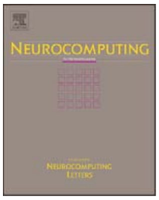
Neurocomputing 73 (2009) 438–448

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



ACO-based hybrid classification system with feature subset selection and model parameters optimization

Cheng-Lung Huang *

Department of Information Management, National Kaohsiung First University of Science and Technology, 2, Juoyue Rd., Nantz District, Kaohsiung 811, Taiwan, ROC

ARTICLE INFO

Article history:
Received 19 February 2008
Received in revised form
21 April 2009
Accepted 27 July 2009
Communicated by T. Heskes
Available online 19 August 2009

Keywords:
Ant colony optimization
Feature selection
Support vector machine

ABSTRACT

This work presents a novel hybrid ACO-based classifier model that combines ant colony optimization (ACO) and support vector machines (SVM) to improve classification accuracy with a small and appropriate feature subset. To simultaneously optimize the feature subset and the SVM kernel parameters, the feature importance and the pheromones are used to determine the transition probability; the classification accuracy and the weight vector of the feature provided by the SVM classifier are both considered to update the pheromone. The experimental results indicate that the hybridized approach can correctly select the discriminating input features and also achieve high classification accuracy.

1. Introduction

Many practical pattern classification tasks require the learning of a classification function that assigns a given input pattern, usually represented by a vector of attribute values, to a finite set of classes. Feature selection is employed to identify a powerfully predictive subset of fields in a database and reduce the number of fields presented to the data mining. Significant computation time can thus be saved and the constructed models can generalize well for unseen data by extracting as much information as possible from a given data set while using the smallest number of features [1]. The choice of features used to represent patterns that are presented to a classifier influences several aspects of pattern classification, including the accuracy of the learned classification algorithm, the time required to learn a classification function, the number of examples required for learning, and the cost associated with the features [2]. The feature subset selection algorithms have been classified into two categories based on whether feature selection is performed independently of the learning algorithm that constructs the classifier—the filter approach and the wrapper approach [3–5]. The filter approach initially selects important features and then the classifier is used for classification. The wrapper approach either modifies the classifier to select important features or combines the classifier with other optimization tools to select features.

A range of machine learning approaches have been developed to build classifiers, including artificial neural networks, k-nearest neighbor algorithms and support vector machines (SVM). SVM [6] has recently been adopted to solve a range of problems. Feature subset selection is an important issue in building an SVM-based classification model (as well as other classification models, such as the neural networks). In addition to the feature selection, a kernel function, kernel parameters and a soft margin constant C (also called the regularization parameter) [6] must be determined to construct an accurate SVM classifier. To identify the best subset of features, a wrapper-based system typically combines a classifier with stochastic optimization techniques, including simulated annealing, genetic algorithms and ant colony optimization.

Ant colony optimization (ACO) [7–9] is an artificial system inspired by the behavior of real ant colonies and is used to solve discrete combinatorial optimization problems [10,11]. Ants deposit pheromones along their trail to a food source. At a decision point, they make a probabilistic choice based on the amount of pheromone along each search branch. ACO has been used as a search procedure for selecting features; and it has been combined with the artificial neural network classifier [12–14], the nearest neighbor classifier [15], rough set theory [16–18] or SVM [19–22]. As well as feature selection, the proper setting of parameters for the classifier can also increase classification accuracy. Both feature subset selection and model parameter setting substantially influence classification accuracy [23,24]. The optimal feature subset and model parameters must be determined simultaneously, since feature subset selection affects the appropriate model (kernel) parameters and vice versa [24]. Since research on the simultaneous optimization of the feature subset

* Tel.: +886 7 6011000x4127; fax: +886 7 6011042.
E-mail address: clhuang@ccms.nkfust.edu.tw

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.07.014

and model parameters using ACO is lacking, this investigation proposes an intelligent system that incorporates two newly developed techniques, ACO and SVM, to solve this problem.

