Journal of Information Hiding and Multimedia Signal Processing ©2023 ISSN 2073-4212
Ubiquitous International Volume **14**, Number **4**, December **2023**

Optimal Parameter-Feature Selection Using Binary PSO for Enhanced Classification Performance

Truong-Giang Ngo
Faculty of Computer Science and Engineering, Thuyloi University, Hanoi, Vietnam
giangnt@tlu.edu.vn

Hoan Nguyen-Cong[1,2]
[1]Software Engineering Department,
University of Information Technology, Vietnam
[2]Vietnam National University, Ho Chi Minh City 700000, Vietnam
hoannc@uit.edu.vn

Thi-Thanh-Tan Nguyen
Information Technology Faculty,
Electric Power University, Hanoi 100000, Vietnam,
tanntt@epu.edu.vn

Thi-Kien Dao[a,b,c*],
[a] Multimedia Communications Lab., University of Information Technology, Vietnam
[b] Vietnam National University, Ho Chi Minh City 70000, Vietnam
[c] Fujian Provincial Key Laboratory of Big Data Mining and Applications,
Fujian University of Technology, Fuzhou 350118, China
[*]Corresponding authors: kiendt@uit.edu.vn;

Received September 2023; revised November 2023; accepted November 2023

Abstract. *Feature selection and classifier parameter optimization are critical for improving classifier performance. Traditionally, these two problems have been addressed separately. However, recent advancements in evolutionary optimization computing technology have allowed for the simultaneous optimization of feature selection and parameter tuning. In this paper, we propose a novel approach called PSO-SVM, which combines binary PSO with SVM parameter optimization. We evaluate the effectiveness of the PSO-SVM scheme through extensive experiments that demonstrate its ability to effectively identify feature subsets and SVM parameters that are well-suited for the given task, resulting in superior classification outcomes. Furthermore, we compare our algorithm with other algorithms and find that the PSO-SVM algorithm offers a wider range of feature reduction capabilities, enabling a more efficient representation of input data. Additionally, the PSO-SVM algorithm demonstrates higher computational efficiency, making it a more practical choice for real-world applications.*
Keywords: Parameter optimization, Feature selection, PSO-SVM algorithm; Intelligent optimization algorithm; Binary particles swarms optimization.

---

1. **Introduction.** The classification problem is a fundamental challenge in pattern recognition, involving tasks such as selecting the appropriate classifier model [1], feature selection, and optimizing classifier parameters [2]. As research in pattern recognition progresses, the complexity of the objects under study has increased [3], resulting in higher-dimensional feature spaces [4]. Many high-dimensional datasets contain redundant or

noisy features, which can negatively impact classification accuracy and significantly increase computational complexity [5]. Therefore, feature selection algorithms are necessary to identify a feature subspace with good separability, reducing dimensionality and mitigating the complexity of machine learning processes [6]. In recent years, there has been a growing trend to simultaneously address feature selection and parameter optimization problems [7]. Traditionally, these two problems were studied independently, but researchers have recognized the benefits of integrating them. For example, genetic algorithms and particle swarm optimization have been employed to synchronize feature selection and classifier parameter optimization, yielding promising results. This research paper proposes a concurrent approach that combines feature selection and parameter optimization using Binary particles swarms optimization (BPSO) [8]. PSO is an emerging optimization technique inspired by artificial life and evolutionary computation theories [9]. It iteratively updates the positions of particles in a swarm, guided by the best solution found by any individual particle and the overall swarm. PSO has been successfully applied in various fields, including pattern recognition and data mining [10]. To address the challenge of synchronizing optimal feature selection, this paper presents a PSO-SVM algorithm that combines binary PSO for the feature selection with synchronization optimization. The algorithm seamlessly integrates feature selection and parameter optimization, improving classification performance, reducing computational time and complexity, and enhancing the interpret ability of the resulting models [11]. This research paper contributes a concurrent framework that uses Binary PSO to integrate feature selection and parameter optimization, a fitness function that efficiently evaluates classification performance, and an experimental evaluation that compares the proposed approach with existing methods. The experimental results demonstrate the effectiveness of the proposed approach in improving classification accuracy and interpret ability. Overall, this research paper presents a novel approach that synchronizes feature selection and parameter optimization, improving classification performance and advancing the field of pattern recognition. The proposed scheme offers enhanced feature reduction capabilities and computational efficiency compared to existing methods, opening up new avenues for optimizing classifier performance.

