

## Research Article

# Feature Selection and Parameter Optimization of Support Vector Machines Based on Modified Artificial Fish Swarm Algorithms

Kuan-Cheng Lin,<sup>1</sup> Sih-Yang Chen,<sup>1</sup> and Jason C. Hung<sup>2</sup>

<sup>1</sup> Department of Management Information Systems, National Chung Hsing University, Taichung 40227, Taiwan

<sup>2</sup> Department of Information Technology, Overseas Chinese University, Taichung 40721, Taiwan

Correspondence should be addressed to Kuan-Cheng Lin; [kuanchenglin@gmail.com](mailto:kuanchenglin@gmail.com)

Received 21 August 2014; Accepted 22 September 2014

Academic Editor: Jong-Hyuk Park

Copyright © 2015 Kuan-Cheng Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rapid advances in information and communication technology have made ubiquitous computing and the Internet of Things popular and practicable. These applications create enormous volumes of data, which are available for analysis and classification as an aid to decision-making. Among the classification methods used to deal with big data, feature selection has proven particularly effective. One common approach involves searching through a subset of the features that are the most relevant to the topic or represent the most accurate description of the dataset. Unfortunately, searching through this kind of subset is a combinatorial problem that can be very time consuming. Metaheuristic algorithms are commonly used to facilitate the selection of features. The artificial fish swarm algorithm (AFSA) employs the intelligence underlying fish swarming behavior as a means to overcome optimization of combinatorial problems. AFSA has proven highly successful in a diversity of applications; however, there remain shortcomings, such as the likelihood of falling into a local optimum and a lack of multiplicity. This study proposes a modified AFSA (MAFSA) to improve feature selection and parameter optimization for support vector machine classifiers. Experiment results demonstrate the superiority of MAFSA in classification accuracy using subsets with fewer features for given UCI datasets, compared to the original FASA.

## 1. Introduction

Advances in information and communications technology have led to a rapid increase in the data processing and computing power of handheld devices. This has made it possible to obtain information anytime and anywhere, ushering in the era of ubiquitous computing (ubicomputing) and the Internet of Things (IoTs). Applications, such as photo sharing and social networking, create enormous volumes of digital data, which is available to aid in decision-making. Feature selection is particularly effective in the classification of big data in fields such as data mining, pattern recognition [1], bioinformatics [2], arrhythmia classification [3], and numerous others.

Supervised learning methods, such as decision trees [4], support vector machines (SVM) [5–7], and neural networks [8–11], are used to classify data into appropriate categories.

The classification of data requires the classification of data we already know and then divide them into training data and testing data. After building a classification model using the training data, the testing data is used to evaluate the model according to classification accuracy. SVM is based on the statistical theories proposed by Chervonenks [12] and the principle of minimizing structural risk. SVM is commonly used to solve classification or regression problems by finding the optimal hyperplane. SVM provides excellent classification accuracy using a small training set and is easy to implement. This study adopted SVM as a classifier in conjunction with the wrapper method of feature selection.

Feature selection is used to filter out large amounts of unnecessary data, much of which is irrelevant, redundant, and/or noisy. Irrelevant features are unrelated to a given goal and redundant features represent the same information,

despite containing different types of data. Noisy features contain wrong or missing data. Sifting through this unnecessary data would result in enormous computing costs or even skew the results. Feature selection is used to identify the features that are essential to a given task.

Two methods are commonly used for feature selection: filter and wrapper. The filter approach is based on assigning weights to every feature, such as distance or dependability, and then combining the features with the highest weights in order to obtain an optimal subset. The wrapper involves collocating using metaheuristic algorithm and then assembling an optimized subset of features by eliminating or combining features and calculating a fitness value for each feature subset on the basis of classification accuracy. The filter approach tends to be quicker but suffers from lower classification accuracy. The need for the classifier to conduct training extends the processing of the wrapper approach; however, the classification accuracy is far higher.

One common wrapper method involves compiling a subset of optimal features of the highest relevance with the most accurate description of the characteristics in the dataset. This can be very time consuming due to the fact that any increase in the number of features exponentially expands the number of combinations in the feature subset, which is known as the curse of dimensionality [13]. This study used a metaheuristic algorithm to obtain good-enough or near-optimal feature subsets within a reasonable amount of time. By reducing much of the unnecessary data, feature selection processes can enhance classification accuracy and reduce processing time.

Metaheuristic algorithms are widely used to solve problems of optimization, such as schedule management [14, 15], function optimization [16], and intrusion detection [17]. Metaheuristic algorithms combine random search functions with empirical rules, and many of these methods have been inspired by mechanisms found in nature, such as genetic algorithms (GA) [18–20], based on gene mutation and mating, and particle swarm optimization (PSO) [21], based on the movements of flocks of birds. In 2002, the artificial fish swarm algorithm (AFSA) [22] was proposed to solve problems of optimality by simulating the movement of schools of fish and the intelligence underlying these behaviors. Numerous studies have demonstrated the efficacy of AFSA [14, 17, 23]; however, a number of shortcomings must still be addressed. In [24], various defects were pointed out, such as the likelihood of falling into a local optimum and a lack of multiplicity.

