Journal of Theoretical and Applied Information Technology
15th January 2017. Vol.95. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OBKA-FS: AN OPPOSITIONAL-BASED BINARY KIDNEY-INSPIRED SEARCH ALGORITHM FOR FEATURE SELECTION

[1] MUSTAFA KADHIM TAQI, [2] ROSMAH ALI

[1,2] Advanced Informatics School, Universiti Teknologi Malaysia, 54100, Kuala Lumpur, Malaysia
E-mail: [1] ktmustafa2@live.utm.my, [2] rosmaha.kl@utm.my

ABSTRACT

Feature selection is a key step when building an automatic classification system. Numerous evolutionary algorithms applied to remove irrelevant features in order to make the classifier perform more accurate. Kidney-inspired search algorithm (KA) is a very modern evolutionary algorithm. The original version of KA performed more effectively compared with other evolutionary algorithms. However, KA was proposed for continuous search spaces. For feature subset selection and many optimization problems such as classification, binary discrete space is required. Moreover, the movement operator of solutions is notably affected by its own best-known solution found up to now, denoted as S_{best}. This may be inadequate if S_{best} is located near a local optimum as it will direct the search process to a suboptimal solution. In this study, a three-fold improvement in the existing KA is proposed. First, a binary version of the kidney-inspired algorithm (BKA-FS) for feature subset selection is introduced to improve classification accuracy in multi-class classification problems. Second, the proposed BKA-FS is integrated into an oppositional-based initialization method in order to start with good initial solutions. Thus, this improved algorithm denoted as OBKA-FS. Third, a novel movement strategy based on the calculation of mutual information (MI), which gives OBKA-FS the ability to work in a discrete binary environment has been proposed. For evaluation, an experiment was conducted using ten UCI machine learning benchmark instances. Results show that OBKA-FS outperforms the existing state-of-the-art evolutionary algorithms for feature selection. In particular, OBKA-FS obtained better accuracy with same or fewer features and higher dependency with less redundancy. Thus, the results confirm the high performance of the improved kidney-inspired algorithm in solving optimization problems such as feature selection.

Keywords: *Feature Selection, Kidney-Inspired Algorithm, Mutual Information, Oppositional-Learning*

1. INTRODUCTION

In various fields of study, ranging from pattern recognition [1], data mining [2], selection of microarray data gene [3], categorization of text [4] and retrieval of multimedia information [5-9], there is involvement of datasets that contain vast number of features. In these studies, feature selection becomes an indispensable procedure. As a result of the presence of noisy, inappropriate or ambiguous features, the aptitude of handling vague and uneven information in real life problems has become a key requirement in feature selection processes [10].

Feature selection can be defined as the procedure used to select subsets of features from an original one, and using the chosen subsets to form blocks of another dataset. A selected subset need to be relevant and adequate in describing the target models, thus retain a high accuracy that depict the features of the original set. The significance of feature selection is mainly to reduce the magnitude of a problem as well as the space used to learn various algorithms. During the process of designing a classifier, there is a potential to augment the speed and quality of a classification. This can be achieved by reducing the feature quantity used to describe a dataset to help in improving the performance of a learning algorithm and by maximizing the accuracy of classification.

The different feature selection algorithms often encompass four main aspects in determining the dimensions of the search process, that is, the starting point of a search space, organizing a search, feature subsets assessment strategy, and a criterion to be used to halt the search process. The method used in the search process is responsible for isolating potential entrant subsets and evaluating the appropriateness of a certain subset. There are three methods of evaluation: (i) filter methods, (ii) hybrid filter-wrapper methods and (iii) wrapper methods [8].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Filter-based methods use statistical information of data in selecting the features. In a given dataset, the algorithm begins to search a subset in the feature space using a set of search criteria. Any subset that is generated is assessed by an autonomous measure. The subset containing the most suitable evaluation measure is regarded as the best one. The search process continues until a predefined stop criterion is reached. Wrapper-based methods are dependent on classifiers and help in maximizing the accuracy of classification by a supervised technique. The two methods take a learning model and are favorites in classifying problems. Wrapper algorithms share similar aspects with filter algorithms, but the latter use a predefined mining algorithm instead of an independent measure during the evaluation process of a subset. While the filter method is efficient in nature in comparison to the wrapper method, its main weakness is that it may contain the initiative and figurative prejudices of the most fitting learning algorithm while constructing classifiers [11, 12]. Hybrid approaches assume the benefits of the two methods. Both filter-based and wrapper-based methods make use of independent measures when deciding the best subsets for any cardinality [8]. These methods use mining algorithm when selecting the final optimal subset selected from the best subsets from various cardinalities. Various research studies have expounded on conventional methods used in the field of feature selection [3, 8, 13].

A problem occurring during feature selection is similar to one taking place during search space optimization. Thus, feature selection method that bases its algorithm on stochastic search is commanding sizeable attention from researchers. Various approaches are being suggested on how to execute feature selection while utilizing evolutionary algorithms [14]. Some researchers proposed Genetic Algorithms (GA) whereas others propose conducting feature selection using binary Particle Swarm Optimization (PSO) [15-17]. Some made use of search technique called tabu in their problem solving method [18].

Kidney-inspired search algorithm (KA) is a new evolutionary optimization algorithm that derives its functionality from the kidney process in the body of a human being, and was initially introduced by [19]. When using the algorithm, the solutions are rated based on the average value of the objective functions of the solutions in a particular populace in a particular round. Optimal solutions are identified in the filtered blood and the rest are considered as inferior solutions. This process simulates the process of filtration known as glomerular in the human kidney. The inferior solutions once again are considered during other reiterations, and if they don't satisfy the filtration rate after the application of a set of movement operators, they are ejected from the set of solutions. This also stimulates the reabsorption and secretion features of a kidney. Additionally, a solution termed as the optimal solution is expelled if it does not prove to be better than the solutions classified in the worst sets; this simulates the blood secretion process by the kidney. After placing each of the solutions in a set, the optimal solutions are ranked, and the filtered and waste blood is combined to form another population that is subjected to an updated filtration rate. Filtration offers the needed manipulation to generate a new solution and reabsorption provides further examination.

The original KA version is executed in a more effective manner in comparison to other evolutionary algorithms. Nevertheless, there is a key issue for KA search performance, that is, KA was designed for search spaces of real-valued vectors. Nonetheless, feature selection, classification and other problems of optimization are defined in the binary discrete space. Furthermore, virtual solutes movement operator is suggestively swayed by its own optimal solution present at that point, denoted as S_{best}. The navigation of the solutes, by the S_{best}, takes them to where they are may be beneficial or detrimental to the condition. It is effective when S_{best} approaches the universal optimal solution in the search space; it is deemed ineffective or destructive when it nears the suboptimal solution. In the latter scenario, it will shift the movement of the solutes towards the suboptimal solution.

In this article, a KA algorithm for binary encoding feature selection is discussed. In previous KA version, generation of a new solution is done when the solution moves from an early iteration to the optimal solution that originates from the algorithm contained in the search space. In BKA (binary version of KA), during feature selection, generation of new solution is done through improvement of a current solution through the optimal (binary) solution based on the most popular method; maximal relevance (Max-Relevance): It involves picking features which have a high relevance to a target class c [20]. Relevance can be characterized as the connection or mutual information with the latter being a measure of the dependency of variables. In this paper, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

discussion emphasizes on mutual-information-based feature selection.

