on a data set obtained from patients who had performed Exercise Stress Testing (EST) and coronary angiography. The results showed that the BPSO-FS method was more successful than GA-FS and SVM on determining CAD. An automatic breast cancer diagnosis framework was designed by Ahmad [174]. The developed hybrid Genetic Algorithm Multilayer Perceptron (GA-MLP) model performed simultaneous FS and parameter optimization of ANN.

Three different variations of the backpropagation training algorithm, namely the resilient backpropagation (GAANN-RP), Levenberg Marquardt (GAANN-LM), and Gradient Descent with momentum (GAANN-GD) were investigated. The Wisconsin Breast Cancer Database (WBCD) was used. The experiments showed that the best accuracy was achieved by the RP. Sheikhpour developed a hybrid model to distinguish between benign and malignant breast tumors [175].

PSO-KDE was used to minimize the kernel density estimation error and avoid the time needed by the surgical biopsy. The Wisconsin Breast Cancer Data set (WBCD) and Wisconsin Diagnosis Breast Cancer Database (WDBC) were used. Sayed [176] developed an automatic system based on MFO for Alzheimer's Disease (AD) diagnosis. It was able to distinguish three kinds of classes including Normal, AD, and Cognitive Impairment. A benchmark data set consisted of 20 patients from the National Alzheimer's Coordinating Center (NACC). Experiments showed that the SVM-polynomial kernel function was the best one in terms of accuracy precision, recall, and f-score. A novel medical diagnosis framework based on IGWO and KELM was developed in [119].

The model was investigated on Parkinson's and breast cancer disease data sets. The comparison was performed between IGWO-KELM, GWO-KELM, and GA-KELM. The experimental results proved that the proposed method was better than the other two competitive counterparts. One year later, Sayed developed a new approach for mitosis detection in breast cancer histopathology slide images based on the MFO FS algorithm [177]. MFO was used to extract the best discriminating features of mitosis cells such as statistical, shape, texture, and energy then the selected features were used to feed the Classification and Regression Tree (CART) to make classification into either mitosis and non-mitosis. Wang [3] developed an efficient medical diagnosis tool based on CMFO and KELM to minimize the number of features and to perform parameters optimization for KELM.

6.4. Handwritten Letter Recognition

The target in [178] was to study which one of the machine learning algorithms had the right bias to solve specific natural language processing tasks. GA achieved the best results on a language processing WSD data set. In [89], authors developed a hybrid GA to mitigate the weakness of standard GA in fine-tuning near the local minima. The proposed approach was validated using a data set gained by extracting the gray-mesh features from the CENPARMI handwritten numeral samples. Galbally [149] tried to find a way to minimize the verification error rate in the online signature verification system. A GA-based approach with new modification was proposed. Experiments were conducted on the MCYT signature database with 330 users and 16,500 signatures. The new approach showed remarkable performance in all the carried out experiments. Zeng [105] developed a novel GA with a dynamic chain-like agent population structure and dynamic neighboring competitive selection strategy. He used a letter-recognition database from UC Irvine (UCI). The experimental results showed that the feature subset generated from CAGA achieved a higher classification rate, more stability, and lower classification complexity in comparison with the other four GAs. A novel FS algorithm based on ACO was presented in [179] to improve the performance of the algorithm in text categorization. Comparisons were conducted with GA, information gain, and Chi Square test (CHI) on the Reuters-21578 data set. The proposed approach proved its superiority concerning the Reuters-21578 data set. In [129], Principal Component Analysis (PCA) was used with the IG filter method and GA optimizer in a model called IG-GA-PCA. In the first stage, the IG method was applied to rank the terms of the document according to their importance. In the second stage, GA and PCA FS and feature extraction methods were applied separately to the ranked terms. Experiments used both Reuters-21578 and Classic3 data sets. The experiments showed that the IG-GA-PCA model could achieve high categorization results as measured by precision, recall, and F-measure. In [154], a GA-FLD-based FS approach was used in order to find features subsets that could optimally discriminate samples from different classes without prior knowledge about features dimensionality. Another modification based on fitness function were also proposed. Three standard databases of handwritten digits and one of handwritten letters were used in the experiments. In [53], authors developed a hybrid intelligent algorithm using BPSO and other operators to solve the FS problem in the text clustering. A new initialization strategy, new fitness function, and new operator were proposed. The Reuters-21578, Classic4, and WebKB benchmark text data sets were used. The results showed higher clustering accuracy and improved the convergence speed of BPSO. Ewees, in [180] introduced a new approach for Arabic handwritten letter recognition (AHLR) called MFO-AHLR. A data set for Arabic handwritten letter images (CENPARMI) was used. Results showed that MFO-AHLR achieved a 99.25% accuracy, which was the highest ratio achieved among all AHLR approaches. Tubishat, in [5], developed a novel hybrid model for Arabic SA. The targets of the study were to mitigate the limitations of the WOA such as local minima, slow convergence diversity, and over-fitting problems. A hybrid model IWOA-SVM-IG was applied over four Arabic benchmark data sets for sentiment analysis. IWOA was compared with six well-known optimization algorithms and two deep learning algorithms, namely Convolution NN (CNN) and Long Short-term Memory (LSTM). The results showed that the IWOA algorithm outperformed all other algorithms.

6.5. Hyper Spectral Images Processing

Tackett in [181] worked on extracting the statistical features from a large noisy US Army NVEOD Terrain Board imagery database using GP. In [182], a new model was proposed based on GA, Bayesian classification, and a new proposed fitness function to discriminate the targets from clutters in SAR images. Jarvis [183] developed a novel approach based on GA and DFA for the selection of important discriminatory variables from Fourier Transform Infrared (FT-IR) spectroscopic data. The GA achieved 16% reduction in the model error. The GA-SVM model for hyper-spectral data classification was proposed in [184]. The proposed GA-SVM was tested on an HYPERION hyper-spectral image. Experiments demonstrated that the number of bands was reduced from 198 to 13, while accuracy increased from 88.81% to 92.51%. A GA-based image annotation system with optimized descriptor weights or/and optimal descriptor subset over MPEG-7 was developed in [150]. The Corel image database consisted of 2000 images with 20 categories used. Experiments showed that the binary-coded GA and the bi-coded GA improved the accuracy of the image annotation system by 7%, 9%, and 13.6%, respectively compared to the commonly used methods.

A new ensemble classifier was proposed in [151]. It was based on AdaBoost and parallel GA in the context of the FS problem for image annotation in MPEG-7 standard. The experiments were performed over 2000 classified Corel images. In [185], a new approach based on GA, SVM, MI, and BB was developed to search for the best combination of bands in the hyper spectra images. MI was used as a pre-processing step for band grouping based on the correlation between bands and classes. GA-SVM was used to search for the optimal combinations of bands that increase accuracy. A post-processing step based on BB was used to filter out those irrelevant band groups. Ghamisi [163] applied the FODPSO SVM approach to determine the most informative bands in the Hekla and Indian Pines hyper-spectral data set using the parallelism modification technique. In the same year, Ghamisi [186] presented a new hybrid approach based on GA, PSO, and SVM. His target was to detect roads from a background in complex urban images. He integrated the standard velocity and update rules of PSO with selection, crossover, and mutation from GA. In [6], Medjahed developed a novel GWO framework for Pavia and AVIRIS hyper-spectral images data sets.

6.6. Protein and Related Genome Annotation

In [116], a new FS model based on ACO-GA was proposed. Both ACO and GA generated the feature subsets in parallel then the generated subsets were evaluated by a certain fitness function. ACO used GA operators to update the solutions. The GPCR-PROSITE dataset and ENZYME-PROSITE challenging protein sequences data sets were used. Mandal [157], developed a prediction system to identify the possible subcellular location of a protein-based on a multi-objective PSO.

6.7. Biochemistry and Drug Design

Raymer [187] developed a system that integrates FE, FS, and classifier training using GA and KNN. This approach was applied in biochemistry and drug design for the identification of favorable water-binding sites on protein surfaces. The approach was validated using protein water interactions from a biochemistry field. Another model was developed by Salcedo [104]. The proposed FS model was based on GA and m-features operator (OR operator). The new approach was evaluated using two machine learning classification problems; the first one used two artificial data sets and the second one was a real application in molecular bioactivity for a drug design taken from the ones used in the KDD Cup. THe m-features operator improved the GA performance over the other existing approaches.

6.8. Electroencephalogram (EEG) Application

Palani [188] used GA and Fuzzy ARTMAP (FA) NN for FS. GA-FA-NN was used with the VEP data which was recorded from 10 alcoholics and 10 controls. The target was to classify alcoholics and controls, using multi-channel EEG signals. The discriminatory spectral bands reduced from 7 to 2. The identification of useful spectral power ratios produced better performance. In [189], a hybrid GA-SVM model was used to extract the favorable patterns from noisy multidimensional time series obtained from EEG which are a base for Brain-computer Interfaces (BCIs). The data set was collected by a procedure in which subjects were placed in a dim, sound controlled room. The proposed nonlinear system was better than other linear approaches with a slight difference. A novel ACO-DE

FS system called ANTDE was presented in [23]. It could cope with the limitations of ACO regarding the sequential generation for solutions. ANTDE was used in EEG and Myoelectric Control (MEC) biosignal applications. Wang [130] developed a BCI system using a hybrid model GA-SVM-entropy and 28 EEG channels. Noori designed an effective BCI in [190]. He used a new version of GA based on SVM to get smaller optimal features from functional Near-infrared Spectroscopy signals (fNIRS). The experiments were established by recruiting seven subjects who do not have any psychological disorder. Subjects were seated in a quiet room and asked to relax to settle down their responses before beginning to perform mental arithmetic tasks for a certain period.

6.9. Financial Prediction

In [191], a new financial prediction model was proposed. A hybrid model SVM-GA was evaluated using 15 business data sets. Each data set consisted of 186 sampled firms. GA-SVM achieved a prediction accuracy of up to 95.56% for all the tested business data. In [192], the authors developed a hybrid fuzzy-GA approach for stock selection. The fuzzy-based scoring mechanism was applied for scoring a set of stocks then the topmost stocks were selected. GA applied for performing a dual job of FS and parameter optimization. The constituent stocks of the 200 largest market capitalization listed in the Taiwan Stock Exchange were used in the experiments.