2. Related Work.

2.1. Feature Selection Techniques.

Techniques for selecting elements that are the most pertinent are essential for enhancing classification performance. Filter, wrapper, and embedded methods are some of these strategies. While wrapper approaches utilize a particular classification algorithm to evaluate feature subsets, filter methods use statistical metrics or information theory to analyze feature significance. Feature selection is integrated into the learning algorithm itself in embedded approaches. Parameter optimization techniques, on the other hand, seek to maximize classification performance by determining the ideal values for algorithm parameters [12]. Conventional techniques, such as random and grid search, investigate a predetermined range of parameter values; however, they can be computationally costly and may not always identify the global optimum. Metaheuristic techniques such as PSO and GA have been widely employed to tackle these problems. By effectively exploring the parameter space, these methods enhance the optimization procedure.

Table 1 provides a general overview and the advantages and applications of the selection technique that may vary depending on the specific problem and dataset. Both feature selection and parameter optimization are essential for enhancing classification performance.

Table 1. An outlining different feature selection techniques

Technique	Description	Advantages	Applications
Filters Methods [7]	Evaluate feature relevance based on statistical measures or information theory	Computationally efficient, independent of classification algorithm	Text mining, bioinformatics
Wrapper Methods [13]	Use a specific classification algorithm to evaluate subsets of features based on their performance	Incorporates classification algorithm feedback, can capture feature interactions	Medical diagnosis, image recognition
Embedded Methods [7]	Incorporate feature selection within the learning algorithm itself	Simultaneously optimizes feature selection and model parameters	Natural language processing, sentiment analysis
Genetic Algorithms (GA) [14]	Optimization algorithm inspired by biological evolution	Can handle large search spaces, global optimization capability	Data mining, financial forecasting
Particles Swarms Optimization (PSO) [9]	Optimization algorithm inspired by social behavior of bird flocking	Efficiently explores parameter space, fast convergence	Image segmentation, pattern recognition

Feature selection techniques help identify the most relevant features, reducing dimensionality and computational complexity. Parameter optimization methods ensure that the algorithm is fine-tuned to achieve the best possible classification results. By combining these two techniques, researchers can further improve classification accuracy and efficiency.

2.2. **Particles Swarms Optimization (PSO).** The PSO algorithm is a computational technique inspired by the behavior of bird predation [9]. It was proposed by Eberhart and Kennedy and is used to search for optimal solutions through iterative processes. Unlike genetic algorithms, PSO does not employ crossover and mutation operations. Instead, particles in the algorithm follow the current optimal particle to explore the solution space. PSO has several advantages over genetic algorithms, such as its simplicity, ease of implementation, and fewer adjustable parameters. Consequently, PSO has been widely applied in various domains, including function optimization, neural network training, and fuzzy system control. In PSO, each particle represents a potential solution to the optimization problem and is treated as a point in the search space. The fitness of each particle is evaluated using an evaluation function. By sharing information about the current optimal particle, particles collectively explore the solution space. In order to represent random solutions, the PSO initializes the particle swarm with a collection of random particles that looks for the best answer through successive rounds. Particles use the evaluation function to compute their fitness values and update their current positions depending on their velocities at each iteration. The particles then use the following speed update formula to update their velocities.

$$Vel_{id} = Vel_{id} + co1 \times rand() \times (p_{id} - S_{id}) + co2 \times rand() \times (p_{gd} - S_{id}) \qquad (1)$$

Where Vel is a particles possess velocities, determine the direction and magnitude of their movement.

$$S_{id} = S_{id} + Vel_{id} \tag{2}$$

where S_i is current position in the solution; P_id is the local optima particle's; Every particle has attributes, denoted as $S_i = (s_{i1}, s_{i2}, .., s_{iD})$, which represent its current position in the solution space. $P_{id} = (p_{i1}, p_{i2}, .., p_{iD})$ represents the local optimal particle, existing the best point that particle in search space so far, $P_{gd} = (p_{g1}, p_{g2}, ..p_{gD})$ represent the global optimal value of the entire particle swarm, that is, the best point the entire particle swarm has hit in the search space so far. co_1 and co_2 are two positive constants, called learning factors. rand represents a random number between 0 and 1. Formula (2) is a position update formula of a particle. In order to adapt to the applied PSO algorithm in binary PSO, each particle is encoded as a binary vector. In binary particles, velocity defines the probability that each position of the particle is assigned a value of 1, so the velocity is converted to the interval [0.0, 1.0] by a transfer function. The sigmoid function was used in this study. The binary particle update formula is as follows:

$$p_{ij} = \begin{cases} 1, & if\ rand() \leq S\left(v_{ij}\right) \\ 0, & otherwise \end{cases}, \tag{3}$$