The structure of this paper is as follows: Section 2 presents previous works related to this topic. Section 3 recommends a feature selection algorithm which includes schemes of integrating mutual information using a movement strategy. Section 4 argues about issues of experimental and implementation setup. Section 5 reports on the results of the experimental setup on ten data sets that include Abalone, Iris, Glass, Spam, Tae, Waveform, Vehicle, Sonar, Wine, and WBC. Section 6 concludes the paper.

2. RELATED WORK

2.1 Feature Selection In Classification Problems

Classification is a primary task during data mining process. There are diverse heuristic search algorithms to enhance classification in feature selection. The researchers in [21] propose a PSO–SVM model which is a hybrid of the SVM (support vector machines) and PSO (particle swarm optimization), hence improving the accuracy of classification with feature selection.

This method simultaneously optimizes the input feature subset selection and the setting of SVM kernel parameter. As evident in [22], ant colony optimization (ACO) which is a hybrid algorithm, can be presented during feature selection by the use of an artificial neural network. Hybrid genetic algorithm (HGA) can be combined with a local search operation that introduces a feature selection as shown by [23].

In [24], there is the use of a modified multi-swarm PSO (MSPSO) unified with support vector machines (SVM) to handle feature selection. MSPSO encompasses many sub-swarms and a multi-swarm scheduler which are used for monitoring and controlling each and every sub-swarm by the help of certain rules. Researchers in [25] combined electromagnetism (EM) mechanism with the 1-nearest neighbor (1-NN) classifier as a wrapper method to select the optimal solution. EM-like methods make use of attraction–repulsion technique similar to the electromagnetism theory in determining the optimal solution. In [26], two chaotic maps types, namely the tent maps and logistics are entrenched in a binary PSO (BPSO) to instrument the feature selection. The objective of chaotic maps is to determine the BPSO inertia weight. Researchers in [27] extended an FS technique for SVM to tune the hyper bounds of the Gaussian automatic relevance determination (ARD) pips. A feature selection technique that utilizes a mixture of variance evolution optimization approaches and a repair technique based on feature distribution measures is introduced in [28].

Scholars in [29] elaborate a hybrid filter-wrapper feature subset selection algorithm based on PSO for SVM classification. The method is named maximum relevance minimum redundancy PSO (mr2PSO). The filter technical is based on mutual information while the wrapper technique is an adjusted distinct PSO algorithm. The mr2PSO makes use of the mutual information of the filter technique to evaluate the bit selection prospects in an isolated PSO. A PSO-based method is devised in [30] to determine the parameters and feature selection in an SVM classifier.

An entrenched technique that consecutively picks relevant features during construction of a classifier is presented in [31]. The method, known as kernel-penalized SVM, improves the anisotropic RBF Kernel shape of a classifier. In [32], a method that simultaneously conducts clustering and feature selection by the use of niching memetic algorithm is discussed. A hybrid filter-wrapper based FS algorithm is introduced in [33] to solve a classification problem by the use of the memetic framework. Researchers in [34] recommend a hybrid GA for feature selection. Devising and embedding local search operations in hybrid GA helps in fine-tuning the achieved results. In [35], a stochastic algorithm that borrows from the GRASP meta-heuristic method is proposed. Various studies introduce rough set techniques in the field of classification and feature selection. A feature selection technique that makes use of rough set theory attempts to discover the subset of features in optimal classification. For instance, a study by [36] discretizes constant features and then makes use of rough set feature selection in improving classifier performance. In [37], the authors suggested an attribute selection technique based on ambiguous gain ratio computations under the context of fuzzy rough set theory studied in tumor classification. In [38], a rough set attribute of reducing algorithm using a search technique based on particle swarm optimization (PSO) is recommended to be used as a predictor of malignancy degree in brain glioma. In [39], a novel hybrid technique that improves accuracy classification with a suitable feature subset in binary problems is proposed. and the technique is founded on enhanced search algorithm that is gravitational in nature. The algorithm utilizes a piece linear chaotic diagram in exploring a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

universal search, and uses the successive quadratic programming in accelerating a local search.

2.2 The Kidney-Inspired Algorithm (KA)

KA is one of the population-based techniques of feature selection. As suggested by its name, it reproduces various processes from the system of a biological kidney. Following are the four main elements of kidney procedures that are referenced during the imitation.

1. *Filtration:* movement of water and solutes from the blood to the tubules.

2. *Reabsorption:* transport of valuable solutes and water from the tubules to the blood.

3. *Secretion:* transfer of additional constituents that are destructive from the bloodstream to the tubule.

4. *Excretion:* moving waste products from the above processes through the urine.

In KA initial phase, an arbitrary populace of potential solutions is formed while the objective function is computed for each of the solutions. In every iteration, there is a generation of other potential solutions through a movement toward the current optimal solution. Thus, through the application of filtration operator, there is a filtration of potential solutions with high intensity toward the filtered blood (FB) with others being transferred to waste (W). The reabsorption, secretion, and excretion methods of the human kidney procedure are replicated here during the search procedure to check various conditions entrenched to the algorithm. When a potential solution is transferred to W, there is an allowance by the algorithm to have a chance of improving a solution to get an opportunity of moving it into FB. When the chance is not well exploited, the solution is expelled from W, and a potential solution is moved into W.

Conversely, when a potential solution is moved into FB after filtration and has a poor quality in comparison to the worst solution contained by FB, the solution is excreted. On the other hand, if the solution proves to be preferable compared to the worst, the worst solution contained in FB is secreted. Lastly, the different solutions contained in FB is ranked, and an update is done on the optimal solution and the filtration rate. FB and W are later combined.

Solutions in KA population represent solutes in a human kidney. For KA, there is a generation of a new solution through shifting of the solution from previous recapitulation process to the current optimal solution. The formula of the movement is as follows:

$$S_{i+1} = S_i + \text{rand}\,(S_{best} - S_i) \quad (1)$$

In Equation 1, S denotes the solution in KA population comparable to a solute in a natural kidney. S_i is a solution involved in the ith iteration. Rand value is an arbitrary value between zero and another number while S_{best} is the current solution based on the previous iterations. The equation can produce a good diversity of solutions based on a current and optimal solution. Moreover, transferring the solutions to the optimal solution strengthens the local conjunction capability of an algorithm.

Filtering of the solutions is done with a filtration rate computed using a filtration function during iterations. Calculation of the filtration rate (*fr*) is done using the following formula:

$$fr = \alpha \times \frac{\sum_{i=1}^{p} f(x_i)}{p} \quad (2)$$

α is a constant value between 0 and 1 and is attuned in advance. p represents the size of the population. $f(x_i)$ represents an objective function of solution x at ith iteration. It is evident in the above formula that the filtration rate, fr for iterations depends on the objective function value of solutions in that population. The equation represents a ratio of MOF for each solution determined by α. When α equals to zero, *fr* will equal to zero, meaning that the process of filtration for that algorithm will not take place. When the value of α is set at 1, the average value for objective functions equals to the value of *fr*. There are different rates of filtration to help in the merging of the algorithm. During iterations, objective function values get closer to the global optimal solution. and the filtration rate is thus computed using the solutions. This provides the algorithm with improved solutions. This is a form of an exploration process.

Reabsorption operator can be defined as the process of giving a solution which is being moved to W an opportunity to be included in FB. Any solution that is moved into W can be assigned to FB if after the operator responsible for the movement (Eq.1) is applied, it meets the rates of filtration and qualifies to be allotted into FB. Ideally, this simulates the reabsorption process of solutes in the kidney of a human being. In exploration, reabsorption is key.