This study proposed a feature selection model combining the modified AFSA (MAFSA) with SVM. MAFSA is used to simulate the mechanism underlying the endocrine system in order to create a different search space for every individual fish in order to enhance the efficiency with which optimal solutions are derived.

Section 2 introduces SVM and the MAFSA. Section 3 outlines the proposed method based on a combination of these principles. Section 4 describes our experiments and results and in Section 5, we draw conclusions and describe our future work.

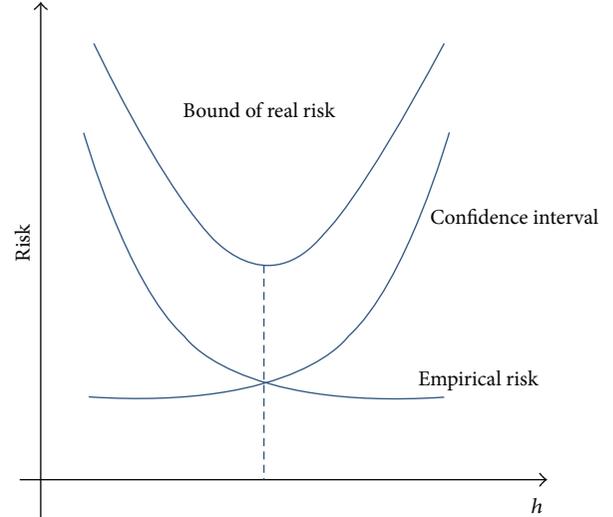


FIGURE 1: Connection between confidence interval and empirical risk.

## 2. Background

**2.1. Support Vector Machine.** Since its introduction by Vapnik in 1995, [5], SVM has become a very popular classifier due to its ease of use and high classification accuracy even when using small training sets. This method of supervised learning is based on the Vapnik-Chervonenkis (VC) dimension and structural risk minimization theory [12, 25]. The VC dimension is used for function sets with only two values: 0 and 1. The function set has  $N$  samples and regardless of how the position of the function samples is changed, the dimension can be separated from the samples. For example, with  $h$  as the maximum number of samples, an increase in  $h$  will lead to an increase in  $N$ . As shown in (1), the upper bound of generalization error ( $R$ ) is the sum of training error ( $R_{\text{Emp}}$ ) and confidence interval (CI):

$$R \leq R_{\text{Emp}} + \Phi\left(\frac{h}{N}\right), \quad (1)$$

where  $R$  represents the generalization error (also called testing error),  $R_{\text{Emp}}$  represents training error, and  $\Phi$  represents the CI. An increase in  $N$  leads to a reduction in the CI and an increase in the VC dimension leads to an increase in  $h$ . When the VC dimension increases, the differences between testing error and training error also enlarge. Thus, reducing the complexity of the classification model and alleviating testing error require that we minimize training error as well as VC dimensions.

The value for  $N$  in the samples is influenced by VC dimension, such that an increase in empirical risk reduced the CI. Conversely, a reduction in empirical risk increases the CI without affecting the total risk. Thus, we must consider empirical risk as well as confidence interval and the tradeoff between them in order to minimize the total risk. Figure 1 [12] illustrates the connection between the CI and empirical risk.

The primary function in SVM involves finding the optimal hyperplane and using it for the classification of data. As

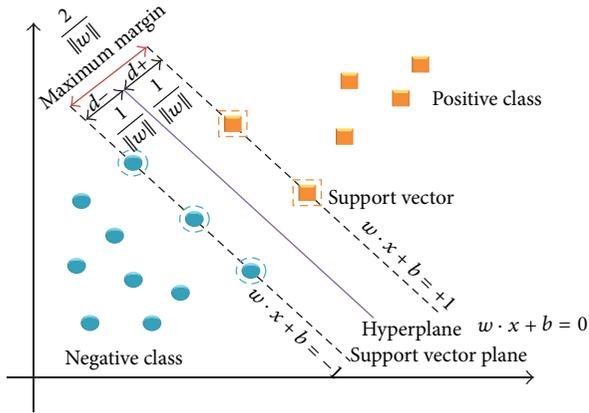


FIGURE 2: Optimal hyperplane.

shown in Figure 2, the optimal has maximal margin to those two classes. In Figure 2, the round and square points in the maximal margin line are the support vectors. This illustrates why the optimal hyperplane (with maximal margin) is able to achieve the highest classification accuracy.

The optimal hyperplane is able to classify only two classes; however, in real world situations, classification problems nearly always involve more than this. Thus, we need to use kernel function ( $\Phi$ ) in order to map the data into a higher VC dimension plane. Three common kernel functions can be used for different situations: radial basis functions (RBFs), polynomials, and sigmoids, as shown in formulas (2), (3), and (4), respectively. The RBF kernel function provides the best performance and versatility of these three methods; therefore, we adopted RBFs as the SVM kernel function.