6.10. Software Product Line Estimation

Oliveira [22] investigated the use of the GA method for simultaneous FS and parameters optimization of Support Vector Regression (SVR) when applied for software effort estimates. GA, SVR, MLP, and model trees were used. Six benchmark data sets of software projects, namely, Desharnais, NASA, COCOMO, Albrecht, Kemerer and Koten, and Gray were used in the experiments. In [106], Guo presented a new methodology for FS in the software application. The target of the new modified GA was optimizing FS in a Software Product Line (SPLs) to find a feature subset with an optimal product capability subject to feature model constraints and resource constraints. The results showed that GA FS algorithms produced a system with high performance and in 45–99% less time than existing heuristic FS techniques.

6.11. Spam Detection in Emails

Temitayo [193] developed a new approach for the classification of emails, either spam or legitimate. GA was used to perform simultaneous FS and parameter optimization. The hybrid GA-SVM spam detection model was evaluated using a Spam Assassin (6000 emails) data set. Experiments showed that GA-SVM improved the results compared with SVM by achieving a higher recognition rate with only a few feature subsets. In [85], a mutation-based BPSO FS model was developed in an email application. A data set of 6000 emails manually collected during the year 2012 was used. The proposed was able to effectively reduce the false-positive error. In [194], a hybrid GA-RWN was used for identifying the most relevant features in spam emails and automatic tuning for the hidden neurons. The GA-RWN achieved promising results according to the spam detection rate and optimization for the configuration of its core classifier. Lately, in [195], a novel Northern Bald Ibis Algorithm (NOA) was used with a SVM classifier to get an optimal feature subset of the Enron-spam dataset.

6.12. Other Various Applications

Zio [156] developed an efficient transient diagnosis system for nuclear power plants based on GA to select among the several measured plant parameters. In [61], the target was addressing the problem of a lengthy Intrusion Detection (ID) process based on attributes of network packets. Rough-PSO was used and evaluated using the KDDCup 1999 data set. An automatic FS model that can choose the most relevant features from password typing patterns was designed in [196]. The data sets were captured on a Sun Sparc-Station by a program in an X window environment in which the keystroke duration times were measured. Rodrigues in [197] proposed a CS-OPF model for theft detection in power distribution systems. The proposed model was evaluated using two data sets from a Brazilian electrical power company. Experiments proved the robustness of the CS-OPF model by increasing the theft recognition up to 40%. Zhang [127] developed a new forecaster FS model based on combining MI and ACO. The ACO-MI model was applied on forecasters data sets at the Australian Bureau of Meteorology. A system for diagnosing different types of fault in a gearbox was designed in [198]. Hassanien [67] developed an automatic tomato disease detection system based on integrating rough set with MFO.

6.13. An Open Source Evolopy-FS Framework

EvoloPy-FS [199] is an open-source FS software tool developed by our team and it is publicly available on (www.evo-ml.com). It serves as an explicit white-box NIAs-FS optimization framework. The main objective was to support researchers from different disciplines with an easy-to-use, transparent, and automated NIAs-FS optimization tool. The framework contains severe recent NIAs algorithms written in Python and a set of different operators such as transfer functions (S-TFs and V-TFs). Moreover, the framework applies wrappers, filters and a hybrid filter-wrapper, different evaluation metrics, and allows for loading data from different resources. Evolopy-FS is a continuation of our path, which is building an integrated optimization environment. The work was started by EvoloPy [200] for global optimization problems then EvoloPy-NN for optimizing MLP and recently Evolopy-FS for optimizing the feature selection process. In [199], authors constructed the experiments based on 30 different well-regarded data sets from common repositories such as UCI and Kaggle. The comparisons were conducted between wrapper FS, filter FS, and hybrid filter-wrapper approaches. It was shown that wrapper and hybrid filter-wrapper were superior and more trustable in dealing with large dimensionality data sets. However, the filter approach was faster and generated results in a shorter time and fewer computational efforts.

7. Assessment and Evaluation of NIAs FS Modification Techniques

As discussed, NIAs-FS approaches achieved big contributions and clear success in solving the FS problem in different domains. This section presents the results from the analysis of modified NIAs-FS studies. Table 1 shows a summary of the main studies in the literature that adopted new operators as modification techniques for NIAs-FS, Table 2 shows a summary of the main studies in the literature that adopted hybridization modification technique for NIAs-FS, Table 3 shows a summary of main studies in the literature that adopted the remaining modifications techniques for NIAs-FS, Table 4 shows a summary of main modifications applied in the literature on main NIAs (applied/not applied), Table 5 shows a summary of main modifications applied in the literature on main NIAs (by numbers), Table **??** shows a summary of the main studies in the literature that applied modified NIAs FS in applications, and Table 7 shows a summary of modifications applied on NIAa-FS in the main applications. It was observed that 34 different operators were applied on NIAs wrappers in 48 different papers. Some references adopted over one operator in their work. As it is clear also from the list, the most applied operator is the chaotic map, which was applied in 10 references, then rough set in 6 references, then selection operators (RWS, TS) in 5, then S-shaped and V-shaped transfer functions and crossover in 4 references. The mutation was applied in 3 and UC, DE, and local search operators each were been adopted by 2 references. A single reference adopted the remaining operators. It was found that the PSO wrapper was the most modified optimizer using newly adopted operators for tackling the FS problem. It was modified using a new operator in 21 references. In addition, GA was modified in 6 references, WOA in 4, CS in 3, SSA in 3, GWO, GOA, and MFO each one was modified by a new operator in 2 references. For DA, FFA, LA, BA, MVO, ABC, CSO, DE, and CSA, the number of references was 1. For FPA and ACO, no work applies new operators to their algorithms for solving the FS problem.

It was clear from the analysis that the hybridization modification technique was applied in 75 references to solve FS. This counting result shows that hybridization is the most widely applied modification technique to enhance NIAs wrappers in the FS domain. This high number of work comes from GA, which is the NIA that had the most number of works regarding wrapper hybridization. GA wasapplied hybridization in 38 different works, which is much higher than 6, the number of works that adopted new operators to GA. We can infer from this works count and from the contribution of studies that hybridization is the best suitable modification technique to be applied with GA. ACO also were hybridized in 7 references, while no work adopted a new operator to modify ACO. Conversely, PSO hybridization works were 11, which is less than 21, the number of works with new operators, thus we can again infer from these counts and the contribution of studies that adopting a new operator to PSO is more suitable than hybridization. It was also noticed that hybridization using different kinds of classifiers was the most prominent hybridization technique.

There are 49 studies that tried to investigate the influence of the classification technique on the performance of wrappers for optimizing FS, some of these studies applied simultaneous optimization for FS and a classification/prediction task by tuning the parameters of the classifier with applying FS. The next widespread hybridization technique is a filter-wrapper, which was applied in 14 studies and was very effective in dealing with large dimensionality feature space. Hybridization techniques that tried to balance the exploration/exploitation of the search space also were adopted by a considerable number of works. In summary, PSO and GA are the most widely modified NIAs-FS approaches. They were equally modified and used. Each one of them was adopted and modified for FS in 56 references of the gathered studies.

On the other side, regarding applications of NIAs-FS, it was evident that microarray gene expression classification is the most dominant application where NIAs-FS approaches were applied in 18 studies with a ratio 24% concerning other applications. The medical application was the second prominent application for applying NIAs-FS approaches with a ratio of 21%. The medical application includes different medical branches SONAR, tumor, mass, and various disease detection, medical diagnosis, medical data, and bio-signal analysis. Then, follows hyper-spectral image with a ratio of 17%, Arabic handwritten recognition with a ratio of 13%, facial expression recognition with a ratio of 9%, EEG application with a ratio of 7%, financial diagnosis with a ratio of 5%, and spam detection with a ratio of 4%. Furthermore, it is noticeable that GA is the most dominant NIA optimizer for optimizing FS in applications with a ratio of 45%. PSO is the second most widespread optimizer with a ratio of 26%, then ACO with a ratio of 11%, MFO with a ratio of 7%, GWO with a ratio of 6%, CS with a ratio of 2%, WOA with a ratio of 2%, and GOA with a ratio of 1%.

No Free Lunch (NFL) theorem [201] states no optimization algorithm can solve all the optimization problems equally. The success in solving a specific problem does not guarantee that the algorithm will perform similarly for other problems. On average, all the optimization algorithms perform equally. This theorem has motivated researchers to develop new algorithms or improve the existing ones to solve another wide area of optimization problems, such as feature selection. Researchers are advised to read the following references as they are the most cited papers after 2010 in the field of NIAs-FS: [17,51,80,90,94,124,202].

Table 1. Summary of main ro pub the tables, and please check if it is the background color can be deleted, same as follows. studies in the literature that adopted new operators as modification techniques for Nature Inspired Algorithms Feature Selection NIAs-FS.

New Operator	NIA Wrapper and No. of Publications	References	Total NIAs
Chaotic maps	PSO(4), SSA(2), MVO(1), CSA(1) WOA(1), MFO(1)	[3,51–58]	10
Rough set	PSO(2), CS(2), FA(1), MFO(1)	[61–64,66,67]	6
Selection operators (RWS,TS)	GOA(1), WOA(2), ABC(1), DE(1)	[26,46,72–74]	5
Sigmoidal function	PSO(1), GWO(1), CS(1), BA(1), GOA(1)	[6,76–79]	5
S-shaped and V-shaped TFs	PSO(1), SSA(1), DA(1), GOA(1)	[79,80,82,83]	4

Crossover
Mutation
	GWO(1), SSA(1), WOA(1), GOA(1)
	PSO(2), GOA(1)
		[73,79,82,84]
		[53,79,85]
			4
			3

Uniform Combination (UC)	PSO(2)	[80,91]	2
Local search	PSO(2)	[24,90]	2
DE evolutionary operators	WOA(1), ABC(1)	[5,26]	2
Boolean algebra operation	PSO(1)	[92]	1
Logistic regression	PSO(1)	[202]	1
Catfish strategy	PSO(1)	[94]	1
Feature subset ranking	PSO(1)	[95]	1
Statistical clustering	PSO(1)	[96]	1
Threshold	PSO(1)	[97]	1
Gaussian sampling	PSO(1)	[91]	1
Reinforced memory strategy	PSO(1)	[91]	1
XOR operator	PSO(1)	[98]	1
Correlation information	PSO(1)	[99]	1
Binary quantum	PSO(1)	[100]	1
Reinitialization strategy	PSO(1)	[101]	1
Non replaceable memory	PSO(1)	[101]	1
Levy flight	CS(1)	[64]	1
Return cost indicator	FFA(1)	[102]	1
Pareto dominance based	FFA(1)	[102]	1
Movement operator and adaptive jump	FFA(1)	[102]	1
Greedy search	ALO(1)	[103]	1
Evolutionary Population Dynamics (EPD)	GOA(1)	[72]	1

DE-based neighborhood mechanism
Repair mechanism
	ABC(1)
	DE(1)
		[26]
		[74]
			1
			1

m-features operator (OR operator)	GA(1)	[104]	1
Dynamic neighboring genetic	GA(1)	[105]	1
Repair operator	GA(1)	[106]	1

Table 2. Summary of main studies in the literature that adopted hybridization modification technique for NIAs-FS.