A secretion is a form of operator for those solutions which have been moved to FB. When a solution that has the opportunity to be moved to FB

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

but does not prove to be improved in comparison to FB worst solution, secretion takes place, and the solution is moved to W; else the solution vestiges in FB while the worst solution assigned in FB is excreted and moved into W.

Secretion of solutions into W takes place if the solutions fail to satisfy the filtration rate after several attempts to be reabsorbed as part of FB.. In such a case, the solution in W is substituted with any other solution. Inserting random solutions emulates the constant process of inserting water and solutes into the glomerular capillaries of the kidney. Figure 1 below shows the pseudo code of KA.

```
set the population
evaluate the solute in the population
set the best solute, S_best
set filtration rate, fr, Eq. 2
set waste, W
set filtered blood, FB
set number of iteration, numofite
do while (ite<numofite)
    for all S_i
        generate new S_i Eq.1
        check the s_i using fr
        if S_i assigned to W
            apply reabsorption and generate S_new, Eq.1
            if reabsorption is not satisfied (S_new cannot be a part of FB)
                remove S_i from W (excretion)
                insert a random S into W to replace S_i
            endif
            S_i is reabsorbed
        else
            if it is better than the s_worst in FB
                s_worst is secreted
            else
                s_i is secreted
            endif
        endif
    endfor
    rank the Ss from FB and update the s_best
    merge W and FB
    update filtration rate, Eq.2
end while
return s_best
```

Figure 1: Pseudocode of KA [19]

In the above algorithm, the strategy of filtration and shifting to a better solution generates an algorithm that has a higher utilization or amplification. The filtration generated by the algorithm works toward creating a focus on the search space of the optimal solution. However, the movement turns more effective only when S_{best} nears the global optimum solution in the search space, and is not effective or possibly damaging, when it nears the suboptimal solution. In the last case, the solutes movement will be directed in the path of the suboptimal solution. Furthermore, KA is premeditated for search spaces of real world valuable vectors. Nonetheless, feature selection, classification together with other optimization problems are defined in the binary discrete space.

2.3 Opposition-Based Learning Strategy

In order to improve the quality of the candidate solution, Tizhoosh [40] introduced Opposition-based learning (OBL). OBL simultaneously considers a solution as well as an opposite solution. Usually population-based meta-heuristic algorithms begin with a randomly generated initial population and attempts to reach the global or near optimal solution(s). The searching process ends when some

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

predefined criterion/criteria is/are satisfied. In fact, there is a correlation between the distance of the optimal solution from the initial population and the convergence rate. The initial population is generated using the random guess in case of information absence. Thus, there is a possibility that the optimal solution is too far away from the random guess, hence may not be reached in a reasonable time. However, according to numerous studies [41-43], the computational time can be reduced by simultaneously taking into account the solution and its opposite solution. Furthermore, the empirical study of Tizhoosh [40] indicates that considering opposite direction can reduce time up to 50%. Therefore, it is far better to include a random guess and its opposite solution as initial solutions in the population-based meta-heuristic algorithms [42, 43]. In this paper, OBL strategy is employed to start the proposed binary version of the KA. This is to ensure good quality initial population and to diversify the search steps in case of stagnation of the best solutes. The idea of opposite number and opposite points is defined as follows.

Definition 1. (Opposite number) Let x be a real number in an interval $[l,u]$ $(x \in [l,u])$; the opposite number x is defined by

$$\bar{x} = u + l - x \quad (3)$$

This definition can be extended to multi dimensions [44, 45] as follows:

Definition 2. (Opposite point) Let $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ be a candidate solution in d-dimensional space, where $(x_{i1}, x_{i2}, \ldots, x_{id}) \in X$ and $x_{i1} \in [l_i, u_i]$ $\forall_i \in 1, 2, \cdots, d$. The opposite point of X_i is defined by $(\bar{X}_i) = (\bar{x}_{i1}, \bar{x}_{i2}, \ldots, \bar{x}_{id})$.

$$\bar{x}_i = u_i + l_i - x_i \quad (4)$$

Now, with the opposite point definition, the opposition-based optimization can be defined as follows:

2.3.1 Opposition-based optimization

Let $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, a point in an d-dimensional space with $x_{i1} \in [l_i, u_i]$ $\forall_i \in 1, 2, \cdots, d$, be a candidate solution. Assume $f(x)$ is a fitness function, which is used to compute the candidate's optimality. According to opposite point definition, the candidate solution $(\bar{X}_i) = (\bar{x}_{i1}, \bar{x}_{i2}, \ldots, \bar{x}_{id})$ is the opposite of $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$. Now, $if\ f(\bar{X}_i) \geq f(X_i)$, the candidate solution X_i can be replaced by the solution $\bar{X}_i$ else continue with the solution X_i. Hence, the candidate solution and its opposite candidate solution are evaluated simultaneously to obtain fitter solution.

3 THE PROPOSED ALGORITHM

Optimization comes with many problems in the form of feature selection, reduction of dimensionality [17, 46-49], data mining [50], unit commitment [51], and formation of cells [52], where it is natural to encrypt solutions to appear as binary vectors. Additionally, problems set in the real space can also be deliberated in the binary space. The solution is displayed in real digits using bits in the binary mode. To some extent, binary search space can be viewed as a hypercube where an agent moves to nearer and farther corners of the hypercube by overturning different bit numbers.

In this segment, a binary version of KA is presented for feature selection (OBKA-FS). Various primary concepts of KA will necessitate a modification procedure. In a discrete binary setting, each dimension assumes either 0 or 1. To move through a dimension translates into having agreeing variable value changes from 0 to 1 or 1 to 0. To allow for the introduction of a binary version of the kidney algorithm, the filtration rate updating formula may be formulated similar to the continuous algorithm (Eq. 2). A dominant difference between continuous and binary KA is the fact that in the binary algorithm, the updating actually means the switch from "0" and "1" values. The switching reflects the relevance of the target class, c which is overtly measured by renown measures in defining the dependence of variables or the mutual information (MI) [53].

Figure 2 below depicts the algorithmic flow of the proposed OBKA-FS. The key idea of our proposed MI-based switching is to update the position in such a way that the active bit value is represented by its conforming feature which is altered according to the MI value of that feature.