RBF kernel is

$$\Phi(x_i - x_j) = \exp(-\gamma \|x_i - x_j\|). \quad (2)$$

Polynomial kernel is

$$\Phi(x_i - x_j) = (1 + x_i \cdot x_j). \quad (3)$$

Sigmoid kernel is

$$\Phi(x_i - x_j) = \tanh(kx_i \cdot x_j - \delta). \quad (4)$$

In [25], the authors introduce the soft margin method to process the mislabel examples located within the margin. The soft margin suggests the slack variables which measure the misclassification degree of the samples. And, the penalty parameter,  $C$ , is used to weight the misclassification degree. It was shown that setting the right values of penalty parameter and RBF kernel parameter for SVM could greatly enhance the effectiveness of classification in [26].

**2.2. Artificial Fish Swarm Algorithm (AFSA).** AFSA is a meta-heuristic algorithm combining the concept of random search and empirical rules. AFSA solves optimization problems by simulating the movement of schools of fish and the intelligence underlying these behaviors. There are three types

TABLE 1: Representation of feature sets.

$C$	$G$	$F_1$	$\dots F_i \dots$	$F_n$
-----	-----	-------	-------------------	-------

TABLE 2: Parameters of AFSA.

Parameter name	Formula
Distance ( $F_i, F_j$ )	$\sum_{k=1}^k  F_i(k) - F_j(k) $
Vision ( $F_i$ )	$\frac{\sum_{i=1}^N \sum_{k=1}^N \text{Distance}(F_i, F_k)}{\text{Total number of Fishes}}$
Neighbor ( $F_i$ )	$\{F_k \mid 0 < \text{distance}(F_i, F_k) \leq \text{vision}\}$
Center ( $F_i$ )	$F_{\text{center}}(i) = \begin{cases} 0, & \sum_{k=1}^k F_k(i) < \frac{k}{2} \\ 1, & \sum_{k=1}^k F_k(i) \geq \frac{k}{2} \end{cases}$
Crowd degree ( $F_i$ )	$\frac{\text{Neighbors of } F_i}{\text{Total number of Fishes}}$

of AFSA: follow, swarm, and prey. AFSA repeatedly executes these three functions for every individual fish in order to find an optimal solution. Every fish represents a solution and a fitness number represents the value of each solution. Solving the problem of feature selection requires conversion into a form that can be dealt with by AFSA. Table 1 presents the form of feature sets for each fish.

In Table 1,  $C$  and  $G$  represent the parameters of RBF kernel functions in SVM, and  $F_1$  to  $F_n$  represent the features. A feature value of 0 means that it is not selected, while a value of 1 means that the feature is selected.

The steps involved in feature selection for AFSA are presented as follows.

(1) *Initialization.* Randomly initialize the value of the feature subsets of each fish.

(2) *Evaluate Fitness.* Use the fitness function to evaluate the fitness value of the feature subset of every fish.

(3) *Optimization Steps of Fish Swarm.* Search for an optimal solution by executing the three search steps (follow, swarm, and prey) for each fish.

(4) *Output the Optimal Feature Subset.* If the terminal conditions of algorithm are satisfied, then stop the algorithm and output the optimal feature subset. Otherwise, initiate the algorithm to proceed through another iteration starting at Step (2).

The three optimal search steps of AFSA (follow, swarm, and prey) are presented in the following and the parameters are listed in Table 2.

(i) *Follow.* Executing follow for  $F_i$  would involve comparing its fitness value with that of the best fish in its vicinity (among its neighbors). If the best fitness value among the neighboring fish is better than that of the fish in question and the crowd degree of the neighbor does not exceed the maximum, then the feature subset and parameters are replaced with those of that best neighbor. As long as the replacement of subset and

parameters is successful, then the algorithm executes follow for next fish; otherwise, swarm is executed for  $F_i$ .

(ii) *Swarm*. In the event that the follow function fails for  $F_i$ , then the algorithm would proceed to swarm. At this point, the fitness value of  $F_i$  would be compared with that of the center of the neighboring fish, as shown in Table 2. If the fitness value of the center is better than that of the fish in question and the crowd degree of the center does not exceed the maximum crowd degree, then the feature subset and parameters are replaced with those of the center. As long as the replacement of subset and parameters is successful, then the algorithm executes follow for next fish; otherwise, prey is executed for  $F_i$ .

(iii) *Prey*. In the event that the swarm function fails for  $F_i$ , then the algorithm would proceed to prey. In this step, the algorithm makes random changes to the features of  $F_i$  in order to create new random fish. The maximum number of changes never exceeds the vision of the fish in question. If the fitness value of the random fish exceeds that of  $F_i$ , then feature subset and parameters are replaced with those of the random fish. As long as the replacement of subset and parameters is successful, then the algorithm executes follow for next fish; otherwise, the algorithm would continue searching for random fish until reaching the stipulated maximum number of tries.

The parameters of AFSA are defined as follows.

(i) *Distance*. The distance between  $F_i$  and  $F_j$  was calculated according to the formula presented in Table 2. For example, if both fish have the same dimension  $k$  and if the first feature of both fish was 0, then the distance between  $F_i$  and  $F_j$  would remain the same. In contrast, if the first feature of  $F_i$  was 0 and the first feature of  $F_j$  was 1, then the distance between the two fish would be increased by 1. The total distance between the two fish is equal to the sum of the differences of all  $k$  features.