Target	Type	Strategy	Model	References	Total
Enhance the exploitation	NIA-NIA (Population-trajectory (mimetic) SI-EA, SI-SI)	Global search followed by local search	GWO-ALO, WOA-SA, CSA-FPA, SSA-PSO, PSO-mGA, ACO-DE, MFO-SA, MFO-LFA	[4,23,46,101,108,110–112]	8
Refine the best solutions	NIA-NIA	Implementing NIAs sequentially as a pipeline. Apply operators of 1st algorithm then apply operators of the 2nd algorithm sequentially	WOA-SA, PSO-GA	[46,113]	2
Speed up the search process	NIA-NIA	Perform parallel exploration	ACO-GA	[116,117]	2
Enhance the initialization process	NIA-NIA	Generate initial solutions by one algorithm then update them using the other algorithm	GA-IGWO	[119]	1
Improve the training process. Improve the evaluation process. Reduce the computation complexity. Investigate the capability of different classifiers. Simultaneous parameters optimization and feature reduction. Study the influence of different evaluation strategies on wrappers performance.	NIA-Classifier	For the simultaneous parameter and FS optimization, NIA works as a tuner to optimize the training parameters set up, selecting the optimal feature subset by making new representations of an individual, in such a way the length of the individual equals the number of parameters and number of features and adjusts the values of genes by either doing real or binary conversion.	PSO-KDE, GA-IGWO-KELM, GWO-NN, CS-OPF, CSA-FPA-OPF, BBA-OPF, BBA-NaiveBayes, {MVO-RF, MVO-J48, MVO-Kstar, MVO-LMT}, MVO-SVM, GOA-SVM, ACO-SVM, ACO-NN, CMFO-KELM, CSO-SVM, GA-KNN, {GA-NN, GA-BNN, GA-MLP, GA-BP-NN, GA-RBF-NN, GA-LQV-NN, GAANN-RP, GAANN-LM, GAANN-GD, GA-RWN}, GA-C4.5, GA-SVM, GA-Bayesian, GA-FKNN, GA-adaboost, GA-SVR,	[3,18,22,22,56,77,78,108,117, 119,121,122,128,130,151, 156,164,166,166–169,171–175,182,185,188–191,194,196–198,203–208]	49

Table 2. *Cont.*

Target	Type	Strategy	Model	References	Total
Minimize the dimensionality of the large datasets, eliminate the redundant/ irrelevant features, evaluate the generated features subsets	NIA-Filter (Wrapper-Filter)	Classically applied in two steps: Filtering of the features applied first then a wrapper is applied on the reduced dataset. Other studies tried to embed the filter in the structure of a wrapper in order to evaluate the generated features subsets	{IG-IBPSO, CFS-IBPSO}, MSPSO-F-Score, MI-PSO, {BPSO-G (Intropy-PSO), BPSO-P (MI-PSO)}, CS-MI-Entropy, LA-QuickReduct-CEBARKCC, IWOA-IG, ACO-MI, ACO-Multivariate filter, GA-MI, IG-GA, GA-Entropy	[5,16–18,64,65,124–130]	14

Table 3. Summary of main studies in the literature that adopted the remaining modifications techniques for NIAs-FS.

Modification	NIA Wrapper and No. Publications	References	Total Works
Update mechanism	PSO(10), SSA(1), DA(1), ACO(5)	[24,25,53,83,90,96,135–139,141,142,179, 209–211]	17
Modified population structure	PSO(1), SSA(1), GA(1)	[101,105,142]	3
Different encoding scheme	PSO(3), GA(4)	[17,90,91,105,149,150,212]	7
New initialization	PSO(2), GWO(1)	[53,84,139]	3
New fitness function	PSO(2), GWO(1), FFA(1), WOA(1), GA(4)	[5,6,106,153,154,182,183,213,214]	9
Multi objective	PSO(4), ACO(1), GA(1)	[156,157,209,215–217]	6
Parallelism	PSO(2), GA(2)	[17,151,162,163]	4

Table 4. Summary of main modifications applied in the literature on main NIAs-FS (applied/ not applied).

Modification	PSO	GWO	SSA	CS	DA	FFA	LA	BA	MVO	GOA	WOA	ACO	MFO	GA
New operator	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Hybridization	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Update mechanism	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
Modified population structure	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Different encoding scheme	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
New initialization	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
New fitness function	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓
Multi objective	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓
Parallelism	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

Table 5. Summary of main modifications applied in the literature on main NIAs-FS (by number of studies).

Modification	**PSO**	**GWO**	**SSA**	**CS**	**DA**	**FFA**	**LA**	**BA**	**MVO**	**GOA**	**WOA**	**ACO**	**MFO**	**GA**	**Total Studies/Modification**
New operator	21	2	3	3	1	1	1	1	1	2	4	0	2	6	48
Hybridization	11	2	1	3	0	0	1	3	2	2	2	7	3	38	75
Update mechanism	10	0	1	0	1	0	0	0	0	0	0	5	0	0	17
Modified population structure	1	0	1	0	0	0	0	0	0	0	0	0	0	1	**3**
Different encoding scheme	3	0	0	0	0	0	0	0	0	0	0	0	0	4	7
New initialization	2	1	0	0	0	0	0	0	0	0	0	0	0	0	**3**
New fitness function	2	1	0	0	0	1	0	0	0	0	1	0	0	4	9
Multi objective	4	0	0	0	0	0	0	0	0	0	0	1	0	1	6
Parallelism	2	0	0	0	0	0	0	0	0	0	0	0	0	2	4
Total studies/FS-NIA	56	6	6	6	2	2	2	4	3	4	7	13	5	56	172

Table 6. Summary of main studies in the literature that applied modified NIAs-FS in applications.

FS Application	NIA Wrapper and No. of Publications	References	Total NIAs
Microarray gene expression classification. (DNA micro array classification)	PSO(9), GWO(1), CS(1), GOA(1), ACO(2), GA(4)	[4,18,24,52,98,100,123,126,128,136–138,164–168]	18
Facial expression recognition	PSO(3), GWO(1), ACO(2), FO(1)	[25,76,97,101,112,135,169]	7
Medical applications (SONAR, tumor, mass and various disease detection, medical diagnosis, medical data, and bio signal analysis)	PSO(4), GWO(1), WOA(1), ACO(1), MFO(3), GA(6)	[3,57,62,113,119,153,172–177,211]	16
Handwritten letter recognition, sentiment analysis, language processing, signature verification system, and text categorization	PSO(1), WOA(1), ACO(1), MFO(1), GA(6)	[5,53,89,105,129,149,154,178–180]	10
Hyper spectral images processing and classification.	PSO(3), GWO(2), GA(9)	[4,6,150,151,163,181–186,212]	13
Intrusion detection	PSO(1)	[61]	1
Protein and related genome annotation.	PSO(1), ACO(1)	[116,157]	2
Meteorology weather forecasting	ACO(1)	[127]	1
Biological application	GA(1)	[162]	1
Biochemistry and drug design	GA(2)	[104]	2
Electroencephalogram (EEG) signals application/Brain Computer Interface (BCI) system	ACO(1), GA(2)	[23,130,188–190]	5
Design an automatic FS model that can choose the most relevant features from password typing	GA(1)	[196]	1
Transient diagnosis system for nuclear power plants	GA(1)	[156]	1
Financial diagnosis /business crisis detection/stock price prediction	PSO(1), GA(3)	[191,192,202]	4
Diagnose different types of fault in a gearbox	GA(1)	[198]	1
Software Product Line (SPLs) and sw effort estimation	GA(2)	[22,106]	2
Theft detection in power distribution systems	CS(1)	[197]	1
Spam detection in emails	PSO(1), GA(2)	[85,193,194]	3
Transient diagnosis system for nuclear power plants	GA(1)	[156]	1
Automatic tomato disease detection system based	MFO(1)	[67]	1

Table 7. Summary of modifications applied on NIA-FS in the main applications.

App	Modification	Model	Classifier	Datasets	Dimension	Year	Ref.
Microarray gene expression	Hybridization	GA-SVM	SVM	2	2000	2003	[164]
	Hybridization	GA-MI	SVM	13	2000	2007	[128]
	Update mechanism	IPSO	1-KNN	11	15,009	2008	[136]
	Update mechanism	BPSO	1-KNN	6	10,509	2008	[92]
	Hybridization	IBPSO-IG, IBPSO-CFS	1-KNN	6	11,225	2008	[123]
	Update mechanism	cuPSO	1-KNN	11	15,009	2010	[137]
	Hybridization	GA-IG	1-KNN	11	15,009	2010	[18]
	Hybridization New operator/chaotic	TCBPSO-CFC	1-KNN	10	9868	2011	[52]
	Update mechanism	IBPSO	SVM	10	12,600	2011	[138]
	Hybridization	GA-NN-MI	NN-BP	11	7129	2011	[166]
	Hybridization	ACOFS	NN	9	2000	2012	[167]
	Hybridization	CS-MI-Entropy	ANN	6	15,009	2014	[126]
	Update mechanism	PSO-LSRG	1-KNN	5	12,600	2014	[24]
	New operator/XOR	HDBPSO	1-KNN, 3-KNN, 5-KNN	3	7129	2015	[98]
	New operator/binaryquantum	BQPSO-SVM	SVM	5	12,600	2016	[100]
	Hybridization	GWO-ALO	5-KNN	7	49151	2018	[4]
	Hybridization	GOA-SVM	SVM	3	17,678	2018	[168]
Medical application	Hybridization	GABPNN, GARBFNN, GALQVNN	BPNN, RBFNN, LQVNN	1	30	2006	[172]
	Hybridization	PSO-GA	SVM	3	7129	2008	[113]
	New fitness function	PSO-MLP	MLP	3	16	2008	[153]
	Hybridization	BPSO-FST, GA-FST	SVM	1	23	2010	[173]
	New operator/rough set	PSO-RR, PSO-QR	Naive Bayes, BayesNet, KStar	4	45	2014	[62]
	Hybridization	GAANNRP, GAANNLM, GAANNGD,	NNRP, NNLM, NNGD	1	10	2015	[174]
	Hybridization	PSO-KDE	KDE	2	32	2016	[175]
	Hybridization	IGWO-GA-KELM	KELM	2	32	2017	[119]
	New operator/chaotic maps	CMFOFS-KELM	KELM	1	22	2017	[3]