Journal of Theoretical and Applied Information Technology
15th January 2017. Vol.95. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

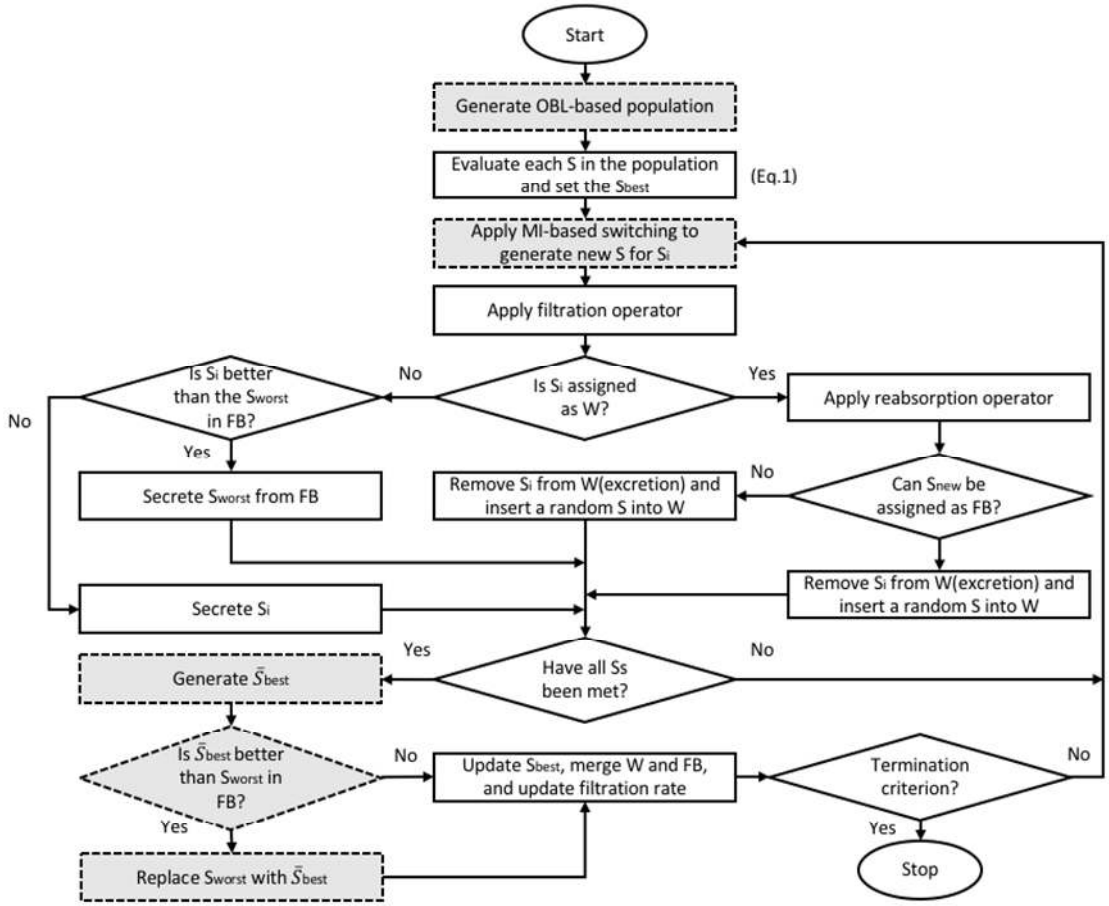

Figure 2: Flowchart of OBKA-FS

In other words, the feature containing the minimum MI (F_{minMI}) will be eradicated if the feature subsets in S_{best} are not more than the number of feature subset in the solution to be transferred (S_i). With, F_{minMI} being replaced with a bit that is yet to be selected and with a high MI value from the S_{best} the amount of feature subset in S_{best} will equal the amount of feature subset in S_i. Another potential movement is when the amount of feature subset in S_{best} is more than the feature subset in S_i. In this case, a still unselected feature from S_{best} with the biggest value of MI will be set to 1 in S_{i+1}. Accordingly, a new solution is generated through an effort to advance an existing solution through the feature set's best solution.

To allow exploration of unmapped sections of the search space and eliminate the suboptimal solution, the suggestion in [43, 45, 54-57] is to highlight trivial random mistakes or contemplate on the reverse direction of the solution. In this study, an opposition on the S_{best} is used to replace the worst fitting solution in FB. It will help the algorithm in exploring regions that are not discovered by the use of the worst fit solution (updated using an opposite S_{best} solute). The algorithm will concurrently hold on to the global optimal solution with the S_{best} solute, as there is no modification executed in the S_{best} solute itself.

3.1 Opposition-Based Population Initialization

Population initialization is the first and crucial step in any meta-heuristic algorithm. This step affects the quality of the final solution as well as the convergence speed [58]. In absence of any information about the solution, the most frequently used approach is the random initialization. However, numerous experimental studies [41, 43, 54, 56, 59] have shown that immediate consideration of the random solutions and their opposite decreases the chance of exploring vain regions in the search space, and increases the chance of selecting good quality initial population. Therefore, integrating OBL with the OBKA-FS is worth investigating. Here, a combined initial population of size 2S is generated using uniform

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

random distribution and the OBL strategy, and finally the finest S solutes (Out of the 2S solutes) are included in the initial population. The pseudocode of opposition-based population initialization is shown in Figure 3.

$\{X\}$ = Randomly generated S solutes
for $i = 1$ *to* S **do**
  for $j = 1$ *to* d **do**
    $\bar{X}_{ij} = u_j + l_j - X_{ij}$
  end for
end for
$\{OX\} = \{X\} \cup \{\bar{X}\}$
Compute fitness of solutes in $\{OX\}$ using Eq. (9).
Sort $\{OX\}$ with fitness values.
$\{X\} = top(\{OX\}/2)$ *solutes*
Return $\{X\}$

Figure 3: Opposition-based population initialization

3.2 The Fitness Function

The condition of optimal classification often translates into the minimal classification error. Here, minimal error usually requires the maximal statistical dependency of the target class c on the data distribution in the subspace R^m (and vice versa) [20]. Nevertheless, during feature selection, it is noted that the blending of independent features often does not translate into noble performance in the classification. Precisely, "the m best features are not the best m features" [60-63]. Various scholars have researched on indirect and direct methods to reduce the redundancy of features (example, [61, 62, 64-67]) and select features that have the least redundancy level and greatest dependency, that is, Min-Redundancy and Max-Dependency. Thus, the creation of fitness function is based on the three-set criteria, that is, accuracy of classification, the number of selected features, and Min-Redundancy-Max-Dependency. Nonetheless, as Max-Dependency condition is complex during its implementation, another option is to select features using Max-Relevance criterion [20]. Feature selection using Max-Relevance seeks to select the features possessing the uppermost applicability to the target class c. Relevance can be characterized in terms of association or mutual information, with MI being a widely used measure of defining variable dependency. Hence, the Max-Relevance and Min-Redundancy (mRMR) is similar to the Max-Dependency and Min-Redundancy.

Attaining a good fitness value is the same as attaining a high accuracy of classification; low numbers of dimensional and a minimal redundancy and maximal dependency. Solving the problem of several objectives is done by generating a fitness function that will integrate the three objectives into a sole objective. The fitness function ca be defined as:

$$\text{fit} = \omega_1 \times \text{nf} + \omega_2 \times \text{acc} + \omega_3 \times \Phi(\text{D}, \text{R}) \quad (5)$$

Here, three weight factors are predefined ω_1, ω_2 and ω_3 where ω_1 is the number of selected features weight factor, ω_2 is accuracy classification weight factor ($accu_i$) of the 1-nearest neighbor (1-NN) found using the 5-fold cross-validation method, and ω_3 is mRMR weight factor. Accuracy weight factor can be attuned to a higher value like 100% if accuracy is an important aspect. The $accu_i$ is achieved by Eq. (6), where *cc* denotes the correctly classified cases and *uc* represents the number of incorrectly classified cases [21, 68].

$$\text{acc}_i = \frac{cc}{cc+uc} \times 100\% \quad (6)$$

For this purpose, discussion is focused on mutual-information-based feature selection. Max-Relevance is used in searching features that satisfy (7), which is the approximation of exact value by computing an average value of MI values amid distinct feature x_i and class c:

$$\max \text{D}(\text{S}, \text{c}), \text{D} = \frac{1}{|\text{S}|}\sum_{\text{x}_\text{i} \in \text{S}} \text{I}(\text{x}_\text{i}; \text{c}) \quad (7)$$