(ii) *Vision*. The vision of every fish was calculated using the formula in Table 2. Vision also determines the maximum number of random feature changes that will be implemented when initiating *prey*.

(iii) *Neighbor*. The neighbors of  $F_i$  were determined using the formula in Table 2. Any fish  $F_j$  with a distance exceeding 0 but not exceeding the vision of  $F_i$  is deemed a neighbor of  $F_i$ .

(iv) *Center*. The center of  $F_i$  was calculated using the formula in Table 2. The center can be viewed as an individual fish. If, among more than half of the fish in the neighborhood of  $F_i$ , the first feature is 0, then the value of the center is set to 0. If, among more than half of these fish, the first feature is not 0, then the value of the center is set to 1.

(v) *Crowd Degree*. The crowd degree of  $F_i$  was calculated using the formula in Table 2. This parameter represents the density of fish in the vicinity of  $F_i$ .

(vi) *Maximum Crowd Degree*. In the execution of steps *follow* and *swarm*, we sought to prevent the agglomeration of all fish at the same point by designating that if the crowd degree

of  $F_i$  exceeded the maximum, then no other fish would be permitted to approach this location. In other words, no other fish would be able to replace its feature subset using with that of  $F_i$ .

(vii) *Maximum Number of Attempts*. This is the maximum number of times that *prey* could be executed.

### 3. Modified Artificial Fish Swarm Algorithm

Researchers have revealed a number of shortcomings in the function of AFSA, such as the likelihood of falling into a local optimum and the lack of multiplicity. This study developed a modified artificial fish swarm algorithm with two fundamental changes: dynamic vision and improvement of the ability to search for the best fish swarm.

3.1. *Dynamic Vision*. The vision parameter plays a crucial role in the performance of AFSA, by determining the number of neighboring fish with which the target fish will interact, which largely determines the success of steps *follow* and *swarm*. Setting the vision parameter higher increases the likelihood of finding fish with higher fitness levels; however, this can lead to the centralization of the fish swarm to a particular location. This also tends to limit the diversity of species, which can easily fall into a local optimum. Setting the vision parameter lower reduces the number of neighboring fish (and thus the likelihood of finding a fish with higher fitness values) and causes the school to scatter while increasing the diversity of species. This increases the search space as well as the likelihood of finding an optimal solution; however, the time required to reach convergence is extended. Finding a reasonable balance with regard to the assignment of the vision parameter can be troublesome. To overcome this difficulty, this study developed a mechanism referred to as dynamic vision, in which each fish is assigned visions parameters suitable to its conditions. For example, individual fish with a lower fitness value require greater vision in order to find a solution quickly. Conversely, fish with a higher fitness value require a smaller vision parameter in order to enhance local searches.

We employed the endocrine-based formula, as outlined in [26], in order to provide dynamic vision. Individual fish obtain their own vision parameter values according to their fitness values. For example, if the fitness value is above the average, then the vision parameter is decreased, and vice versa. The endocrine-based formula is presented as follows:

$$EM(i) = f_1 \left( \frac{f_{\max} - f_i}{f_{\max} - f_{\text{avg}}} \right) \cdot \left[ \frac{\pi}{2} + f_2 \left( f_i - \frac{f_{i-1} + f_{i+1}}{2} \right) \right], \quad (5)$$

$$\text{Vision}(i) = \text{Vision}(\text{static}) \cdot EM(i) \cdot CV, \quad (6)$$

where  $EM(i)$  represents the endocrine of fish  $F_i$ ,  $f_{\max}$  represents the maximum fitness value of the fish in the school, and  $f_{\text{avg}}$  represents the average fitness value in the school.  $f_{i-1}$  represents the fitness values of fish  $f_{i-1}$ ;  $f_{i+1}$  represents the fitness value of fish  $f_{i+1}$ . In order to adjust the range of

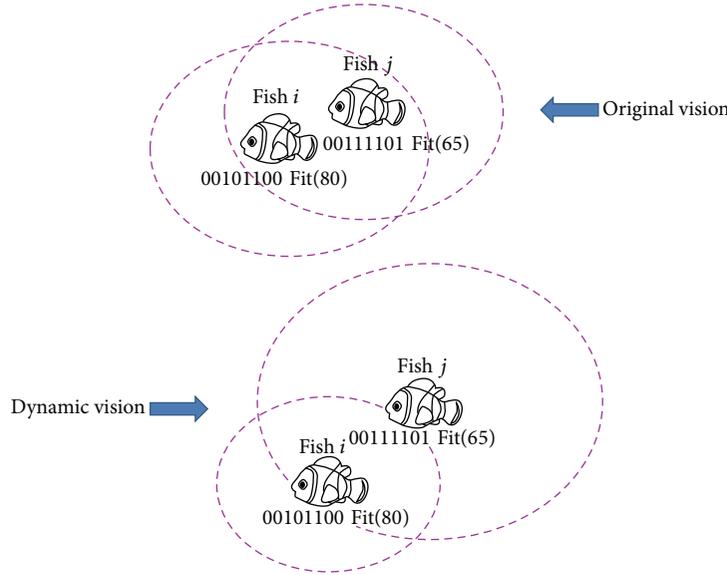


FIGURE 3: Original vision and dynamic vision.