Table 7. *Cont.*

App	Modification	Model	Classifier	Datasets	Dimension	Year	Ref.
Hyper spectral images processing	New fitness function hybridization	GA-MDLP-Bayesian	Bayesian	3	20	2003	[182]
	New fitness function	GA-DFA	Validation by projection	1	882	2004	[183]
	Hybridization	GA-SVM	SVM	1	198	2008	[184]
	Different encoding schemes	GA-KNN	KNN	1	25	2008	[150]
	Different encoding schemes	GA-KNN	KNN	1	25	2008	[212]
	Hybridization parallelism	Parallel-GA-Adaboost	Adaboost ensemble, KNN	1	25	2010	[151]
	Different encoding schemes	BGAFS, BCGAFS	Adaboost ensemble, KNN	1	25	2010	[151]
	Hybridization	MI-GA-SVM-BB	SVM,BB	2	202	2011	[185]
	Parallelism	FODPSO-SVM	SVM	2	220	2015	[163]
	Hybridization	HGAPSO-SVM	SVM	1	220	2015	[186]
	New fitness function	GWO-KNN	7-KNN	3	224	2016	[6]
	Hybridization	GWO-ALO	5-KNN	5	10,304	2018	[4]
Arabic HR	New operator/local search	HGA	1-KNN	1	16	2004	[89]
	Different encoding schemes	GA	No classifier	1	100	2007	[149]
	Update mechanism	ACO	KNN	1	7542	2009	[179]
	New population structure	CAGA	NN-BP	1	16	2009	[105]
	Hybridization	IG-GA-PCA	KNN, C4.5	4	7542	2011	[129]
	New fitness function	GA-FLD	KNN, MLP, SVM (RBF, Poly, Sigm)	4	780	2014	[154]
	New initialization+New operator/mutation	BPSO	No classifier/clustering problem	3	8830	2016	[53]
	Hybridization new operator/DE evolutionary	IWOA-IG-SVM	SVM	4	8057	2018	[5]
Face recognition	Update mechanism	ACO	KNN	1	400	2007	[25]
	Update mechanism	ACO	KNN	1	400	2008	[135]
	New operator/intelligent acceleration	ABPSO	Euclidean classifier	2	2204	2012	[76]
	New operator/threshold	BPSO	Euclidean classifier	7	16,380	2014	[97]
	Hybridization new operator update	PSO-mGA	NN-BP,SVM-RBF,ensembles	2	1280	2017	[101]
	Hybridization	GWO-NN	NN	2	486	2018	[169]
	New operators/return-cost	Rc-BBFA	1-KNN	10	1280	2016	[112]

8. Conclusions and Future Research Directions

In this study, a survey about modifications of NIAs for tackling the FS optimization problem is presented. The review is based on a solid theoretical, applied, and technical foundation. Three main research streams are identified in this review: Meta-heuristic optimization, feature selection, and modification on NIAs for tackling FS. This review aims to draw the map for researchers and guide them when creating new research in this area. This survey is based on 156 articles collected and studied on modifications of NIAs for solving the FS problem. The sources of the information search came mainly from six well-regarded scientific databases: Elsevier, Springer, Hindawi, ACM, World scientific, and IEEE. From the review, it can be seen that the NIAs algorithms have been extensively investigated over the past years to improve the FS problem. About 34 different operators were investigated. The most popular operator is chaotic maps. Hybridization is the most widely used modification technique. There are three types of hybridization: Integrating NIA with another NIA, integrating NIA with a classifier, and integrating NIA with a classifier. The most widely used hybridization is the one that integrates a classifier with the NIA. Microarray and medical applications are the dominated applications where most of the NIA-FS are modified and used. Despite the popularity of the NIAs-FS, there are still many areas that need further investigation:

- Until now, there are few works in the binary optimization field. Many new operators can be proposed to enhance the performance of binary optimizers in a binary space. This is an interesting research direction;
- The proposed enhanced binary versions of optimizers can be used as a data mining tool in various applications. There are some applications where the usage of modified NIAs-FS in them is still limited;
- It would also be interesting to look at the dimensionality and number of instances in data sets. Nowadays, the majority of FS works to address problems with dimensionality up to several thousand but the question that may arise is what will happen if the data sets scaled up to millions of features? There is a scalability gap that should be addressed in the future;
- There is still room for improvement through parallel NIAs-FS. This might be a fruitful direction for research;
- Hyper volume Pareto optimal dominance and many-objective optimization need further crucial investigation.

Based on the above trends, the size of the NIAs-FS research area can be recognized. Besides, it can be imagined that a thorough investigation and improvement of NIAs will improve the FS process in various high-dimensional areas. This review paper will be used to help researchers take an excellent view of the modification strategies in nature-inspired algorithms for tackling the feature selection problem.

Author Contributions: Conceptualization, R.A.K., I.A. and A.S.; methodology, R.A.K., I.A. and A.S.; formal analysis, R.A.K., I.A. and A.S.; resources, R.A.K.; validation, M.A.E., R.D. and T.K.; writing—original draft preparation, R.A.K., I.A. and A.S.; writing—review and editing, M.A.E., R.D. and T.K.; supervision, M.A.E.; funding acquisition, T.K. All authors read and agreed to the published version of the manuscript.

Funding: No funding received for this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

Conflicts of Interest: All authors declare that they have no conflict of interest.

References

1. Liu, H.; Yu, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 491–502.
2. Hawkins, D.M. The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1–12. [CrossRef] [PubMed]
3. Wang, M.; Chen, H.; Yang, B.; Zhao, X.; Hu, L.; Cai, Z.; Huang, H.; Tong, C. Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing* **2017**, *267*, 69–84. [CrossRef]
4. Zawbaa, H.M.; Emary, E.; Grosan, C.; Snasel, V. Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach. *Swarm Evol. Comput.* **2018**, *42*, 29–42. [CrossRef]
5. Tubishat, M.; Abushariah, M.A.; Idris, N.; Aljarah, I. Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Appl. Intell.* **2018**, *49*, 1–20. [CrossRef]
6. Medjahed, S.A.; Saadi, T.A.; Benyettou, A.; Ouali, M. Gray wolf optimizer for hyperspectral band selection. *Appl. Soft Comput.* **2016**, *40*, 178–186. [CrossRef]
7. Makhadmeh, S.N.; Al-Betar, M.A.; Alyasseri, Z.A.A.; Abasi, A.K.; Khader, A.T.; Damaševičius, R.; Mohammed, M.A.; Abdulkareem, K.H. Smart home battery for the multi-objective power scheduling problem in a smart home using grey wolf optimizer. *Electronics* **2021**, *10*, 447. [CrossRef]
8. Faris, H.; Abukhurma, R.; Almanaseer, W.; Saadeh, M.; Mora, A.M.; Castillo, P.A.; Aljarah, I. Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: A case from the Spanish market. *Prog. Artif. Intell.* **2019**, *9*, 1–23. [CrossRef]
9. Al-Madi, N.; Faris, H.; Abukhurma, R. Cost-Sensitive Genetic Programming for Churn Prediction and Identification of the Influencing Factors in Telecommunication Market. *Int. J. Adv. Sci. Technol.* **2018**, 13–28.
10. Pourzangbar, A.; Losada, M.A.; Saber, A.; Ahari, L.R.; Larroudé, P.; Vaezi, M.; Brocchini, M. Prediction of non-breaking wave induced scour depth at the trunk section of breakwaters using Genetic Programming and Artificial Neural Networks. *Coast. Eng.* **2017**, *121*, 107–118. [CrossRef]
11. Okewu, E.; Misra, S.; Maskeliunas, R.; Damasevicius, R.; Fernandez-Sanz, L. Optimizing green computing awareness for environmental sustainability and economic security as a stochastic optimization problem. *Sustainability* **2017**, *9*, 1857. [CrossRef]
12. Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A survey on evolutionary computation approaches to feature selection. *IEEE Trans. Evol. Comput.* **2016**, *20*, 606–626. [CrossRef]
13. Alweshah, M.; Al Khalaileh, S.; Gupta, B.B.; Almomani, A.; Hammouri, A.I.; Al-Betar, M.A. The monarch butterfly optimization algorithm for solving feature selection problems. *Neural Comput. Appl.* **2020**, *2020*, 1–15. [CrossRef]
14. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [CrossRef]
15. Dash, M.; Liu, H. Feature selection for classification. *Intell. Data Anal.* **1997**, *1*, 131–156. [CrossRef]
16. Tabakhi, S.; Moradi, P.; Akhlaghian, F. An unsupervised feature selection algorithm based on ant colony optimization. *Eng. Appl. Artif. Intell.* **2014**, *32*, 112–123. [CrossRef]
17. Liu, Y.; Wang, G.; Chen, H.; Dong, H.; Zhu, X.; Wang, S. An improved particle swarm optimization for feature selection. *J. Bionic Eng.* **2011**, *8*, 191–200. [CrossRef]
18. Yang, C.H.; Chuang, L.Y.; Yang, C.H. IG-GA: A hybrid filter/wrapper method for feature selection of microarray data. *J. Med Biol. Eng.* **2010**, *30*, 23–28.
19. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature selection: A data perspective. *ACM Comput. Surv. (CSUR)* **2018**, *50*, 94. [CrossRef]
20. Brezočnik, L.; Fister, I.; Podgorelec, V. Swarm Intelligence Algorithms for Feature Selection: A Review. *Appl. Sci.* **2018**, *8*, 1521. [CrossRef]
21. Diao, R.; Shen, Q. Nature inspired feature selection meta-heuristics. *Artif. Intell. Rev.* **2015**, *44*, 311–340. [CrossRef]
22. Oliveira, A.L.; Braga, P.L.; Lima, R.M.; Cornélio, M.L. GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Inf. Softw. Technol.* **2010**, *52*, 1155–1166. [CrossRef]
23. Khushaba, R.N.; Al-Ani, A.; AlSukker, A.; Al-Jumaily, A. A combined ant colony and differential evolution feature selection algorithm. In *Proceedings of the International Conference on Ant Colony Optimization and Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–12.
24. Tran, B.; Xue, B.; Zhang, M. Improved PSO for feature selection on high-dimensional datasets. In *Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 503–515.
25. Kanan, H.R.; Faez, K.; Taheri, S.M. Feature selection using ant colony optimization (ACO): A new method and comparative study in the application of face recognition system. In *Proceedings of the Industrial Conference on Data Mining*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 63–76.
26. Hancer, E.; Xue, B.; Karaboga, D.; Zhang, M. A binary ABC algorithm based on advanced similarity scheme for feature selection. *Appl. Soft Comput.* **2015**, *36*, 334–348. [CrossRef]
27. Balázs, K.; Botzheim, J.; Kóczy, L.T. Comparative Investigation of Various Evolutionary and Memetic Algorithms. In *Computational Intelligence in Engineering*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 129–140. [CrossRef]
28. Sahlol, A.T.; Elaziz, M.A.; Jamal, A.T.; Damaševičius, R.; Hassan, O.F. A novel method for detection of tuberculosis in chest radiographs using artificial ecosystem-based optimisation of deep neural network features. *Symmetry* **2020**, *12, 1146* . [CrossRef]