In order to obtain better optimization efficiency and effect, this study made some improvements to the PSO algorithm, and mutated the worst 10% particles in each iteration. Experiments show that these 10% particles waste computing resources. After mutation, the particle swarm can find the optimal value faster, and avoid the situation where it is easy to gather in the local optimum.

2.3. **Support Vector Machine (SVM).** Statistical pattern recognition methods have traditionally relied on the assumption that the number of samples available is sufficient to ensure that the proposed method's performance can be theoretically guaranteed, provided that the number of samples approaches infinity [15]. However, in reality, the number of available samples is often limited, which poses a significant challenge to the development of effective pattern recognition methods. In recent years, the field of statistical learning theory has emerged as a specialized pattern recognition theory that focuses on small sample learning [11]. It has provided a more robust theoretical framework for studying statistical pattern recognition and a broader range of machine learning problems when faced with limited samples. One of the most promising new pattern recognition methods to emerge from this field is the support vector machine (SVM), which has proven to be highly effective in addressing the challenges of small sample, non-linear, and high-dimensional pattern recognition [16]. SVM has several unique advantages that make it particularly well-suited for small sample learning and classification. The following section provides a brief overview of the principles of SVM for pattern learning and classification [12]. Suppose that the input pattern set D contains Mn-dimensional samples that are divided into two categories [1]. Samples belonging to the first category are labeled as 1, while those belonging to the second category are labeled as -1. This can be mathematically expressed as:

$$D = \begin{cases} (x_i, y_i) \,| i \in \{1, 2, \ldots, M\} \\ x_i \in R^n, y_i \in \{1, -1\} \end{cases}, \tag{4}$$

Assume that the samples in the input pattern set D can be correctly classified by a hyperplane described as $(w \times x) - b = 0$ that is expressed as follows.

$$\begin{cases} (w \times x_i) - b \geq 1\ ,\ if\ (\ y_i = 1) \\ (w \times x_i) - b \leq 1,\ otherwise\ ,\ if\ y_i = -1 \end{cases}, \tag{5}$$

p_C^i	$\dots p_C^i \dots$	$p_C^{n^c}$	p_γ^1	$\dots p_\gamma^i \dots$	$p_\gamma^{n^\gamma}$	p_F^1	$\dots p_F^i \dots$	$p_F^{n_F}$

FIGURE 1. An example of a encode particle

The above formula can be simplified to $y_i\left[(w \times x_i) - \mathrm{b}\right] \geq 1$. It can be easily concluded that solving the optimal hyperplane problem that satisfies condition can be expressed.

$$Minimize\Phi\left(\mathrm{w}\right) = w^2 \tag{6}$$

The optimization problem is transformed into the form as follows.

$$f\left(x\right) = sign\left(\sum_k a_k y_k K\left(x_k, x\right) + b\right) \tag{7}$$

Among them, $K(x, y)$ is the kernel function, b is the threshold value determined according to the training samples, and a_k is determined by the quadratic programming. In the SVM, the decision surface's characteristics are determined by the choice of the kernel function. The SVM utilizes this decision surface to classify samples effectively. There are various types of kernel functions available for use in SVM, including linear kernel functions, polynomial kernel functions, radial basis kernel functions, and more. Each kernel function has its own unique properties and is suitable for different types of data and classification tasks [2]. The selection of the kernel function is crucial in determining the SVM's performance and its ability to accurately classify samples.