the endocrine system,  $f_1(x) = \text{atan}(x)$ ,  $f_2(x) = \text{atan}(-x)$ . In formula (6),  $\text{vision}(i)$  represents the original vision parameter value calculated as the average distance between each fish and CV is the adjustment constant. Figure 3 presents an example of dynamical vision. In the original vision parameter, fish  $i$  and  $j$  would have had the same vision, regardless of their fitness values. Following adjustment using the endocrine-based formula, the fitness value of fish  $i$  is improved to 80, which decreases its vision in order to enhance its ability to perform local searches. In contrast, the fitness value of fish  $j$  dropped to 65, which enhanced the vision of fish  $j$ , thereby enhancing global search capacity in order to obtain solutions more quickly.

**3.2. Searching for the Best Fish Swarm.** This study employed a simple method of searching for the best fish swarm to enhance local search ability and prevent falling into a local optimum. After each iteration has finished, the algorithm copies the fish with the best fitness value into a fish swarm and then executes a simple local search to enhance the possibility of finding an optimal solution. Four parameters were added to facilitate this change: best fish number (BFN), best fish search number (BSN), best fish mutation rate (BMR), and minimum change number (MCN).

BFN represents the number of fish copied. For example, setting  $\text{BFN} = 5$  would cause the five best fish to be copied to the best fish swarm. BSN represents the local search number for each fish in the best fish swarm. Setting  $\text{BSN} = 10$  would cause the algorithm to execute local ten searches for each fish in the best fish swarm. BMR represents the mutation rate of local searches and the MCN represents the minimum number of feature changes in each local search. For example, suppose that feature number was set to 8, BMR was 0.1, and MCN was set to  $1, 8 * 0.1 = 0.8$ . In this case, 0.8 is less than the value of MCN (1); therefore, 1 random change would be executed in every local search in the best fish swarm. Figure 5 presents

TABLE 3: UCI dataset.

Number	Dataset	Number of classes	Number of features	Number of instances
1	Bupa	2	6	345
2	Pima	2	8	768
3	Glass	6	9	214
4	Vowel	11	10	528
5	Heart	2	13	270
6	Australian	2	14	690
7	Vehicle	4	18	846
8	Robot	4	24	5456
9	German	2	24	1000
10	Sonar	2	60	208

an example of a local search intended to assemble a swarm of the best fish in the vicinity of a given fish. After five local searches among this collection of high scoring fish, the fish that had previously been identified as best fish 1 is replaced by the fish with the best feature subset. Figure 4 presents a flowchart of the MAFSA.

## 4. Experiment Results

**4.1. Experiment Environment.** To compare the performance of MAFSA and AFSA, we used datasets commonly used in machine learning, the UCI dataset [27]. Ten datasets with different numbers of records, different features, different classes, and from different fields are presented in Table 3. The parameters of MAFAS and AFSA are presented in Table 4.

As shown in Table 4, AFSA does not have the BFN, BSN, BMR, or MCN parameters, which are used only for assembling a swarm of the best fish when using MAFSA.  $C$  and  $G$  are the parameters of SVM with  $C$  representing

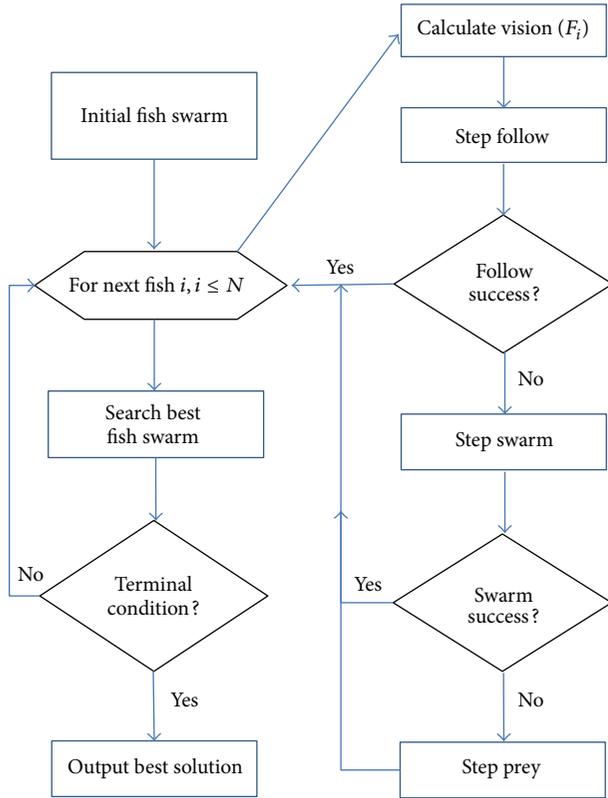


FIGURE 4: Flowchart of modified artificial fish swarm algorithm (MAFSA).

TABLE 4: Parameters of AFSA and MAFSA.