There is a probability that features selected using Max-Relevance may come with rich redundancy, that is, there will be a large dependency between the features. If features have a high dependency among them, the corresponding class-discriminative influence will not be changed if some features are detached. Thus, the minimal redundancy (Min-Redundancy) stated below can be brought forth to select features that are mutually exclusive [67]:

$$\min \text{R}(\text{S}), \text{R} = \frac{1}{|\text{S}|^2}\sum_{\text{x}_\text{i}, \text{x}_\text{j} \in \text{S}} \text{I}(\text{x}_\text{i}; \text{x}_\text{j}) \quad (8)$$

A criterion in which two limitations are combined is known as "minimal-redundancy-maximal-relevance" (mRMR) [67]. The operator is denoted as$\Phi(\text{D}, \text{R})$. The easiest form of optimizing D and R is considered as shown below:

$$\max \Phi(\text{D}, \text{R}), \Phi = \text{D} - \text{R} \quad (9)$$

3.3 Mutual Information (MI)

Mutual Information can be verified in an equivalent manner as attaining data for binary problems. However, it is not the same case for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

diverse groups. Thus, there is the provision of Mutual Information where the equation acts as the dividing characteristic assortment algorithm. Computation of Mutual Information, as well as a category pair, is as shown below in Eq. (10);

$$\mathrm{MI}(\mathrm{F},\mathrm{C_k}) = \sum_{\mathrm{v_f}\in(1,0)}\sum_{\mathrm{v_{C_k}}\in(1,0)} \mathrm{P}(\mathrm{F}=\mathrm{v_f},\mathrm{C_k}=\mathrm{v_{C_k}}) \ln\frac{\mathrm{P}(\mathrm{F}=\mathrm{v_f},\mathrm{C_k}=\mathrm{v_{C_k}})}{\mathrm{P}(\mathrm{F}=\mathrm{v_f})\mathrm{P}(\mathrm{C_k}=\mathrm{v_{C_k}})} \quad (10)$$

Here, F is an independent arbitrary non-consistent "feature" taking the value $\mathrm{v_F} = (1,0)$ (feature F may occur in document or fail), $\mathrm{C_k}$ is a distinct arbitrary variable "category" taking the values $\mathrm{v_{C_k}} = (1,0)$ (document may or may not belong to category $\mathrm{C_k}$).

The predictions can be made using tallies from different documents picked from the training set. By using the notation stated at the start of section 2.3, Equation (10) can be rewritten to form Equation (11) as follows:

$$\begin{aligned}\mathrm{MI}(\mathrm{F},\mathrm{C_k}) = {} & \frac{\mathrm{N_{F,C_k}}}{\mathrm{N}}\ln\frac{\mathrm{NN_{F,C_k}}}{\mathrm{N_F N_{C_k}}} \\ & + \frac{\mathrm{N_{F,\overline{C_k}}}}{\mathrm{N}}\ln\frac{\mathrm{NN_{F,\overline{C_k}}}}{\mathrm{N_F N_{\overline{C_k}}}} \\ & + \frac{\mathrm{N_{\overline{F},C_k}}}{\mathrm{N}}\ln\frac{\mathrm{NN_{\overline{F},C_k}}}{\mathrm{N_{\overline{F}} N_{C_k}}} \\ & + \frac{\mathrm{N_{\overline{F},\overline{C_k}}}}{\mathrm{N}}\ln\frac{\mathrm{NN_{\overline{F},\overline{C_k}}}}{\mathrm{N_{\overline{F}} N_{\overline{C_k}}}}\end{aligned} \quad (11)$$

We can then weigh and summarize the values to form a global ranked list of features:

$$\mathrm{MI}(\mathrm{F}) = \sum_{\mathrm{k}=1}^{|\mathrm{C}|}\frac{\mathrm{N_{C_k}}}{\mathrm{N}}\mathrm{MI}(\mathrm{F},\mathrm{C_k}) \quad (12)$$

Nevertheless, using this technique, all continuous-valued features are quantized to three levels by the use of quantization boundaries at $\mu \pm \sigma$ where μ and σ stand for the feature's projected mean and standard deviation respectively. This produces a list of discretized features, $\mathrm{y(f)}, \mathrm{f} \in \mathrm{F}$. Selection of features takes place one at a time while referencing the rule below;

$$\mathrm{S_d} = \mathrm{S_{d-1}} \cup \operatorname{argmax}\left[\mathrm{I}(\mathrm{y(f)},\mathrm{z}) - \frac{1}{\mathrm{d}-1}\sum_{\mathrm{g}\in\mathrm{S_{d-1}}}\mathrm{I}(\mathrm{y(f)},\mathrm{y(g)})\right] \quad (13)$$

Here, I is the mutual information (reference Eq. (13)) while z is the categorical variable that contains the class labeling.

3.4 Feature Selection Using OBKA-FS

Feature selection problem using heuristic search algorithms is coded in binary format. The resulting solution is therefore, a binary string representing the subset of features, i.e., the best features for classification objective. With 'p' numbers of features, search space can be as an n=p dimensional binary space. Each solute is a binary vector in this search space representing a subset of features. In this binary vector, every bit is related to a feature. When the i-th bit of the vector is equal to 1, then the ith feature is permitted to take part in the classification, else, the respective feature is omitted. For the purpose of evaluation, the subset of features allied to the 1-bits in the binary string of the agent during classification as well as the output results are examined. The evaluation function is set prior to the calculation of the classification accuracy.

In this study, the OBKA-FS is used as a feature selection tool. The objective of OBKA-FS in feature selection is to find an optimum binary vector with every bit being related to a feature. Evaluation of each subset of features is done with respect to a classification fitness function. The use of OBKA-FS ensures selection of the optimal set of features with the objective to optimize the evaluation function.

Figure 4 shows the block diagram. The features selected for each solution are brought to the classifier with fitness function value being fed back to the OBKA-FS. The solutes go into the search space using a strategy that results in the optimum solution of the evaluation function. The OBKA-FS iterates until it satisfies the stopping criteria [8]. Eventually, the best solution obtained represents a subset of features offered by OBKA-FS.

Journal of Theoretical and Applied Information Technology
15th January 2017. Vol.95. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

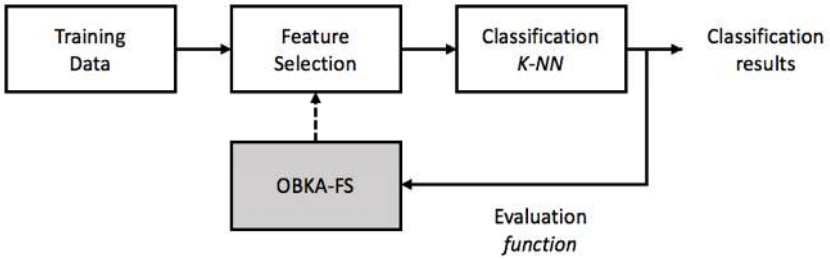

Figure 4: The proposed classification model based on OBKA-FS

4. EXPERIMENTAL SETUP AND IMPLEMENTATION ISSUES

For the purpose of this study, experimentation of feature selection using OBKA-FS is done in the classification of renowned datasets. The outcomes are three algorithms that can be likened to each other, that is, BPSO by Chuang, Yang [26], IBGSA by Rashedi and Nezamabadi-pour [69], and the proposed OBKA-FS. In BPSO, the positive constants $c_1 = c_2 = 2$, and inertia factor (w) decreases linearly from 0.9 to 0.2 [70]. In this experiment, the population size is 20 and the maximum iteration number is set to 50. Eq. (14) is used for the gravitational constant where G_0 is set to 1 for IBGSA. In IBGSA, k_1, which is the initial number of agents, is set to equal to 1 while the total number of agents, k_2 equals to 500.

$$G(t) = G_0(1 - \frac{t}{T}) \quad (14)$$

4.1. Dataset Description

The datasets shown in Table 1, derived from UCI Machine Learning Repository can be utilized in evaluating the performance levels of the proposed FS method: Abalone, Iris, Glass, Spam, Vehicle, Tae, Waveform, Sonar, Wine, and WBC. Table 1 outlines the characteristics of the datasets showing a significant diversity in the given examples, features, and classes.