3. **Optimal Feature selection with parameter synchronization.** A powerful tool for optimal feature selection with BPSO[17] is designed to optimize both the feature subset and the SVM parameters simultaneously. This section discusses the particle design, fitness function, and algorithm flow of the PSO-SVM algorithm. Particle design is implemeted with the PSO-SVM algorithm, particles are represented by a binary bit string. The structure of the particle is determined based on the kernel function used in the SVM [18]. For instance, in this study, the radial basis function (RBF) is used as the kernel function for the SVM is defined as follows.

$$K\left(s, z\right) = \exp\left(-\gamma s - z^2\right), \gamma > 0 \tag{8}$$

For SVM with RBF kernel function, there are two parameters that need to be optimized: kernel parameter $n_\gamma > 0$ and penalty parameter (penalty parameter) $C > 0$. In addition, feature selection must be performed simultaneously. Therefore, the particle should contain three parts, namely parameter C, parameter γ, and feature mask, as shown in Figure 1. The first n_C bits of the particle represent the parameter C, the middle n_γ bits represent the parameter γ, and the last n_F bits represent the feature mask. n_C and n_γ are determined according to the accuracy requirements, and n_F is determined by the number of features in the data set. Schematic diagram of particle is structured as follows.

The binary bit strings representing the parameters C and γ in the particles can be converted into their decimal equivalents using Eq. (9), where p represents the decimal value of the parameter, min_p and max_p represents the minimum and maximum value of the parameter, and l represents the length of the binary bit string for the parameter C and γ are n_C and n_γ, respectively; d denote the decimal value of the binary bit string representing the parameter. The precision of the parameter can be adjusted by changing the length of the bit string, while the range of the parameter can be set by specifying its minimum and maximum values.

Step 1:	Initialize the particle swarm with random binary bit strings.
Step 2:	Evaluate the fitness of each particle using the SVM classifier with the current feature subset and SVM parameters.
Step 3:	Update the personal best position and the global best position of each particle based on the fitness value.
Step 4:	Update the velocity and position of each particle using the PSO algorithm.
Step 5:	Evaluate the fitness of each particle with the updated feature subset and SVM parameters.
Step 6:	Repeat steps 3-5 until the stopping criterion is met.
Step 7:	Select the particle with the highest fitness value as the final solution.

$$p = min_p + \frac{max_p - min_p}{2^l - 1} \times d \tag{9}$$

In the feature mask section, each binary bit corresponds to a feature in the feature set. A bit of 1 indicates that the corresponding feature is selected in the feature subset, while a bit of 0 indicates that the corresponding feature is not included in the selected feature subset. The fitness function of the PSO-SVM algorithm evaluates the performance of the SVM classifier by measuring its classification accuracy on a given dataset. It is calculated as the ratio of correctly classified samples to the total number of samples in the dataset. A higher fitness value indicates better performance of the SVM classifier [19]. The goal of the algorithm is to optimize the SVM parameters of the feature subset to improve classification accuracy while minimizing the number of selected features. As a result, a smaller number of features corresponds to a higher fitness value. Eq. (10) defines the particle fitness function used in this study.

$$fitness = w_a \times svm - accuracy + w_f \times \left(\sum_{i=1}^{nF} f_i \right)^{-1} \tag{10}$$

Among them, w_a represents the weight of classification accuracy; svm-accuracy represents the classification accuracy of SVM; w_f represents the weight of the inverse of the number of features; f_i represents the corresponding bit of the i-th feature in the feature mask, 1 represents the feature is selected, and 0 represents not Select; w_a and w_f are determined according to actual needs. The PSO-SVM algorithm works by iteratively updating the particle positions and velocities based on the fitness function. The algorithm flow can be summarized as follows:

The PSO-SVM algorithm's main advantage is its ability to simultaneously optimize the feature subset and SVM parameters, leading to improved classification accuracy and reduced computational complexity.

4. **Experimental Results and Discussion.** Experimental data and experimental environment is setted as for assessing the effectiveness of the algorithm, several UCI datasets were utilized for experimental research. The datasets' names and relevant information are presented in Table 1. All experiments were conducted on a computer equipped with an Intel Pentium 4 3.0G CPU and 512M memory. The algorithm was implemented using Matlab programming, and the SVM utilized LIBSVM [12].

TABLE 2. UCI Datasets Used in the Experiment

Names	Number of classes	Number of instances	Nominal features	Numeric features	Total features
The cases diabetes	2	768	0	8	8
Heart disease	227	0	7	6	13
Ionosphere	2	351	0	34	34
Sonar	2	208	0	60	60
Iris	3	150	0	4	4
Vehicle	4	990	0	18	18

The UCI datasets chosen for the experiment vary in terms of the number of instances, features, and classes. This diversity ensures a comprehensive evaluation of the algorithm's performance across different types of datasets. The experimental environment's specifications indicate the computational resources available for running the experiments. The Matlab programming language and the LIBSVM library were selected for implementation, ensuring reliable and efficient execution of the algorithm.