Parameters	AFSA	MAFSA
Fish number	30	30
Maximum try number of prey	20	20
Maximum crowd degree	0.5	0.5
CV	N/A	0.5
BFN	N/A	5
BSN	N/A	20
BMR	N/A	0.15
MCN	N/A	1
C	[0.01, 1024]	[0.01, 1024]
G	[0.00001, 8]	[0.00001, 8]

the penalty parameter and  $G$  representing the parameters of the RBF kernel function. This used libsvm [28] as a classifier. In [26], it was shown that setting the right parameters for SVM could greatly enhance the effectiveness of classification. As mentioned in Section 2.2, the proposed algorithm determines the optimal values for these two parameters. To ensure the reliability of the experiment, we used tenfold cross-validation.

Figure 6 presents a flowchart of the proposed model used for feature selection using MAFSA and SVM. To achieve

tenfold cross-validation, the dataset was divided into two parts, training data and testing data. Various feature subsets were created along with throughout the process of MAFSA. After deleting the features and corresponding data which were not selected, the training data was input into SVM to build a classification model. SVM then outputs a value representing the accuracy of classification for each feature subset in the form of a fitness value. MAFSA would stop and output the optimal feature subset only after the terminal conditions were satisfied. In this case, the terminal condition of each fold was a lack of changes in the optimal features subset for a period of 1 hour.

**4.2. Experiment 1: Feature Selection without Parameter Optimization.** To compare the effectiveness of AFSA and MAFSA, we examined three experiment results: (1) the dataset without features, (2) the dataset after feature selection using AFSA and SVM, and (3) the dataset after feature selection using MAFSA and SVM. Experiment 1 did not involve search for the optimal SVM parameters  $C$  and  $G$ . In other words, Experiment 1 used the default values of parameters  $C$  and  $G$  found in libsvm. The results are presented in Table 5.

In Table 5, the first group is the dataset without feature selection, which did not provide satisfactory results with regard to classification accuracy or the number of selected features. Furthermore, because this group used data without feature selection, the number of selected features was the same as the in the original. The second group was the result for a dataset that underwent feature selection using AFSA, and the third group was the result after feature selection using MAFSA. The classification accuracy of Groups 2 and 3 were far better than those of Group 1, and the number of selected features was far less than that of Group 1. MAFSA provided the best classification accuracy in all ten of the datasets; however, in five of the datasets, the results were on par with those obtained using AFSA. MAFSA resulted in fewer selected features in eight of the ten datasets.

In five of the datasets with fewer features, MAFSA was unable to exceed the classification accuracy of AFSA. Nonetheless, MAFSA still produced fewer features. Experiment 1 did not involve SVM parameters  $C$  or  $G$ ; therefore, AFSA and MAFSA were able to search nearly all of the possible feature subsets. In some folds of the datasets, more than one of the feature subsets provided the same best classification accuracy. For this reason, MAFSA resulted in fewer selected features but maintained the same classification accuracy, thereby demonstrating the superior search ability of MAFSA.

**4.3. Experiment 2: With Feature Selection and Parameter Optimization.** Experiment 2 compares the performance of AFSA and MAFSA in which SVM parameters  $C$  and  $G$  were also considered. In other words, AFSA and MAFSA searched not only for the optimal feature subset but also the optimal parameters of  $C$  and  $G$ . The results are presented in Table 6.

In Table 6, MAFSA is shown to have higher classification accuracy in eight of the ten datasets as well as fewer selected features in six of the ten datasets. After determining SVM parameters  $C$  and  $G$ , the classification accuracy in all ten of

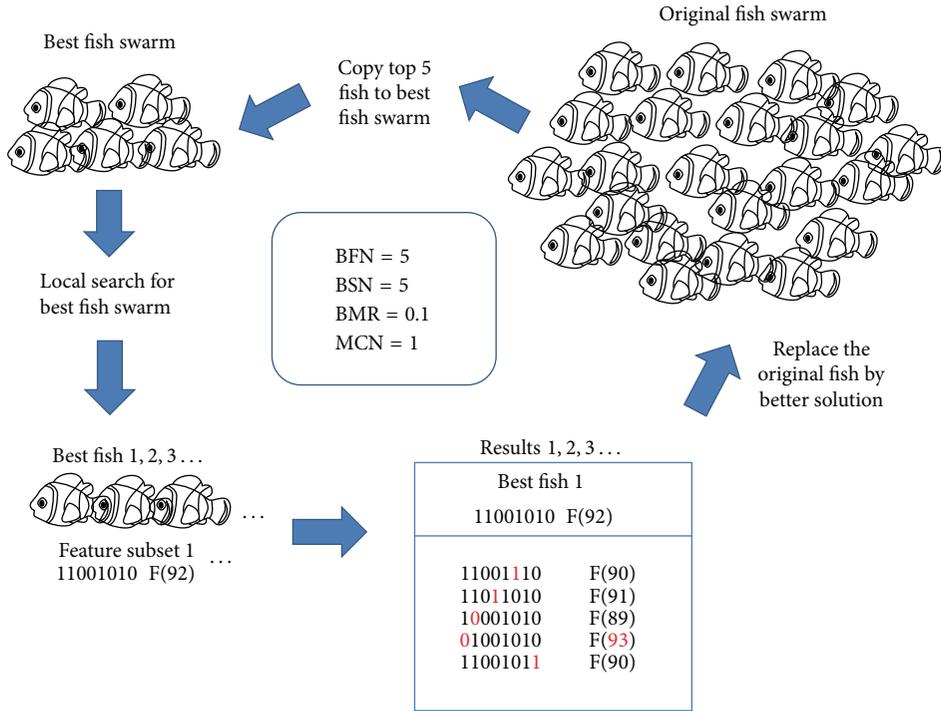


FIGURE 5: Process of assembling swarm of best fish.