29. Sahlol, A.T.; Yousri, D.; Ewees, A.A.; Al-qaness, M.A.A.; Damasevicius, R.; Elaziz, M.A. COVID-19 image classification using deep features and fractional-order marine predators algorithm. *Sci. Rep.* **2020**, *10*, 15–36. [CrossRef]
30. Polap, D.; Woźniak, M. Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* **2017**, *9*, 203. [CrossRef]
31. Połap, D.; Woźniak, M. Red fox optimization algorithm. *Expert Syst. Appl.* **2021**, *166*, 114107. [CrossRef]
32. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
33. Jouhari, H.; Lei, D.; Al-qaness, M.A.A.; Abd Elaziz, M.; Damaševičius, R.; Korytkowski, M.; Ewees, A.A. Modified Harris Hawks optimizer for solving machine scheduling problems. *Symmetry* **2020**, *12*, 1460. [CrossRef]
34. Ksiazek, K.; Połap, D.; Woźniak, M.; Damaševičius, R. Radiation heat transfer optimization by the use of modified ant lion optimizer. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence, Honolulu, HI, USA, 27 November–1 December 2017; Volume 2018, pp. 1–7.
35. Damaševičius, R.; Maskeliūnas, R. Agent state flipping based hybridization of heuristic optimization algorithms: A case of bat algorithm and krill herd hybrid algorithm. *Algorithms* **2021**, *14*, 358. [CrossRef]
36. Faris, H.; Aljarah, I.; Al-Betar, M.A.; Mirjalili, S. Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl.* **2018**, *30*, 413–435. [CrossRef]
37. Hira, Z.; Gillies, D. A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Adv. Bioinform.* **2015**, *2015*, 198363. [CrossRef] [PubMed]
38. Somol, P.; Grim, J.; Novovičová, J.; Pudil, P. Improving feature selection process resistance to failures caused by curse-of-dimensionality effects. *Kybernetika* **2011**, *47*, 401–425.
39. von Luxburg, U.; Bousquet, O. Distance-Based Classification with Lipschitz Functions. In *Learning Theory and Kernel Machines*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 314–328. [CrossRef]
40. Higgins, I.; Amos, D.; Pfau, D.; Racaniere, S.; Matthey, L.; Rezende, D.; Lerchner, A. Towards a Definition of Disentangled Representations. *arXiv* **2018**, arXiv:cs.LG/1812.02230.
41. Brank, J.; Mladenić, D.; Grobelnik, M.; Liu, H.; Mladenić, D.; Flach, P.A.; Garriga, G.C.; Toivonen, H.; Toivonen, H. Feature Selection. In *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 2011; pp. 402–406. [CrossRef]
42. Achille, A.; Soatto, S. Emergence of Invariance and Disentanglement in Deep Representations. In Proceedings of the 2018 Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 11–16 February 2018. [CrossRef]
43. Ratner, A.J.; Ehrenberg, H.R.; Hussain, Z.; Dunnmon, J.; Ré, C. Learning to Compose Domain-Specific Transformations for Data Augmentation. *arXiv* **2017**, arXiv:stat.ML/1709.01643.
44. Khaire, U.M.; Dhanalakshmi, R. Stability of feature selection algorithm: A review. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *2019*. [CrossRef]
45. Gheyas, I.A.; Smith, L.S. Feature subset selection in large dimensionality domains. *Pattern Recognit.* **2010**, *43*, 5–13. [CrossRef]
46. Mafarja, M.M.; Mirjalili, S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [CrossRef]
47. Glover, F. Tabu search-part I. *ORSA J. Comput.* **1989**, *1*, 190–206. [CrossRef]
48. Mirjalili, S.Z.; Mirjalili, S.; Saremi, S.; Faris, H.; Aljarah, I. Grasshopper optimization algorithm for multi-objective optimization problems. *Appl. Intell.* **2018**, *48*, 805–820. [CrossRef]
49. Mirjalili, S.; Lewis, A.; Mostaghim, S. Confidence measure: A novel metric for robust meta-heuristic optimisation algorithms. *Inf. Sci.* **2015**, *317*, 114–142. [CrossRef]
50. Lu, H.; Wang, X.; Fei, Z.; Qiu, M. The effects of using Chaotic map on improving the performance of multiobjective evolutionary algorithms. *Math. Probl. Eng.* **2014**, *2014*, 924652. [CrossRef]
51. Chuang, L.Y.; Yang, C.H.; Li, J.C. Chaotic maps based on binary particle swarm optimization for feature selection. *Appl. Soft Comput.* **2011**, *11*, 239–248. [CrossRef]
52. Chuang, L.Y.; Yang, C.S.; Wu, K.C.; Yang, C.H. Gene selection and classification using Taguchi chaotic binary particle swarm optimization. *Expert Syst. Appl.* **2011**, *38*, 13367–13377. [CrossRef]
53. Bharti, K.K.; Singh, P.K. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. *Appl. Soft Comput.* **2016**, *43*, 20–34. [CrossRef]
54. Ahmed, S.; Mafarja, M.; Faris, H.; Aljarah, I. Feature selection using salp swarm algorithm with chaos. In *Proceedings of the 2nd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*; ACM: New York, NY, USA, 2018; pp. 65–69.
55. Sayed, G.I.; Khoriba, G.; Haggag, M.H. A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl. Intell.* **2018**, *48*, 1–20. [CrossRef]
56. Ewees, A.A.; El Aziz, M.A.; Hassanien, A.E. Chaotic multi-verse optimizer-based feature selection. *Neural Comput. Appl.* **2017**, *31*, 991–1006. [CrossRef]
57. Sayed, G.I.; Darwish, A.; Hassanien, A.E. A New Chaotic Whale Optimization Algorithm for Features Selection. *J. Classif.* **2018**, *35*, 300–344. [CrossRef]
58. Sayed, G.I.; Hassanien, A.E.; Azar, A.T. Feature selection via a novel chaotic crow search algorithm. *Neural Comput. Appl.* **2017**, *31*, 171–188. [CrossRef]
59. Qasim, O.S.; Al-Thanoon, N.A.; Algamal, Z.Y. Feature selection based on chaotic binary black hole algorithm for data classification. *Chemom. Intell. Lab. Syst.* **2020**, *15*, 104104. [CrossRef]

60. Pawlak, Z. Rough sets. *Int. J. Comput. Inf. Sci.* **1982**, *11*, 341–356. [CrossRef]
61. Zainal, A.; Maarof, M.A.; Shamsuddin, S.M. Feature selection using Rough-DPSO in anomaly intrusion detection. In *Proceedings of the International Conference on Computational Science and Its Applications*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 512–524.
62. Inbarani, H.H.; Azar, A.T.; Jothi, G. Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Comput. Methods Programs Biomed.* **2014**, *113*, 175–185. [CrossRef]
63. Alia, A.F.; Taweel, A. Feature selection based on hybrid Binary Cuckoo Search and rough set theory in classification for nominal datasets. *Algorithms* **2017**, *14*, 65. [CrossRef]
64. El Aziz, M.A.; Hassanien, A.E. Modified cuckoo search algorithm with rough sets for feature selection. *Neural Comput. Appl.* **2018**, *29*, 925–934. [CrossRef]
65. Mafarja, M.M.; Mirjalili, S. Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. *Soft Comput.* **2018**, *23*, 6249–6265. [CrossRef]
66. Chen, Y.; Zhu, Q.; Xu, H. Finding rough set reducts with fish swarm algorithm. *Knowl.-Based Syst.* **2015**, *81*, 22–29. [CrossRef]
67. Hassanien, A.E.; Gaber, T.; Mokhtar, U.; Hefny, H. An improved moth flame optimization algorithm based on rough sets for tomato diseases detection. *Comput. Electron. Agric.* **2017**, *136*, 86–96. [CrossRef]
68. Tawhid, M.A.; Ibrahim, A.M. Hybrid binary particle swarm optimization and flower pollination algorithm based on rough set approach for feature selection problem. In *Nature-Inspired Computation in Data Mining and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 249–273.
69. Ropiak, K.; Artiemjew, P. On a Hybridization of Deep Learning and Rough Set Based Granular Computing. *Algorithms* **2020**, *13*, 63. [CrossRef]
70. Dennett, D.C. Darwinś dangerous idea. *Science* **1995**, *35*, 34–40.
71. Tanaka, M.; Watanabe, H.; Furukawa, Y.; Tanino, T. GA-based decision support system for multicriteria optimization. Systems, Man and Cybernetics. In *Proceedings of the Intelligent Systems for the 21st Century*; Springer: Berlin/Heidelberg, Germany, 1995; Volume 2, pp. 1556–1561.
72. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Hammouri, A.I.; Faris, H.; AlaḾ, A.Z.; Mirjalili, S. Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowl.-Based Syst.* **2018**, *145*, 25–45. [CrossRef]
73. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **2018**, *62*, 441–453. [CrossRef]
74. Khushaba, R.N.; Al-Ani, A.; Al-Jumaily, A. Feature subset selection using differential evolution and a statistical repair mechanism. *Expert Syst. Appl.* **2011**, *38*, 11515–11526. [CrossRef]
75. Leibowitz, N.; Baum, B.; Enden, G.; Karniel, A. The exponential learning equation as a function of successful trials results in sigmoid performance. *J. Math. Psychol.* **2010**, *54*, 338–340. [CrossRef]
76. Aneesh, M.; Masand, A.A.; Manikantan, K. Optimal feature selection based on image pre-processing using accelerated binary particle swarm optimization for enhanced face recognition. *Procedia Eng.* **2012**, *30*, 750–758. [CrossRef]
77. Pereira, L.; Rodrigues, D.; Almeida, T.; Ramos, C.; Souza, A.; Yang, X.S.; Papa, J. A binary cuckoo search and its application for feature selection. In *Cuckoo Search and Firefly Algorithm*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 141–154.
78. Rodrigues, D.; Pereira, L.A.; Nakamura, R.Y.; Costa, K.A.; Yang, X.S.; Souza, A.N.; Papa, J.P. A wrapper approach for feature selection based on bat algorithm and optimum-path forest. *Expert Syst. Appl.* **2014**, *41*, 2250–2258. [CrossRef]
79. Mafarja, M.; Aljarah, I.; Faris, H.; Hammouri, A.I.; AlaḾ, A.Z.; Mirjalili, S. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst. Appl.* **2019**, *117*, 267–286. [CrossRef]
80. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [CrossRef]
81. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. Systems, Man, and Cybernetics. In Proceedings of the Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108.
82. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; AlaḾ, A.Z.; Mirjalili, S.; Fujita, H. An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowl.-Based Syst.* **2018**, *154*, 43–67. [CrossRef]
83. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl.-Based Syst.* **2018**, *161*, 185–204. [CrossRef]
84. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [CrossRef]
85. Zhang, Y.; Wang, S.; Phillips, P.; Ji, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowl.-Based Syst.* **2014**, *64*, 22–31. [CrossRef]
86. Sihwail, R.; Omar, K.; Ariffin, K.A.Z.; Tubishat, M. Improved Harris Hawks Optimization Using Elite Opposition-Based Learning and Novel Search Mechanism for Feature Selection. *IEEE Access* **2020**, *8*, 121127–121145. [CrossRef]
87. Sims, D.W.; Humphries, N.E.; Bradford, R.W.; Bruce, B.D. Lévy flight and Brownian search patterns of a free-ranging predator reflect different prey field characteristics. *J. Anim. Ecol.* **2012**, *81*, 432–442. [CrossRef]
88. Khurmaa, R.A.; Aljarah, I.; Sharieh, A. An intelligent feature selection approach based on moth flame optimization for medical diagnosis. *Neural Comput. Appl.* **2020**, *33*, 7165–7204. [CrossRef]
89. Oh, I.S.; Lee, J.S.; Moon, B.R. Hybrid genetic algorithms for feature selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1424–1437. [PubMed]