4.1. **Experimental Evaluation.** Cross-validation was used in the experimental evaluation of the PSO-SVM algorithm. The PSO-SVM technique was applied k times to each of the k subsets randomly selected from the dataset. The remaining *k-1* subsets were used as the training set, and one subset was used as the test set for each run. The final classification result for the dataset was calculated by averaging the outcomes of the k runs. In this investigation, k was set to 10. The performance of the PSO-SVM algorithm was compared to that of SVM without feature selection or parameter optimization, as well as the GA-SVM technique suggested in [1], to provide a comprehensive comparison. The cross-validation technique was also used to assess the SVM and GA-SVM approaches. The performance of the PSO-SVM algorithm was evaluated using the particle fitness function and the classification accuracy index, svm-accuracy. The svm-accuracy is defined as the ratio of correctly identified samples to all samples in the dataset. It is a commonly used performance metric in classification tasks, measuring the algorithm's accuracy in classifying data. The effectiveness of the PSO-SVM method in feature selection and parameter optimization can be assessed by comparing its classification accuracy with that of SVM and GA-SVM. The algorithm performs better as the categorization accuracy increases. Below is a description of the classification accuracy index, svm-accuracy, used in the particle fitness function. This article initially specifies three hit rates for the second-type dataset, where there are only two types of samples (positive instances and negative examples): positive hit rate, reverse hit rate, and overall hit rate.

$$\begin{cases} r_p = num_p/Num_p \\ r_n = num_n/Num_n \\ r_n = (num_n + num_p)/(Num_n + Num_p) \end{cases} \tag{11}$$

where r_p is the positive hit rate; r_n is the counter hit rate; r_n is the overall hit rate; num_p represents the number of positive examples correctly classified by the classifier; num_n represents the number of negative examples correctly classified by the classifier; Num_p represents positive examples in the data set The total number of samples; Num_n represents the total number of negative examples in the data set. For a data set with two categories, $r_p \times r_n$ instead of r_a is used to measure the classification accuracy. Because a good classifier not only has a high overall hit rate, but also performs uniformly, that is,

TABLE 3. Number of particles used in each data set

Names	Total Features	Particle No. Cases
The diabetes	8	100
Heart disease	13	100
Ion ophere	34	200
Sonar	60	300
Iris	4	50
Vehicle	18	150

TABLE 4. Performance comparison of three methods of SVM, PSO-SVM and GA-SVM on the experimental data set

Datasets	SVM- Average			**PSO-SVM: Average**			GA-SVM: Average		
	Positive hit rate	Negative hit rate	Overall hit rate	positive hit rate	Average negative hit rate	Average overall hit rat e	Average positive hit rate	Average negative hit rate	Average overall hit rate
Diabetes	0.7318	0.7440	0.7396	0.7433	0.7920	0.7748	0.7835	0.8704	0.8150
Heart disease	0.8000	0.8067	0.8037	0.9417	0.9267	0.9333	0.9447	0.9511	0.9480
Ionosphere	0.9532	0.9377	0.9431	0.9923	1.000	0.9971	0.9963	0.9876	0.9856
Sonar	0.9477	0.6722	0.8187	1.000	1.000	1.000	0.9863	0.9842	0.9800
Iris	NA	NA	0.9667	NA	NA	0.9800	NA	NA	1.000
Vehicle	NA	NA	0.7364	NA	NA	0.9256	NA	NA	0.8406

the classification accuracy of different classes cannot be very different. This is especially important for the second type of data set. Therefore, it is more scientific to use $r_p \times r_n$ for svm- accuracy. For multi-class datasets, the overall hit ratio r_a is used to represent its svm — accuracy.

4.2. **Experimental Parameter Settings.** The experimental parameters for the algorithm are as follows: $co_1 = co_2 = 2$: These are the acceleration coefficients used in the PSO algorithm [20]. They control the balance between the particle's personal best position and the global best position when updating the velocity and position.

- Initial value range of particle velocity: [-5, 5]; this range determines the initial velocity of each particle in the PSO algorithm. The velocity affects how particles explore the search space and find the optimal solution.