TABLE 5: Result of Experiment 1.

Dataset	Number of original features	Group 1		Group 2		Group 3	
		No feature selection		Feature selection by AFSA		Feature selection by MAFSA	
		Number of selected features	Average accuracy rate (%)	Number of selected features	Average accuracy rate (%)	Number of selected features	Average accuracy rate (%)
Bupa	6	6	58.85	3.3	59.73**	3.2*	59.73**
Pima	8	8	77.21	5*	80.47**	5.6	80.47**
Glass	9	9	48.11	4.8	64.89**	4.4*	64.89**
Vowel	10	10	71.91	9	77.57	8.5*	77.67**
Heart	13	13	81.85	7.2	94.07**	7.1*	94.07**
Australian	14	14	85.79	8	88.26**	7.7*	88.26**
Vehicle	18	18	71.98	12.4*	75.41	12.4*	75.53**
Robot	24	24	85.90	17*	86.73	17.6	87.48**
German	24	24	74.90	14.7	82.90	12.4*	83.50**
Sonar	60	60	73.19	29.7	93.33	27.2*	94.28**

\*It indicates the fewest number of selected features.

\*\*It indicates the highest classification accuracy.

the datasets improved significantly. MAFSA was shown to have higher classification accuracy in most of the datasets; however, it little differentiated AFSA from MAFSA. We therefore used statistical analysis to verify the improvements obtained by using MAFSA. Data analysis was performed using SPSS. We employed the Friedman test, which is a statistical analysis method used to test the performance of algorithms. The Friedman statistics test is used for nonparametric statistical verification of whether  $k$  related samples differ significantly. We use the classification accuracy of AFSA and MAFSA in the ten datasets in Table 6 to verify

the superior performance of MAFSA. The results indicated a significance value of 0.02, which is less than 0.05; that is, the test provides 95% CI that MAFSA provides an improvement in classification accuracy over the results obtained using AFSA.

### 5. Conclusion and Future Works

This research proposed a modified version of the artificial fish swarm algorithm in conjunction with support vector machine for feature selection. MAFSA differs in its use of

TABLE 6: Result of Experiment 2.

Dataset	Number of original features	Feature selection by AFSA		Feature selection by MAFSA	
		Number of selected features	Average accuracy rate (%)	Number of selected features	Average accuracy rate (%)
Bupa	6	4.1	84.94	3.9*	85.22**
Pima	8	4.8	83.72	4.2*	83.85**
Glass	9	4.9*	89.69	5.1	91.58**
Vowel	10	7.8	100**	7*	100**
Heart	13	6.1*	97.03	6.2	97.77**
Australian	14	7.1	93.62**	5.7*	93.33
Vehicle	18	11.2	91.01	11*	92.08**
Robot	24	7.6*	96.57	8.3	97.25**
German	24	14.2	83.7	13.7*	84.6**
Sonar	60	27.2*	99.04	29.1	100**

\*It indicates the fewest number of selected features.

\*\*It indicates the highest classification accuracy.

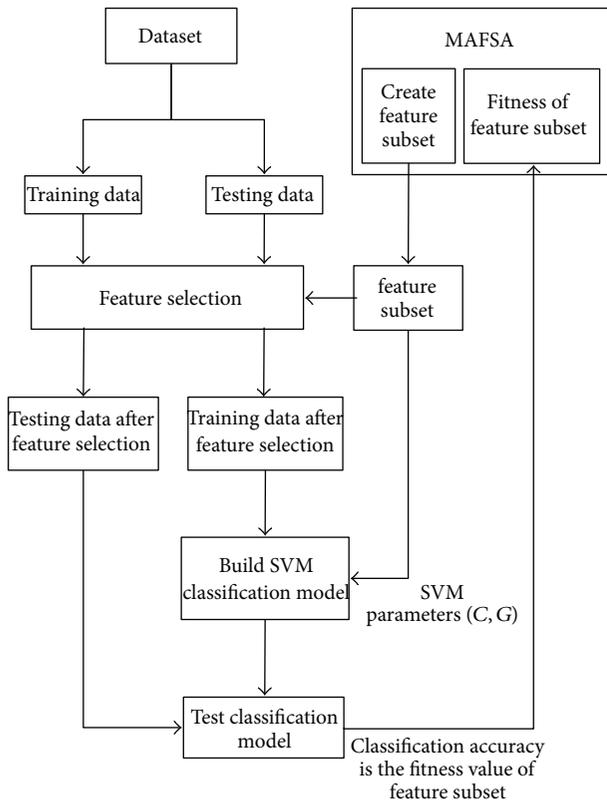


FIGURE 6: Flowchart of feature selection of MAFSA and SVM.

an endocrine-based formula to provide dynamic vision to assign an appropriate search space for each fish. We also included a mechanism by which a swarm of the best fish can be assembled to enhance local search ability. Experiments demonstrate that MAFSA is superior to AFSA with regard to classification accuracy as well as the number of selected features.