Table 1: The datasets used in the experiment

No	Database name	Number of classes	Number of features	Number of samples
1	Abalone	11	8	3842
2	Glass	6	9	214
3	Iris	3	4	150
4	Spam	2	57	4601
5	Tae	3	5	151
6	Vehicle	4	18	846
7	Waveform	3	21	5000
8	Wine	3	13	178
9	Sonar	2	60	208
10	WBC	2	9	683

4.2. Evaluation Criteria

In this study, other than the fitness function parameters, the feature reduction ratio (Fr) criterion is also defined. For a database, F containing PT samples, $F = (F_1, F_2, \ldots, F_{PT})$. Samples from C different categories were derived. Each sample, F_i is a feature vector containing p number of features, that is, $F_i = [f_i^1, f_i^2, \ldots, f_i^P]$. A feature selection method limits the number of features to *p*. Therefore, with the help of feature selection, a reduction of the number of features can be obtained by using a feature reduction ratio shown in Eq. (15). In other cases, feature selection efficiency in improving the classification results is evaluated using some evaluation functions where each result has a certain point of view.

$$Fr = \frac{p-q}{p} \quad (15)$$

5. RESULTS AND DISCUSSIONS

In this part, OBKA-FS is examined with respect to its efficiency and classification accuracy. Each experiment is carried out using a Mac OS X environment and a machine that has a core i5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

processor and 8GB of RAM. Coding the algorithms is done using MATLAB. Table 2 shows the performance of 1-NN classifier with and without using the feature selection method based on OBKA-FS. In particular, the mean of accuracy, selected features, and feature reduction ratio (Fr) values from 5 independent runs on every dataset and algorithm are reported. One can observe that there is a reduction in the number of features in all datasets. At the same time, there is an improvement in the accuracy of 1-NN classifier.

Table 1: Comparisons between the performance of 1-NN classifier with and without OBKA-FS

Dataset	1-NN without FS		OBKA-FS+1-NN		Performance	
	Accuracy (%)	# of feature	Accuracy (%)	# of feature	Feature Reduction Ratio (%)	Accuracy Improvement (%)
Abalone	52.79	8	54.62	5	0.38	1.83
Glass	58.88	9	74.95	3.8	0.58	6.54
iris	96.67	4	98.13	3	0.25	1.47
Sonar	83.65	60	90.10	14.6	0.76	6.44
Spam	91.68	57	92.21	19.2	0.66	0.53
Tae	47.02	5	57.48	2	0.60	10.46
Vehicle	70.21	18	74.28	6.8	0.62	4.07
WBC	70.20	9	98.37	7.2	0.20	4.85
Waveform	83.76	21	84.62	15	0.29	0.86
Wine	79.53	13	98.12	7	0.46	13.73

Figure 5 depicts the best fitness function values obtained from the 5 running times of the algorithms over all dataset. In seven out of eleven datasets OBKA-FS produced better results than the other competitive algorithms, i.e. IBGSA and BPSO. No substantial differences can be observed for the other datasets results.

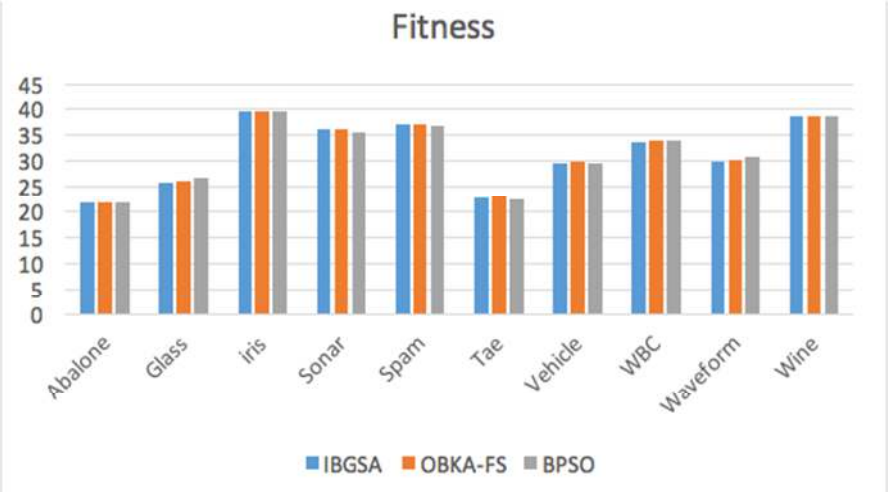

Figure 5: Best fitness values

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3-5 below displays the results of the performance factors used in the fitness function. Table 3 shows the mean and standard deviation of classification accuracy and the average selected features of each algorithm and data set. From the results, it can be seen that the classification accuracy of 1-NN utilizing the OBKA-FS outperformed other state-of-the-art algorithms on seven benchmarks. Although there was a non-significant differences between the average number of selected features among the tested algorithms, it can be inferred that OBKA-FS has selected the most useful set of features for all datasets.

Table 3: Best number of features and accuracy

Dataset	IBGSA+1-NN		OBKA-FS+1-NN		BPSO+1-NN	
	Accuracy(%)	# of features	Accuracy(%)	# of features	Accuracy(%)	# of features
Abalone	54.54±0.15	5.00±0.00	**54.62±0.19**	5.00±0.00	54.46±0.18	5.00±0.00
Glass	64.20±0.53	4.80±0.45	**74.95±0.39**	**3.80±0.45**	66.07±0.85	6.20±0.84
Sonar	90.10±0.43	15.60±2.07	**90.10±0.65**	14.60±1.34	88.56±0.53	14.20±0.84
Spam	92.51±0.02	20.60±2.70	92.21±0.08	19.20±1.30	91.87±0.07	18.20±2.28
Tae	56.82±0.98	2.00±0.00	**57.48±1.09**	2.00±0.00	55.89±0.36	2.00±0.00
Vehicle	73.40±0.36	6.80±1.10	**74.28±0.49**	**6.80±1.64**	73.95±0.37	7.40±0.55
WBC	84.17±0.22	15.00±0.00	**98.37±0.19**	**15.00±1.23**	84.47±0.27	15.20±1.30
Waveform	74.04±0.58	7.00±1.58	75.05±1.16	7.20±1.64	76.57±1.27	6.80±1.30
Wine	96.29±0.31	4.60±0.55	**98.12±0.25**	4.60±0.55	96.18±0.47	4.80±0.45

The best values of minimal-redundancy-maximal-relevance are presented in Table 4. OBKA-FS has outperformed other algorithms by selecting the minimal redundancy and maximal relevance feature set in seven datasets. However, in the remaining datasets, OBKA-FS nearly obtained similar results.