90. Vieira, S.M.; Mendonça, L.F.; Farinha, G.J.; Sousa, J.M. Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. *Appl. Soft Comput.* **2013**, *13*, 3494–3504. [CrossRef]
91. Zhang, Y.; Gong, D.; Hu, Y.; Zhang, W. Feature selection algorithm based on bare bones particle swarm optimization. *Neurocomputing* **2015**, *148*, 150–157. [CrossRef]
92. Yang, C.S.; Chuang, L.Y.; Ke, C.H.; Yang, C.H. Boolean binary particle swarm optimization for feature selection. In Proceedings of the Evolutionary Computation 2008, Hong Kong, China, 1–6 June 2008; pp. 2093–2098.
93. Danyadi, Z.; Foldesi, P.; Koczy, L.T. Solution of a fuzzy resource allocation problem by various evolutionary approaches. In Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), Edmonton, AB, Canada, 24–28 June 2013. [CrossRef]
94. Chuang, L.Y.; Tsai, S.W.; Yang, C.H. Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Syst. Appl.* **2011**, *38*, 12699–12707. [CrossRef]
95. Xue, B.; Zhang, M.; Browne, W.N. Single feature ranking and binary particle swarm optimisation based feature subset ranking for feature selection. In *Proceedings of the Thirty-fifth Australasian Computer Science Conference*; Australian Computer Society, Inc.: Darlinghurst, Australia, 2012; Volume 122, pp. 27–36.
96. Lane, M.C.; Xue, B.; Liu, I.; Zhang, M. Gaussian based particle swarm optimisation and statistical clustering for feature selection. In *Proceedings of the European Conference on Evolutionary Computation in Combinatorial Optimization*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 133–144.
97. Krisshna, N.A.; Deepak, V.K.; Manikantan, K.; Ramachandran, S. Face recognition using transform domain feature extraction and PSO-based feature selection. *Appl. Soft Comput.* **2014**, *22*, 141–161. [CrossRef]
98. Banka, H.; Dara, S. A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. *Pattern Recognit. Lett.* **2015**, *52*, 94–100. [CrossRef]
99. Moradi, P.; Gholampour, M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Appl. Soft Comput.* **2016**, *43*, 117–130. [CrossRef]
100. Xi, M.; Sun, J.; Liu, L.; Fan, F.; Wu, X. Cancer Feature Selection and Classification Using a Binary Quantum-Behaved Particle Swarm Optimization and Support Vector Machine. *Comput. Math. Methods Med.* **2016**, *2016*, 3572705. [CrossRef] [PubMed]
101. Mistry, K.; Zhang, L.; Neoh, S.C.; Lim, C.P.; Fielding, B. A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition. *IEEE Trans. Cybern.* **2017**, *47*, 1496–1509. [CrossRef] [PubMed]
102. Zhang, Y.; Song, X.F.; Gong, D.W. A return-cost-based binary firefly algorithm for feature selection. *Inf. Sci.* **2017**, *418*, 561–574. [CrossRef]
103. Lin, K.C.; Hung, J.C.; Wei, J.T. Feature selection with modified lionś algorithms and support vector machine for high-dimensional data. *Appl. Soft Comput.* **2018**, *68*, 669–676. [CrossRef]
104. Salcedo-Sanz, S.; Prado-Cumplido, M.; Pérez-Cruz, F.; Bousoño-Calzón, C. Feature selection via genetic optimization. In *Proceedings of the International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 547–552.
105. Zeng, X.P.; Li, Y.M.; Qin, J. A dynamic chain-like agent genetic algorithm for global numerical optimization and feature selection. *Neurocomputing* **2009**, *72*, 1214–1228. [CrossRef]
106. Guo, J.; White, J.; Wang, G.; Li, J.; Wang, Y. A genetic algorithm for optimized feature selection with resource constraints in software product lines. *J. Syst. Softw.* **2011**, *84*, 2208–2221. [CrossRef]
107. Blum, C.; Puchinger, J.; Raidl, G.R.; Roli, A. Hybrid metaheuristics in combinatorial optimization: A survey. *Appl. Soft Comput.* **2011**, *11*, 4135–4151. [CrossRef]
108. Sayed, S.A.F.; Nabil, E.; Badr, A. A binary clonal flower pollination algorithm for feature selection. *Pattern Recognit. Lett.* **2016**, *77*, 21–27. [CrossRef]
109. Alweshah, M.; Alkhalaileh, S.; Albashish, D.; Mafarja, M.; Bsoul, Q.; Dorgham, O. A hybrid mine blast algorithm for feature selection problems. *Soft Comput.* **2020**, *25*, 17–534. [CrossRef]
110. Ibrahim, R.A.; Ewees, A.A.; Oliva, D.; Elaziz, M.A.; Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3155–3169. [CrossRef]
111. Sayed, G.I.; Hassanien, A.E. A hybrid SA-MFO algorithm for function optimization and engineering design problems. *Complex Intell. Syst.* **2018**, *4*, 195–212. [CrossRef]
112. Zhang, L.; Mistry, K.; Neoh, S.C.; Lim, C.P. Intelligent facial emotion recognition using moth-firefly optimization. *Knowl.-Based Syst.* **2016**, *111*, 248–267. [CrossRef]
113. Li, S.; Wu, X.; Tan, M. Gene selection using hybrid particle swarm optimization and genetic algorithm. *Soft Comput.* **2008**, *12*, 1039–1048. [CrossRef]
114. Abdel-Basset, M.; Ding, W.; El-Shahat, D. A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. *Artif. Intell. Rev.* **2020**, *54*, 593–637. [CrossRef]
115. Ibrahim, A.M.; Tawhid, M.A. A New Hybrid Binary Algorithm of Bat Algorithm and Differential Evolution for Feature Selection and Classification. In *Applications of Bat Algorithm and Its Variants*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–18.
116. Nemati, S.; Basiri, M.E.; Ghasem-Aghaee, N.; Aghdam, M.H. A novel ACO–GA hybrid algorithm for feature selection in protein function prediction. *Expert Syst. Appl.* **2009**, *36*, 12086–12094. [CrossRef]
117. Wan, Y.; Wang, M.; Ye, Z.; Lai, X. A feature selection method based on modified binary coded ant colony optimization algorithm. *Appl. Soft Comput.* **2016**, *49*, 248–258. [CrossRef]