- Number of particles: The number of particles varies depending on the dimensionality of the dataset. The specific values are shown in Table 2. A larger number of particles is used when the dimensionality of the dataset is high to ensure thorough exploration of the search space, while a smaller number of particles can be used for lower dimensional datasets.

- Stopping criterion: The PSO algorithm stops when either the number of iterations reaches 300 or the global optimal fitness value remains unchanged for 100 consecutive iterations. This ensures that the algorithm terminates when it has reached a satisfactory solution or when it has converged.

Other parameters used in the experiment are not specified in the given information. $w_a = 0.8, w_f = 0.2, n_R = n_S = 20. C \in [2^{(-5)}, 500], \gamma \in [2^{(-15)}, 1]$.

4.3. **Experimental Results.** The experimental outcomes and classification accuracy for each data set for the SVM, GA-SVM, and PSO-SVM algorithms are displayed in Table 4. The average value is the mean of the outcomes of executing the algorithm ten times on a given batch of data. Table 5 is a comparison of the GA-SVM method and the PSO-SVM

Table 5. Comparison of the average number of feature selections of the PSO-SVM and GA-SVM methods on the experimental data

Datasets	PSO-SVM Average number of selected features	GA-SVM Average number of selected features
Diabetes (8)	1.4	3.7
Heart disease (13)	4.1	5.4
Ionosphere (34)	3.9	6.0
Sonar (60)	5.8	15.0
Iris (4)	1.0	1.0
Vehicle (18)	9.7	9.2

method in feature selection. It shows that the two algorithms are run 10 times on the same data set. Table 4 Comparison of the average number of feature selections of the PSO-SVM and GA-SVM methods on the experimental data

Table 5 with the experimental results and classification accuracy of the SVM algorithm, GA-SVM algorithm, and PSO-SVM algorithm on each dataset. Based on the experimental setup and evaluation method described earlier, the average classification accuracy values can be calculated by averaging the results obtained from running each algorithm 10 times on each dataset. These average values would provide insights into the performance of the algorithms and allow for comparison among them.

4.4. **Discussion Results.** Based on the experimental results shown in Table 3, the following three methods (SVM, GA-SVM, and PSO-SVM) will be compared and analyzed in terms of classification accuracy, feature selection ability, and algorithm operation efficiency.

- Classification Accuracy: The PSO-SVM method demonstrates significantly improved classification accuracy compared to the SVM method. On each experimental dataset, the PSO-SVM method outperforms the SVM method in terms of classification accuracy. For example, on the Heart dataset, the positive hit rate, negative hit rate, and overall hit rate of the SVM method are 0.8000, 0.867, and 0.880, respectively. In contrast, the PSO-SVM method achieves higher hit rates of 0.9417, 0.9267, and 0.9333, respectively, which are 17.71%, 14.88%, and 16.13% higher than the SVM method. Similarly, on the Vehicle dataset, the average hit rate of the SVM method is 0.7634, while the PSO-SVM method achieves a hit rate of 0.9256, representing a 21.25% improvement over the SVM method. Moreover, the PSO-SVM method produces a more balanced classifier, with similar hit rates for different classes, as evident in the Sonar dataset.
- Feature Selection Ability: On average, the PSO-SVM method selects fewer features compared to the GA-SVM method on most datasets. For example, on the Ionosphere dataset with 34 features and the Sonar dataset with 60 features, the average number of features selected by PSO-SVM is only 3.9 and 5.8, respectively. In contrast, the average number of features selected by GA-SVM is 6.0 and 15.0, which are 1.54 times and 2.57 times higher than PSO-SVM, respectively. Despite the similar classification accuracy of both methods, this indicates that the PSO-SVM method has a better feature selection effect compared to GA-SVM.
- Algorithm Operation Efficiency: In terms of efficiency, the SVM method is the fastest during the training phase since it does not require an optimization algorithm like PSO or GA for feature selection and parameter optimization. However, the lack of

Table 6. A comparison of the PSO-SVM with the SVM and GA-SVM schemes in terms of classification accuracy, feature selection ability, and algorithm operation efficiency.