Table 4: Best $\Phi(D,R)$

Dataset	IBGSA	OBKA-FS	BPSO	Dataset	IBGSA	OBKA-FS	BPSO
Abalone	-0.11±0.00	-0.11±0.00	-0.11±0.00	Tae	0.04±0.00	0.04±0.00	0.04±0.00
Glass	0.25±0.03	0.24±0.01	0.23±0.02	Vehicle	0.04±0.01	**0.05±0.01**	0.03±0.01
iris	0.80±0.01	**0.80±0.13**	0.71±0.05	WBC	0.18±0.01	**0.19±0.00**	0.18±0.00
Sonar	0.027±0.00	**0.031±0.01**	0.03±0.00	Waveform	-0.00±0.02	-0.01±0.03	-0.03±0.06
Spam	0.078±0.01	**0.078±0.00**	0.08±0.01	Wine	**0.45±0.04**	**0.45±0.04**	0.43±0.03

According to Rashedi and Nezamabadi-pour [69], there is no universal heuristic algorithm that can get the best results on the entire available benchmarks. However, the results obtained by OBKA-FS verify that the suggested algorithm can be a useful method for feature selection.

6. CONCLUSION

In recent years, various meta-heuristic optimization algorithms have been developed. KA is a new meta-heuristic search algorithm constructed based on the functionality of the kidney in the body of a human being. In this article, a binary version of KA has been introduced for feature selection. To improve the results, some improvements are made in KA algorithm. The proposed version of KA for feature selection (OBKA-FS) has integrated an opposition-based initialization method in order to start with good initial solutes. Moreover, a new movement strategy based on the calculation of mutual information (MI) has been used. This strategy gives OBKA-FS the ability to work in discrete binary environment. The proposed feature selection model using OBKA-FS is tested on the classification of some UCI databases. OBKA-FS is compared with some well-known algorithms, namely the BPSO and IBGSA. The experimental results confirm the effectiveness

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and efficiency of the proposed method and show that it can be successfully applied as a feature selection method for classification problems beside other algorithms that have proved their efficiencies thus far. The proposed classification model may be used for classification purposes in our future work.

REFRENCES:

[1] Mitra, P., C.A. Murthy, and S.K. Pal, *Unsupervised feature selection using feature similarity.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. **24**(3): p. 301-312.

[2] Dash, M., et al. *Feature selection for clustering-a filter solution*. in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. 2002. IEEE.

[3] Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection.* The Journal of Machine Learning Research, 2003. **3**: p. 1157-1182.

[4] Karahoca, A., et al., *Feature selection on Persian fonts: A comparative analysis on GAA, GESA and GA.* Procedia Computer Science, 2011. **3**: p. 1249-1255.

[5] Lew, M.S., *Principles of Visual Information Retrieval*. 2010: Springer Publishing Company, Incorporated. 356.

[6] Li, C. and J. Zhou, *Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm.* Energy Conversion and Management, 2011. **52**(1): p. 374-381.

[7] Liu, L., W. Zhong, and F. Qian, *An improved chaos-particle swarm optimization algorithm.* Journal of East China University of Science and Technology, 2010. **36**(2): p. 267-272.

[8] Liu, H. and L. Yu, *Toward integrating feature selection algorithms for classification and clustering.* IEEE Transactions on knowledge and data engineering, 2005. **17**(4): p. 491-502.

[9] Yanxi, L. and F. Dellaert. *A classification based similarity metric for 3D image retrieval*. in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. 1998.

[10] Maldonado, S. and R. Weber, *A wrapper method for feature selection using support vector machines.* Information Sciences, 2009. **179**(13): p. 2208-2217.

[11] Dash, M. and H. Liu, *Feature selection for classification.* Intelligent Data Analysis, 1997. **1**(1): p. 131-156.

[12] Huan, L. and Y. Lei, *Toward integrating feature selection algorithms for classification and clustering.* IEEE Transactions on Knowledge and Data Engineering, 2005. **17**(4): p. 491-502.

[13] Dash, M. and H. Liu, *Feature selection for classification.* Intelligent data analysis, 1997. **1**(3): p. 131-156.

[14] Raymer, M.L., et al., *Dimensionality reduction using genetic algorithms.* IEEE Transactions on Evolutionary Computation, 2000. **4**(2): p. 164-171.

[15] Tanaka, K., T. Kurita, and T. Kawabe. *Selection of import vectors via binary particle swarm optimization and cross-validation for kernel logistic regression*. in *2007 International Joint Conference on Neural Networks*. 2007. IEEE.

[16] Bello, R., et al. *Two-step particle swarm optimization to solve the feature selection problem*. in *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*. 2007. IEEE.

[17] Wang, X., et al., *Feature selection based on rough sets and particle swarm optimization.* Pattern Recognition Letters, 2007. **28**(4): p. 459-471.

[18] Zhang, H. and G. Sun, *Feature selection using tabu search method.* Pattern Recognition, 2002. **35**(3): p. 701-711.

[19] Jaddi, N.S., J. Alvankarian, and S. Abdullah, *Kidney-inspired algorithm for optimization problems.* Communications in Nonlinear Science and Numerical Simulation, 2017. **42**: p. 358-369.

[20] Hanchuan, P., L. Fuhui, and C. Ding, *Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005. **27**(8): p. 1226-1238.

[21] Huang, C.-L. and J.-F. Dun, *A distributed PSO–SVM hybrid system with feature selection and parameter optimization.* Applied Soft Computing, 2008. **8**(4): p. 1381-1391.

[22] Kabir, M.M., M. Shahjahan, and K. Murase, *A new hybrid ant colony optimization algorithm for feature selection.* Expert Systems with Applications, 2012. **39**(3): p. 3747-3763.