118. Mafarja, M.; Qasem, A.; Heidari, A.A.; Aljarah, I.; Faris, H.; Mirjalili, S. Efficient hybrid nature-inspired binary optimizers for feature selection. *Cogn. Comput.* **2020**, *12*, 150–175. [CrossRef]
119. Li, Q.; Chen, H.; Huang, H.; Zhao, X.; Cai, Z.; Tong, C.; Liu, W.; Tian, X. An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Comput. Math. Methods Med.* **2017**, *2017*, 9512741. [CrossRef]
120. Kihel, B.K.; Chouraqui, S. Firefly optimization using artificial immune system for feature subset selection. *Int. J. Intell. Eng. Syst* **2019**, *12*, 337–347. [CrossRef]
121. Faris, H.; Hassonah, M.A.; AlaḾ, A.Z.; Mirjalili, S.; Aljarah, I. A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. *Neural Comput. Appl.* **2018**, *30*, 2355–2369. [CrossRef]
122. Aljarah, I.; AlaḾ, A.Z.; Faris, H.; Hassonah, M.A.; Mirjalili, S.; Saadeh, H. Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. *Cogn. Comput.* **2018**, *10*, 478–495. [CrossRef]
123. Yang, C.S.; Chuang, L.Y.; Ke, C.H.; Yang, C.H. A Hybrid Feature Selection Method for Microarray Classification. *IAENG Int. J. Comput. Sci.* **2008**, *35*, 285–290.
124. Unler, A.; Murat, A.; Chinnam, R.B. mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Inf. Sci.* **2011**, *181*, 4625–4641. [CrossRef]
125. Cervante, L.; Xue, B.; Zhang, M.; Shang, L. Binary particle swarm optimisation for feature selection: A filter based approach. In Proceedings of the Evolutionary Computation (CEC), Brisbane, Australia, 10–15 June 2012; pp. 1–8.
126. Moghadasian, M.; Hosseini, S.P. Binary cuckoo optimization algorithm for feature selection in high-dimensional datasets. In Proceedings of the International Conference on Innovative Engineering Technologies (ICIET2́014), Bangkok, Thailand, 28–29 December 2014; pp. 18–21.
127. Zhang, C.K.; Hu, H. Feature selection using the hybrid of ant colony optimization and mutual information for the forecaster. Machine Learning and Cybernetics. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 3, pp. 1728–1732.
128. Huang, J.; Cai, Y.; Xu, X. A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognit. Lett.* **2007**, *28*, 1825–1844. [CrossRef]
129. Uğuz, H. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowl.-Based Syst.* **2011**, *24*, 1024–1032. [CrossRef]
130. Wang, L.; Xu, G.; Wang, J.; Yang, S.; Guo, L.; Yan, W. GA-SVM based feature selection and parameters optimization for BCI research. Natural Computation (ICNC). In Proceedings of the 2011 Seventh International Conference on Natural Computation, Shanghai, China, 26–28 July 2011; Volume 1, pp. 580–583.
131. Zainudin, M.; Sulaiman, M.; Mustapha, N.; Perumal, T.; Nazri, A.; Mohamed, R.; Manaf, S. Feature selection optimization using hybrid relief-f with self-adaptive differential evolution. *Int. J. Intell. Eng. Syst* **2017**, *10*, 21–29. [CrossRef]
132. Guha, R.; Ghosh, M.; Mutsuddi, S.; Sarkar, R.; Mirjalili, S. Embedded chaotic whale survival algorithm for filter-wrapper feature selection. *arXiv* **2020**, arXiv:2005.04593.
133. Hassonah, M.A.; Al-Sayyed, R.; Rodan, A.; AlaḾ, A.Z.; Aljarah, I.; Faris, H. An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowl.-Based Syst.* **2020**, *192*, 105353. [CrossRef]
134. Ahmed, N.; Rafiq, J.I.; Islam, M.R. Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors* **2020**, *20*, 317. [CrossRef]
135. Kanan, H.R.; Faez, K. An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. *Appl. Math. Comput.* **2008**, *205*, 716–725. [CrossRef]
136. Chuang, L.Y.; Chang, H.W.; Tu, C.J.; Yang, C.H. Improved binary PSO for feature selection using gene expression data. *Comput. Biol. Chem.* **2008**, *32*, 29–38. [CrossRef] [PubMed]
137. Martinez, E.; Alvarez, M.M.; Trevino, V. Compact cancer biomarkers discovery using a swarm intelligence feature selection algorithm. *Comput. Biol. Chem.* **2010**, *34*, 244–250. [CrossRef] [PubMed]
138. Mohamad, M.S.; Omatu, S.; Deris, S.; Yoshioka, M. A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data. *IEEE Trans. Inf. Technol. Biomed.* **2011**, *15*, 813–822. [CrossRef]
139. Xue, B.; Zhang, M.; Browne, W.N. Novel initialisation and updating mechanisms in PSO for feature selection in classification. In *Proceedings of the European Conference on the Applications of Evolutionary Computation*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 428–438.
140. Mafarja, M.; Sabar, N.R. Rank based binary particle swarm optimisation for feature selection in classification. In Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, Amman, Jordan, 26–27 June 2018; pp. 1–6.
141. Mafarja, M.; Jarrar, R.; Ahmad, S.; Abusnaina, A.A. Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*; ACM: New York, NY, USA, 2018; p. 18.
142. Aljarah, I.; Mafarja, M.; Heidari, A.A.; Faris, H.; Zhang, Y.; Mirjalili, S. Asynchronous accelerating multi-leader salp chains for feature selection. *Appl. Soft Comput.* **2018**, *71*, 964–979. [CrossRef]
143. Hammouri, A.I.; Mafarja, M.; Al-Betar, M.A.; Awadallah, M.A.; Abu-Doush, I. An improved Dragonfly Algorithm for feature selection. *Knowl.-Based Syst.* **2020**, *203*, 106131. [CrossRef]

144. Faris, H.; Heidari, A.A.; AlaḾ, A.Z.; Mafarja, M.; Aljarah, I.; Eshtay, M.; Mirjalili, S. Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Syst. Appl.* **2020**, *140*, 112898. [CrossRef]
145. Ouadfel, S.; Abd Elaziz, M. Enhanced Crow Search Algorithm for Feature Selection. *Expert Syst. Appl.* **2020**, *159*, 113572. [CrossRef]
146. Gholami, J.; Pourpanah, F.; Wang, X. Feature selection based on improved binary global harmony search for data classification. *Appl. Soft Comput.* **2020**, *93*, 106402. [CrossRef]
147. Khurma, R.A.; Aljarah, I.; Sharieh, A. Rank based moth flame optimisation for feature selection in the medical application. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.
148. Khurma, R.A.; Castillo, P.A.; Sharieh, A.; Aljarah, I. Feature Selection using Binary Moth Flame Optimization with Time Varying Flames Strategies. In *Proceedings of the 12th International Joint Conference on Computational Intelligence*; SciTePress: Setubal, Portugal, 2020; pp. 17–27. [CrossRef]
149. Galbally, J.; Fierrez, J.; Freire, M.R.; Ortega-Garcia, J. Feature selection based on genetic algorithms for on-line signature verification. In Proceedings of the IEEE Workshop on Automatic Identification Advanced Technologies, Alghero, Italy, 7–8 June 2007; pp. 198–203.
150. Lu, J.; Zhao, T.; Zhang, Y. Feature selection based-on genetic algorithm for image annotation. *Knowl.-Based Syst.* **2008**, *21*, 887–891. [CrossRef]
151. Li, R.; Lu, J.; Zhang, Y.; Zhao, T. Dynamic Adaboost learning with feature selection based on parallel genetic algorithm for image annotation. *Knowl.-Based Syst.* **2010**, *23*, 195–201. [CrossRef]
152. Hans, R.; Kaur, H. Quasi-opposition-Based Multi-verse Optimization Algorithm for Feature Selection. In *Proceedings of the First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019)*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 345–359.
153. Chakraborty, B. Feature subset selection by particle swarm optimization with fuzzy fitness function. In Proceedings of the 2008 International Conference on Intelligent System and Knowledge Engineering, ISKE, Xiamen, China, 17–19 November 2008; Volume 1, pp. 1038–1042.
154. De Stefano, C.; Fontanella, F.; Marrocco, C.; Di Freca, A.S. A GA-based feature selection approach with an application to handwritten character recognition. *Pattern Recognit. Lett.* **2014**, *35*, 130–141. [CrossRef]
155. Khurma, R.A.; Castillo, P.A.; Sharieh, A.; Aljarah, I. New Fitness Functions in Binary Harris Hawks Optimization for Gene Selection in Microarray Datasets. In *Proceedings of the 12th International Joint Conference on Computational Intelligence*; SciTePress: Setubal, Portugal, 2020; Volume 1, pp. 139–146. [CrossRef]
156. Zio, E.; Baraldi, P.; Pedroni, N. Selecting features for nuclear transients classification by means of genetic algorithms. *IEEE Trans. Nucl. Sci.* **2006**, *53*, 1479–1493. [CrossRef]
157. Mandal, M.; Mukhopadhyay, A.; Maulik, U. Prediction of protein subcellular localization by incorporating multiobjective PSO-based feature subset selection into the general form of Chous PseAAC. *Med Biol. Eng. Comput.* **2015**, *53*, 331–344. [CrossRef]
158. Aljarah, I.; Habib, M.; Faris, H.; Al-Madi, N.; Heidari, A.A.; Mafarja, M.; Abd Elaziz, M.; Mirjalili, S. A Dynamic Locality Multi-Objective Salp Swarm Algorithm for Feature Selection. *Comput. Ind. Eng.* **2020**, *147*, 106628. [CrossRef]
159. Niu, B.; Yi, W.; Tan, L.; Geng, S.; Wang, H. A multi-objective feature selection method based on bacterial foraging optimization. *Nat. Comput.* **2019**, *120*, 63–76. [CrossRef]
160. Habib, M.; Aljarah, I.; Faris, H. A Modified Multi-objective Particle Swarm Optimizer-Based Lévy Flight: An Approach Toward Intrusion Detection in Internet of Things. *Arab. J. Sci. Eng.* **2020**, *45*, 6081–6108. [CrossRef]
161. Al-Tashi, Q.; Abdulkadir, S.J.; Rais, H.M.; Mirjalili, S.; Alhussian, H. Approaches to Multi-Objective Feature Selection: A Systematic Literature Review. *IEEE Access* **2020**, *8*, 125076–125096. [CrossRef]
162. Punch, W.F., III; Goodman, E.D.; Pei, M.; Chia-Shun, L.; Hovland, P.D.; Enbody, R.J. Further Research on Feature Selection and Classification Using Genetic Algorithms. *ICGA J.* **1993**, *93*, 557–564.
163. Ghamisi, P.; Couceiro, M.S.; Benediktsson, J.A. A novel feature selection approach based on FODPSO and SVM. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2935–2947. [CrossRef]
164. Frohlich, H.; Chapelle, O.; Scholkopf, B. Feature selection for support vector machines by means of genetic algorithm. In Proceedings of the 2003 15th IEEE International Conference, Sacramento, CA, USA, 3–5 November 2003; pp. 142–148.
165. Robbins, K.; Zhang, W.; Bertrand, J.; Rekaya, R. The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. *Math. Med. Biol. A J. IMA* **2007**, *24*, 413–426. [CrossRef] [PubMed]
166. Kabir, M.M.; Shahjahan, M.; Murase, K. A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing* **2011**, *74*, 2914–2928. [CrossRef]
167. Kabir, M.M.; Shahjahan, M.; Murase, K. A new hybrid ant colony optimization algorithm for feature selection. *Expert Syst. Appl.* **2012**, *39*, 3747–3763. [CrossRef]
168. Ibrahim, H.T.; Mazher, W.J.; Ucan, O.N.; Bayat, O. A grasshopper optimizer approach for feature selection and optimizing SVM parameters utilizing real biomedical data sets. *Neural Comput. Appl.* **2018**, *31*, 5965–5974. [CrossRef]
169. Sreedharan, N.P.N.; Ganesan, B.; Raveendran, R.; Sarala, P.; Dennis, B. Grey Wolf optimisation-based feature selection and classification for facial emotion recognition. *IET Biom.* **2018**, *7*, 490–499. [CrossRef]
170. Handels, H.; Roß, T.; Kreusch, J.; Wolff, H.H.; Poeppl, S.J. Feature selection for optimized skin tumor recognition using genetic algorithms. *Artif. Intell. Med.* **1999**, *16*, 283–297. [CrossRef]