Method	**Classification Accuracy**	**Feature Selection Ability**	**Algorithm Operation Efficiency**
SVM	Good, but can be improved with feature selection. Prone to overfitting with many irrelevant features.	No inherent feature selection. All features used.	Fast computation once training is complete. Training can be slow for large datasets.
GA-SVM	Higher than SVM due to feature selection. Reduces overfitting.	Good at selecting relevant features and removing irrelevant ones. Improves classification.	Slower than SVM due to additional genetic algorithm steps for feature selection.
PSO-SVM	Higher than SVM due to feature selection. Reduces overfitting.	Good at selecting relevant features and removing irrelevant ones. Improves classification.	Faster than GA-SVM but slower than SVM due to additional particle swarm optimization steps for feature selection.

feature selection in the training phase can lead to reduced classification accuracy in the use phase, while an excessive number of features can slow down the SVM's runtime. In the long run, the efficiency of the SVM method is poor. Comparatively, the PSO-SVM method achieves similar classification accuracy with fewer iterations in the training phase (300 iterations compared to 600 iterations for GA-SVM) and adapts the number of particles based on the dataset's feature count (ranging from 50 to 300). This demonstrates that the PSO-SVM algorithm has better operation efficiency compared to GA-SVM.

Table 6 compares the PSO-SVM with the SVM and GA-SVM schemes regarding classification accuracy, feature selection ability, and algorithm operation efficiency. From a comprehensive analysis of classification accuracy, feature selection capability, and algorithm operation efficiency, the PSO-SVM algorithm proves to be superior to the SVM and GA-SVM algorithms. It achieves higher classification accuracy, better feature selection, and improved operation efficiency, making it a more effective method for classification tasks. GA-SVM and PSO-SVM have higher accuracy than SVM due to feature selection which reduces overfitting. All three methods have feature selection ability, but GA-SVM and PSO-SVM are designed to explicitly select features while SVM does not. The trade-off is improved accuracy vs increased computational cost for the hybrid GA/PSO based approaches compared to native SVM. SVM is the most efficient, followed by PSO-SVM, with GA-SVM being the slowest due to its evolutionary algorithm operations for feature selection. All three methods have feature selection ability, but GA-SVM and PSO-SVM are designed to explicitly select features while SVM does not. The trade-off is improved accuracy vs increased computational cost for the hybrid GA/PSO based approaches compared to native SVM.

5. **Conclusions.** This study presented a PSO-SVM technique that combines PSO synchronization for feature selection and SVM parameter optimization, aiming to improve

feature selection performance. The experimental findings have validated the efficiency and effectiveness of the PSO-SVM algorithm in enhancing classification accuracy. The PSO-SVM algorithm demonstrated its ability to identify optimal solutions by efficiently searching the search space and selecting relevant feature subsets. The feature reduction capability is crucial in real-world applications as it reduces dataset dimensionality, improves computational performance, and prevents overfitting. Furthermore, the PSO-SVM algorithm outperformed the GA-SVM algorithm regarding operating efficiency. The PSO-SVM method balances accuracy and efficiency by dynamically adjusting the number of particles based on the dataset's characteristics. Overall, the PSO-SVM approach presented in this study offers a viable solution for feature selection and SVM parameter optimization in classification tasks. It overcomes the limitations of the GA-SVM algorithm by providing a more comprehensive feature reduction range and increased operational efficiency. These results contribute to the advancement of machine learning and provide practitioners and researchers with a valuable tool to enhance classification accuracy across various domains. Future studies can further explore the potential of the PSO-SVM algorithm in broader machine-learning domains and evaluate its efficacy on more extensive [21] and diverse datasets [22]. The findings of this study lay the foundation for further research and development in the field, opening up new possibilities for improving classification outcomes.

REFERENCES

[1] S. C. Chu, T. K. Dao, J. S. Pan, and T. T. Nguyen, "Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification," *Eurasip Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 52, pp. 1-16, Dec. 2020, doi: 10.1186/s13638-020-01671-y.

[2] T. Dao, T. Nguyen, J. Pan, Y. Qiao, and Q. Lai, "Identification Failure Data for Cluster Heads Aggregation in WSN Based on Improving Classification of SVM," *IEEE Access,* vol. 8, pp. 61070-61084, 2020, doi: 10.1109/ACCESS.2020.2983219.

[3] T. K. Dao, T. S. Pan, T. T. Nguyen, and S. C. Chu, "A compact Articial bee colony optimization for topology control scheme in wireless sensor networks," *Journal of Information Hiding and Multimedia Signal Processing,* vol. 6, no. 2, pp. 297-310, 2015.