[23] Kabir, M.M., M. Shahjahan, and K. Murase, *A new local search based hybrid genetic algorithm for feature selection.* Neurocomputing, 2011. **74**(17): p. 2914-2928.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[24] Liu, Y., et al., *An improved particle swarm optimization for feature selection.* Journal of Bionic Engineering, 2011. **8**(2): p. 191-200.
[25] Su, C.-T. and H.-C. Lin, *Applying electromagnetism-like mechanism for feature selection.* Information Sciences, 2011. **181**(5): p. 972-986.
[26] Chuang, L.-Y., C.-H. Yang, and J.-C. Li, *Chaotic maps based on binary particle swarm optimization for feature selection.* Applied Soft Computing, 2011. **11**(1): p. 239-248.
[27] Wang, J.N. and X.T. Li. *An improved gravitation search algorithm for unconstrained optimization*. in *Advanced Materials Research*. 2011. Trans Tech Publ.
[28] Khushaba, R.N., A. Al-Ani, and A. Al-Jumaily, *Feature subset selection using differential evolution and a statistical repair mechanism.* Expert Systems with Applications, 2011. **38**(9): p. 11515-11526.
[29] Unler, A., A. Murat, and R.B. Chinnam, *mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification.* Information Sciences, 2011. **181**(20): p. 4625-4641.
[30] Lin, S.-W., et al., *Particle swarm optimization for parameter determination and feature selection of support vector machines.* Expert systems with applications, 2008. **35**(4): p. 1817-1824.
[31] Maldonado, S., R. Weber, and J. Basak, *Simultaneous feature selection and classification using kernel-penalized support vector machines.* Information Sciences, 2011. **181**(1): p. 115-128.
[32] Sheng, W., X. Liu, and M. Fairhurst, *A niching memetic algorithm for simultaneous clustering and feature selection.* IEEE Transactions on Knowledge and Data Engineering, 2008. **20**(7): p. 868-879.
[33] Zhu, Z., Y.S. Ong, and M. Dash, *Wrapper–Filter Feature Selection Algorithm Using a Memetic Framework.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007. **37**(1): p. 70-76.
[34] Il-Seok, O., L. Jin-Seon, and M. Byung-Ro, *Hybrid genetic algorithms for feature selection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004. **26**(11): p. 1424-1437.
[35] Bermejo, P., J.A. Gámez, and J.M. Puerta, *A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets.* Pattern Recognition Letters, 2011. **32**(5): p. 701-711.
[36] Tian, D., X.-j. Zeng, and J. Keane, *Core-generating approximate minimum entropy discretization for rough set feature selection in pattern classification.* International Journal of Approximate Reasoning, 2011. **52**(6): p. 863-880.
[37] Dai, J. and Q. Xu, *Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification.* Applied Soft Computing, 2013. **13**(1): p. 211-221.
[38] Wang, X., et al., *Rough set feature selection and rule induction for prediction of malignancy degree in brain glioma.* Computer methods and programs in biomedicine, 2006. **83**(2): p. 147-156.
[39] Xiang, J., et al., *A novel hybrid system for feature selection based on an improved gravitational search algorithm and k-NN method.* Applied Soft Computing, 2015. **31**: p. 293-307.
[40] Tizhoosh, H.R. *Opposition-Based Learning: A New Scheme for Machine Intelligence*. in *Cimca/iawtic*. 2005.
[41] Ahandani, M.A. and H. Alavi-Rad, *Opposition-based learning in shuffled frog leaping: An application for parameter identification.* Information Sciences, 2015. **291**: p. 19-42.
[42] Ma, X., et al., *MOEA/D with opposition-based learning for multiobjective optimization problem.* Neurocomputing, 2014. **146**: p. 48-64.
[43] Shaw, B., V. Mukherjee, and S.P. Ghoshal, *Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm.* International Journal of Electrical Power & Energy Systems, 2014. **55**: p. 29-40.
[44] Tizhoosh, H.R. *Reinforcement learning based on actions and opposite actions*. in *International conference on artificial intelligence and machine learning*. 2005.
[45] Rahnamayan, S., H.R. Tizhoosh, and M.M.A. Salama, *Opposition-Based Differential Evolution.* Evolutionary Computation, IEEE Transactions on, 2008. **12**(1): p. 64-79.
[46] Pal, A. and J. Maiti, *Development of a hybrid methodology for dimensionality reduction in Mahalanobis–Taguchi system using Mahalanobis distance and binary particle swarm optimization.* Expert Systems with Applications, 2010. **37**(2): p. 1286-1293.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[47] Bereta, M. and T. Burczyński, *Comparing binary and real-valued coding in hybrid immune algorithm for feature selection and classification of ECG signals.* Engineering Applications of Artificial Intelligence, 2007. **20**(5): p. 571-585.

[48] Chuang, L.-Y., et al., *Improved binary PSO for feature selection using gene expression data.* Computational Biology and Chemistry, 2008. **32**(1): p. 29-38.

[49] Zeng, X.-P., Y.-M. Li, and J. Qin, *A dynamic chain-like agent genetic algorithm for global numerical optimization and feature selection.* Neurocomputing, 2009. **72**(4): p. 1214-1228.

[50] Srinivasa, K., K. Venugopal, and L.M. Patnaik, *A self-adaptive migration model genetic algorithm for data mining applications.* Information Sciences, 2007. **177**(20): p. 4295-4313.

[51] Yuan, X., et al., *An improved binary particle swarm optimization for unit commitment problem.* Expert Systems with applications, 2009. **36**(4): p. 8049-8055.

[52] Wu, T.-H., C.-C. Chang, and S.-H. Chung, *A simulated annealing algorithm for manufacturing cell formation problems.* Expert Systems with Applications, 2008. **34**(3): p. 1609-1617.

[53] Coelho, F., A.P. Braga, and M. Verleysen, *A Mutual Information estimator for continuous and discrete variables applied to Feature Selection and Classification problems.* International Journal of Computational Intelligence Systems, 2016. **9**(4): p. 726-733.

[54] Gao, W.-f., S.-y. Liu, and L.-l. Huang, *Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique.* Communications in Nonlinear Science and Numerical Simulation, 2012. **17**(11): p. 4316-4327.

[55] Tsafarakis, S., et al., *Hybrid particle swarm optimization with mutation for optimizing industrial product lines: An application to a mixed solution space considering both discrete and continuous design variables.* Industrial Marketing Management, 2013. **42**(4): p. 496-506.

[56] Wang, H., et al., *Enhancing particle swarm optimization using generalized opposition-based learning.* Information Sciences, 2011. **181**(20): p. 4699-4714.

[57] Xiao-Jun, Z., et al., *A particle swarm optimization algorithm with variable random functions and mutation.* Acta Automatica Sinica, 2014. **40**(7): p. 1339-1347.

[58] Seif, Z. and M.B. Ahmadi, *Opposition versus randomness in binary spaces.* Applied Soft Computing, 2015. **27**: p. 28-37.

[59] Cheng, M.-Y. and D.-H. Tran, *Opposition-based Multiple Objective Differential Evolution (OMODE) for optimizing work shift schedules.* Automation in Construction, 2015. **55**: p. 1-14.

[60] Cover, T.M., *The best two independent measurements are not the two best.* IEEE Transactions on Systems, Man, and Cybernetics, 1974(1): p. 116-117.

[61] Cover, T.M. and J.A. Thomas, *Elements of information theory*. 2012: John Wiley & Sons.

[62] Jain, A.K., R.P.W. Duin, and J. Mao, *Statistical pattern recognition: A review.* IEEE Transactions on pattern analysis and machine intelligence, 2000. **22**(1): p. 4-37.

[63] Webb, A.R., *Statistical pattern recognition*. 2003: John Wiley & Sons.

[64] Li, W. and Y. Yang, *How many genes are needed for a discriminant microarray data analysis*, in *Methods of microarray data analysis*. 2002, Springer. p. 137-149.

[65] Kohavi, R. and G.H. John, *Wrappers for feature subset selection.* Artificial intelligence, 1997. **97**(1): p. 273-324.

[66] Jäger, J., R. Sengupta, and W.L. Ruzzo. *Improved gene selection for classification of microarrays*. in *Proceedings of the eighth Pacific Symposium on Biocomputing: 3–7 January 2003; Lihue, Hawaii*. 2002.

[67] Ding, C. and H. Peng, *Minimum redundancy feature selection from microarray gene expression data.* Journal of bioinformatics and computational biology, 2005. **3**(02): p. 185-205.

[68] Huang, C.-L. and C.-J. Wang, *A GA-based feature selection and parameters optimizationfor support vector machines.* Expert Systems with Applications, 2006. **31**(2): p. 231-240.

[69] Rashedi, E. and H. Nezamabadi-pour, *Feature subset selection using improved binary gravitational search algorithm.* Journal of Intelligent & Fuzzy Systems, 2014. **26**(3): p. 1211-1221.

[70] Rashedi, E., H. Nezamabadi-pour, and S. Saryazdi, *BGSA: binary gravitational search algorithm.* Natural Computing, 2010. **9**(3): p. 727-745.