Nonetheless, MAFSA still has room for improvement. For example, all of the parameters of MAFSA could be selected dynamically to further enhance adaptability to different datasets. We designed binary-coded algorithms to deal with the encoding of feature selection. We expect to apply MAFSA to the optimization problem of continuum. We expect that MAFSA could be used to solve real world optimization problems in a wide range of applications.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] J. K. Kishore, L. M. Patnaik, V. Mani, and V. K. Agrawal, "Application of genetic programming for multiclass pattern classification," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 242–258, 2000.
- [2] R. Stevens, C. Goble, P. Baker, and A. Brass, "A classification of tasks in bioinformatics," *Bioinformatics*, vol. 17, no. 2, pp. 180–188, 2001.
- [3] E. Namsrai, T. Munkhdalai, M. Li, J.-H. Shin, O.-E. Namsrai, and K. H. Ryu, "A feature selection-based ensemble method for arrhythmia classification," *Journal of Information Processing Systems*, vol. 9, no. 1, pp. 31–40, 2013.
- [4] J. R. Quinlan, *C4.5: Programs for Machine Learning*, The Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann, San Mateo, Calif, USA, 1993.
- [5] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.
- [6] Y. S. Hwang, J. B. Kwon, J. C. Moon, and S. Je Cho, "Classifying malicious web pages by using an adaptive support vector machine," *Journal of Information Processing Systems*, vol. 9, no. 3, pp. 395–404, 2013.

- [7] J. Uddin, R. Islam, and J. Kim, "Texture feature extraction techniques for fault diagnosis of induction motors," *Journal of Convergence*, vol. 5, no. 2, pp. 15–20, 2014.
- [8] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000.
- [9] M. Malkawi and O. Murad, "Artificial neuro fuzzy logic system for detecting human emotions," *Human-Centric Computing and Information Sciences*, vol. 3, article 3, 2013.
- [10] A. K. Gopalakrishna, "A subjective job scheduler based on a backpropagation neural network," *Human-Centric Computing and Information Sciences*, vol. 3, article 17, 2013.
- [11] H. Lee, H. Kim, and J. Seo, "An integrated neural network model for domain action determination in goal-oriented dialogues," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 259–270, 2013.
- [12] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [13] R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*, IEEE Press Series on Computational Intelligence, Wiley, Oxford, UK, 2008.
- [14] S. Farzi, "Efficient job scheduling in grid computing with modified artificial fish swarm algorithm," *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, pp. 13–18, 2009.
- [15] O. Mirzaei and M.-R. Akbarzadeh-T, "A novel learning algorithm based on a multi-agent structure for solving multi-mode resource-constrained project scheduling problem," *Journal of Convergence*, vol. 4, no. 1, pp. 47–52, 2014.
- [16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [17] T. Liu, A.-L. Qi, Y.-B. Hou, and X.-T. Chang, "Feature optimization based on artificial fish-swarm algorithm in intrusion detections," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, vol. 1, pp. 542–545, Wuhan, China, April 2009.
- [18] J. H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [19] H. T. T. Binh and S. H. Ngo, "All capacities modular cost survivable network design problem using genetic algorithm with completely connection encoding," *Human-Centric Computing and Information Sciences*, vol. 4, article 13, 2014.
- [20] H. T. T. Binh, "Multi-objective genetic algorithm for solving the multilayer survivable optical network design problem," *Journal of Convergence*, vol. 5, no. 1, pp. 20–25, 2014.
- [21] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, Australia, December 1995.
- [22] X. L. Li, Z. J. Shao, and J. X. Qian, "Optimizing method based on autonomous animats: fish-swarm Algorithm," *System Engineering Theory and Practice*, vol. 22, no. 11, pp. 32–38, 2002.
- [23] J.-P. Wang and M.-J. Hu, "A solution for TSP based on artificial fish algorithm," in *Proceedings of the International Conference on Computational Intelligence and Natural Computing (CINC '09)*, pp. 26–29, Wuhan, China, June 2009.
- [24] C. Wang, T. Xue, and L. Sun, "A survey of Artificial Fish-swarm Algorithm and its Improve Technique," 2014, <http://www.paper.edu.cn/html/releasepaper/2008/03/581/>.
- [25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [26] K.-C. Lin, S.-H. Hsu, and J. C. Hung, "Adaptive SVM-based classification systems based on the improved endocrine-based PSO algorithm," *Lecture Notes in Computer Science*, vol. 7669, pp. 543–552, 2012.
- [27] UC Irvine Machine Learning Repository, 2014, <http://archive.ics.uci.edu/ml/>.
- [28] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2014, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