[4] D.-L. Zhang, J. Qiao, J.-B. Li, L.-Y. Qiao, S.-C. Chu, and J. F. Roddick, "Optimizing Matrix Mapping with Data Dependent Kernel for Image Classification," *Journal of Information Hiding and Multimedia Signal Processing,* vol. 5, no. 1, pp. 72-79, 2014.

[5] T. Dao, J. Yu, T. Nguyen, and T. Ngo, "A Hybrid Improved MVO and FNN for Identifying Collected Data Failure in Cluster Heads in WSN," *IEEE Access,* vol. 8, pp. 124311-124322, 2020, doi: 10.1109/ACCESS.2020.3005247.

[6] S. C. Chu, T. K. Dao, J. S. Pan, and T. T. Nguyen, "Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification," *Eurasip Journal on Wireless Communications and Networking,* vol. 2020, no. 1, Dec. 2020, doi: 10.1186/s13638-020-01671-y.

[7] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers* & *Electrical Engineering,* vol. 40, no. 1, pp. 16-28, 2014.

[8] M. A. Khanesar, M. Teshnehlab, and M. A. Shoorehdeli, "A novel binary particle swarm optimization," *in 2007 Mediterranean conference on control* & *automation*, Athens, Greece, 2007, pp. 1-6. doi: 10.1109/MED.2007.4433821

[9] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," *in 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence* , (Cat. No.98TH8360), 1998, pp. 69-73, doi: 10.1109/ICEC.1998.699146.

[10] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft computing,* vol. 22, pp. 387-408, 2018.

[11] M. Zhao, C. Fu, L. Ji, K. Tang, and M. Zhou, "Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes," *Expert Systems with Applications,* vol. 38, no. 5, pp. 5197-5204, 2011.

[12] T.-T. Nguyen, J. Yu, T.-T.-T. Nguyen, Q.-A. Lai, T.-G. Ngo, and T.-K. Dao, "An Optimizing Parameters and Feature Selection in SVM Based on Improved Cockroach Swarm Optimization " BT - *Advances in Intelligent Information Hiding and Multimedia Signal Processing,* 2021, pp. 349-357.

[13] G. Chen and J. Chen, "A novel wrapper method for feature selection and its applications," *Neurocomputing*, vol. 159, pp. 219-226, 2015, doi: https://doi.org/10.1016/j.neucom.2015.01.070.

[14] M. Tomassini, "A Survey of Genetic Algorithms," *Annual Reviews of Computational Physics, World Scientific*, vol. 3, pp. 87-118, 1995.

[15] S. Suthaharan and S. Suthaharan, "Support vector machine," *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pp. 207-235, 2016.

[16] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," *in New directions in statistical physics: econophysics, bioinformatics, and pattern recognition,* Springer, 2004, pp. 273-309.

[17] C.-L. Huang and J.-F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter optimization," *Applied Soft Computing,* vol. 8, no. 4, pp. 1381-1391, Sep. 2008, doi: 10.1016/J.ASOC.2007.10.007.

[18] H. Y. Jiang and L. B. Zou, "A Hybrid PSO-SA Optimizing Approach for Svm Models in Classification," *International Journal of Biomathematics,* vol. 6, no. 5, 1350036, 2013, doi: Artn 1350036Doi 10.1142/S1793524513500368.

[19] S. Z. Fangjun Kuang, Weihong Xu, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Applied Soft Computing,* vol. 18, pp. 178-184, 2014://doi.org/10.1016/j.asoc.2014.01.028

[20] T. Bai, Y. Kan, J. Chang, Q. Huang, and F.-J. Chang, "Fusing feasible search space into PSO for multi-objective cascade reservoir optimization," *Applied Soft Computing,* vol. 51, pp. 328-340, 2017.

[21] D.-T. Pham, D. -T. -T. Hoang, T. T. Nguyen, V.-T. Nguyen, T. D. Nguyen, "An Improved Whale Optimization Algorithm for Optimal Multi-threshold Image Segmentation," *Journal of Information Hiding and Multimedia Signal Processing,* vol. 14, no. 2, pp. 1-11, June 2023.

[22] P. N. Huu and T. N. T. Mai, "Proposing a fruit classification system using Tensorflow model," *Journal of Information Hiding and Multimedia Signal Processing,* vol. 14, no. 1, pp. 10-19, March 2023.