171. Zheng, B.; Chang, Y.H.; Wang, X.H.; Good, W.F.; Gur, D. Feature selection for computerized mass detection in digitized mammograms by using a genetic algorithm. *Acad. Radiol.* **1999**, *6*, 327–332. [CrossRef]
172. Li, T.S. Feature selection for classification by using a GA-based neural network approach. *J. Chin. Inst. Ind. Eng.* **2006**, *23*, 55–64. [CrossRef]
173. Babaoglu, İ.; Findik, O.; Ülker, E. A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine. *Expert Syst. Appl.* **2010**, *37*, 3177–3183. [CrossRef]
174. Ahmad, F.; Isa, N.A.M.; Hussain, Z.; Osman, M.K.; Sulaiman, S.N. A GA-based feature selection and parameter optimization of an ANN in diagnosing breast cancer. *Pattern Anal. Appl.* **2015**, *18*, 861–870. [CrossRef]
175. Sheikhpour, R.; Sarram, M.A.; Sheikhpour, R. Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer. *Appl. Soft Comput.* **2016**, *40*, 113–131. [CrossRef]
176. Sayed, G.I.; Hassanien, A.E.; Nassef, T.M.; Pan, J.S. Alzheimerś Disease Diagnosis Based on Moth Flame Optimization. In *Proceedings of the International Conference on Genetic and Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 298–305.
177. Sayed, G.I.; Hassanien, A.E. Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images. *Appl. Intell.* **2017**, *47*, 397–408. [CrossRef]
178. Daelemans, W.; Hoste, V.; De Meulder, F.; Naudts, B. Combined optimization of feature selection and algorithm parameters in machine learning of language. In *European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 84–95.
179. Aghdam, M.H.; Ghasem-Aghaee, N.; Basiri, M.E. Text feature selection using ant colony optimization. *Expert Syst. Appl.* **2009**, *36*, 6843–6853. [CrossRef]
180. Ewees, A.A.; Sahlol, A.T.; Amasha, M.A. A Bio-inspired moth-flame optimization algorithm for Arabic handwritten letter recognition. Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO). In Proceedings of the 2017 International Conference, Brussels, Belgium, 13–17 July 2017; pp. 154–159.
181. Tackett, W.A. Genetic Programming for Feature Discovery and Image Discrimination. *ICGA J.* **1993**, *1993*, 303–311.
182. Bhanu, B.; Lin, Y. Genetic algorithm based feature selection for target detection in SAR images. *Image Vis. Comput.* **2003**, *21*, 591–608. [CrossRef]
183. Jarvis, R.M.; Goodacre, R. Genetic algorithm optimization for pre-processing and variable selection of spectroscopic data. *Bioinformatics* **2004**, *21*, 860–868. [CrossRef] [PubMed]
184. Zhuo, L.; Zheng, J.; Li, X.; Wang, F.; Ai, B.; Qian, J. A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine. In *Proceedings of the Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Classification of Remote Sensing Images*; International Society for Optics and Photonics: Bellingham, WA, USA, 2008; Volume 7147, p. 71471.
185. Li, S.; Wu, H.; Wan, D.; Zhu, J. An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine. *Knowl.-Based Syst.* **2011**, *24*, 40–48. [CrossRef]
186. Ghamisi, P.; Benediktsson, J.A. Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 309–313. [CrossRef]
187. Raymer, M.L.; Punch, W.F.; Goodman, E.D.; Kuhn, L.A.; Jain, A.K. Dimensionality reduction using genetic algorithms. *IEEE Trans. Evol. Comput.* **2000**, *4*, 164–171. [CrossRef]
188. Palaniappan, R.; Raveendran, P. Genetic Algorithm to select features for Fuzzy ARTMAP classification of evoked EEG. In Proceedings of the 2002 Asia-Pacific Conference, Taipei, Taiwan, China, 8 August 2002; Volume 2, pp. 53–56.
189. Garrett, D.; Peterson, D.A.; Anderson, C.W.; Thaut, M.H. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2003**, *11*, 141–144. [CrossRef]
190. Noori, F.M.; Qureshi, N.K.; Khan, R.A.; Naseer, N. Feature selection based on modified genetic algorithm for optimization of functional near-infrared spectroscopy (fNIRS) signals for BCI. In Proceedings of the 2016 2nd International Conference on IEEE, Chengdu, China, 26–27 October 2016; pp. 50–53.
191. Chen, L.H.; Hsiao, H.D. Feature selection to diagnose a business crisis by using a real GA-based support vector machine: An empirical study. *Expert Syst. Appl.* **2008**, *35*, 1145–1155. [CrossRef]
192. Huang, C.F.; Chang, B.R.; Cheng, D.W.; Chang, C.H. Feature Selection and Parameter Optimization of a Fuzzy-based Stock Selection Model Using Genetic Algorithms. *Int. J. Fuzzy Syst.* **2012**, *14*, 1.
193. Temitayo, F.; Stephen, O.; Abimbola, A. Hybrid GA-SVM for efficient feature selection in e-mail classification. *Comput. Eng. Intell. Syst.* **2012**, *3*, 17–28.
194. Faris, H.; AlaṀ, A.Z.; Heidari, A.A.; Aljarah, I.; Mafarja, M.; Hassonah, M.A.; Fujita, H. An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Inf. Fusion* **2019**, *48*, 67–83. [CrossRef]
195. Saidala, R.K. Variant of Northern Bald Ibis Algorithm for Unmasking Outliers. *Int. J. Softw. Sci. Comput. Intell. (IJSSCI)* **2020**, *12*, 15–29. [CrossRef]
196. Yu, E.; Cho, S. GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification. In Proceedings of the International Joint Conference on IEEE, Portland, OR, USA, 20–24 July 2003; Volume 3, pp. 2253–2257.

197. Rodrigues, D.; Pereira, L.A.; Almeida, T.; Papa, J.P.; Souza, A.; Ramos, C.C.; Yang, X.S. BCS: A binary cuckoo search algorithm for feature selection. In Proceedings of the 2013 IEEE International Symposium on IEEE, Beijing, China, 19–23 May 2013; pp. 465–468.
198. Hajnayeb, A.; Ghasemloonia, A.; Khadem, S.; Moradi, M. Application and comparison of an ANN-based feature selection method and the genetic algorithm in gearbox fault diagnosis. *Expert Syst. Appl.* **2011**, *38*, 10205–10209. [CrossRef]
199. Khurma, R.A.; Aljarah, I.; Sharieh, A.; Mirjalili, S. EvoloPy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 131–173.
200. Faris, H.; Aljarah, I.; Mirjalili, S.; Castillo, P.A.; Guervós, J.J.M. EvoloPy: An Open-source Nature-inspired Optimization Framework in Python. *IJCCI (ECTA)* **2016**, *1*, 171–177.
201. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
202. Unler, A.; Murat, A. A discrete particle swarm optimization method for feature selection in binary classification problems. *Eur. J. Oper. Res.* **2010**, *206*, 528–539. [CrossRef]
203. Nakamura, R.Y.; Pereira, L.A.; Costa, K.; Rodrigues, D.; Papa, J.P.; Yang, X.S. BBA: A binary bat algorithm for feature selection. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 22–25 August 2012; pp. 291–297.
204. Taha, A.M.; Mustapha, A.; Chen, S.D. Naive bayes-guided bat algorithm for feature selection. *Sci. World J.* **2013**, *2013*, 325973. [CrossRef]
205. Huang, C.L. ACO-based hybrid classification system with feature subset selection and model parameters optimization. *Neurocomputing* **2009**, *73*, 438–448. [CrossRef]
206. Lin, K.C.; Chien, H.Y. CSO-based feature selection and parameter optimization for support vector machine. In Proceedings of the 2009 Joint Conferences on IEEE, Tamsui, Taiwan, China, 3–5 December 2009; pp. 783–788.
207. ElAlami, M.E. A filter model for feature subset selection based on genetic algorithm. *Knowl.-Based Syst.* **2009**, *22*, 356–362. [CrossRef]
208. Huang, C.L.; Wang, C.J. A GA-based feature selection and parameters optimizationfor support vector machines. *Expert Syst. Appl.* **2006**, *31*, 231–240. [CrossRef]
209. Zhang, Y.; Gong, D.W.; Cheng, J. Multi-objective particle swarm optimization approach for cost-based feature selection in classification. *IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB)* **2017**, *14*, 64–75. [CrossRef] [PubMed]
210. Kashef, S.; Nezamabadi-pour, H. An advanced ACO algorithm for feature subset selection. *Neurocomputing* **2015**, *147*, 271–279. [CrossRef]
211. Kashef, S.; Nezamabadi-pour, H. A new feature selection algorithm based on binary ant colony optimization. In Proceedings of the 2013 5th Conference on IEEE, Kuala Lumpur, Malaysia, 4–5 December 2013; pp. 50–54.
212. Zhao, T.; Lu, J.; Zhang, Y.; Xiao, Q. Feature selection based on genetic algorithm for cbir. In Proceedings of the Image and Signal Processing, 2008. CISP'08, Congress on IEEE, Sanya, China, 27–30 May 2008; Volume 2, pp. 495–499.
213. Xue, B.; Zhang, M.; Browne, W.N. New fitness functions in binary particle swarm optimisation for feature selection. In Proceedings of the 2012 IEEE Congress on IEEE, Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–8.
214. Emary, E.; Zawbaa, H.M.; Ghany, K.K.A.; Hassanien, A.E.; Parv, B. Firefly optimization algorithm for feature selection. In Proceedings of the 7th Balkan Conference on Informatics Conference, ACM, Craiova, Romania, 2–4 September 2015; p. 26.
215. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Trans. Cybern.* **2013**, *43*, 1656–1671. [CrossRef]
216. Xue, B.; Cervante, L.; Shang, L.; Browne, W.N.; Zhang, M. Binary PSO and rough set theory for feature selection: A multi-objective filter based approach. *Int. J. Comput. Intell. Appl.* **2014**, *13*, 1450009. [CrossRef]
217. Doerner, K.; Gutjahr, W.J.; Hartl, R.F.; Strauss, C.; Stummer, C. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Ann. Oper. Res.* **2004**, *131*, 79–99. [CrossRef]