In designing an ACO-based system, the construction of a traversal path for an ant, the pheromone update strategy, and the design of the transition probability are critical. In the proposed scheme, the pheromone and heuristic information are used to determine the transition probability; a solution is generated for an ant that uses the transition probability to determine the classifier's model parameters and the input features; the SVM classifier evaluates the quality of the solution in terms of the classification accuracy and the weight vector of the SVM model; the solution quality is then used to update the pheromone tables. The proposed ACO–SVM is a wrapper-based hybrid system; thus the overall classification accuracy and the feature importance provided by the classifier can be integrated together into the ACO algorithm. But a filter-based approach, which selects feature subset first and then trains the classifier, cannot tightly combine the overall classification accuracy and feature importance provided by the classifier. While the filter approach is generally computationally more efficient than the wrapper approach, however, its major drawback is that an optimal selection of features may not be independent of the inductive and representational biases of the learning algorithm that is used to construct the classifier [25]. The proposed hybrid system can avoid this drawback.

This paper is organized as follows. Section 2 addresses work related to the basic SVM and ACO concepts. Section 3 describes the proposed ACO–SVM hybrid system. Section 4 presents the experimental results obtained using a simulated dataset and three public datasets. Section 5 draws conclusions.

2. Brief introduction to the ant colony algorithm and support vector machines

2.1. Support vector machines

Given a training set of instance-label pairs $(\mathbf{x}_i, y_i)$, $i = 1, 2, \ldots, m$ where $\mathbf{x} \in R^n$ and $y \in \{+1, -1\}$, the SVM finds an optimal separating hyperplane, $d(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$, by solving the following optimization problem:

$$\underset{\mathbf{w},b,\xi}{\text{Minimize}}\ \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\xi_i$$

$$\text{Subject to}: y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) + \xi_i - 1 \geq 0; \xi_i \geq 0 \tag{1}$$

where C is a regularization parameter associated with the training error; ξ_i is the non-negative slack variable, and ϕ is a mapping function that maps the training samples from the input space into a higher-dimensional feature space.

The optimal hyperplane provides the minimum number of training errors (to keep the constraint violation as small as possible), and can be solved by introducing Lagrange multipliers for the dual optimization model [26–28]. After the optimal hyperplane has been solved, the decision function (classifier) is given by,

$$\text{sign}(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \tag{2}$$

or

$$\text{sign}\Big(\sum_{i=1}^{m}\alpha_i y_i[\phi^T(\mathbf{x}_i)\phi(\mathbf{x})] + b\Big) \tag{3}$$

In Eq. (3), the required scalar product $\phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$ is calculated directly by computing the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$ for given training data in an input space. The radial basis function (RBF) is a common kernel function, as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2) \tag{4}$$

The multi-class classifier is based on two major multi-class SVM classification strategies—"one-against-all" and "one-against-one." (1) In the one-against-all strategy [29], ν binary SVM decision functions are constructed for an ν-class problem. The jth ($j = 1, 2, \ldots, \nu$) decision function is trained by labeling all of the examples in the jth class with positive labels, and all of the examples that are not in the jth class with negative labels. A new $\mathbf{x}$ is classified into the class that has the largest decision function. (2) In the one-against-one strategy [30,31], $C_2^{\nu} = \nu(\nu - 1)/2$ classifiers are constructed, and each classifier is trained using two classes (such as class c_i vs. class c_j). A new $\mathbf{x}$ is classified into the majority class that is voted on by all of the decision functions.

2.2. Ant colony optimization

The ant-based algorithm, used to solve mainly combinatorial optimization problems, was inspired by observations of the foraging behavior of real ants. The first ACO was developed to solve the classical traveling salesman problem (TSP) [8,9]. This section introduces the basic concepts based on which the standard ACO algorithm is used to solve the TSP problem. More details can be found elsewhere [32].

(1) *Transition probability*: In the TSP problem, the transition probability from city i to city j for the kth ant at time step t is expressed as,

$$PROB_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{j\in I_i^k}[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}} & \text{if } j \in I_i^k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\tau_{ij}(t)$ is the amount of pheromone trail on edge (i,j) at time t; η_{ij} is the *a priori* available heuristic information; α and β are two factors that specify the relative effects of pheromone trail and heuristic information; and I_i^k is the set of feasible neighborhood cities that have not yet been visited by ant k.

(2) *Pheromone update*: The solution is generated after each ant has completed a tour. Then, the pheromone trails are updated by initially lowering them with a constant evaporation rate and then allowing each ant to deposit pheromone on the arcs that are part of its tour, as indicated in the following equation:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{NumberOfAnts}\Delta\tau_{ij}^{k} \tag{6}$$

where ρ is the pheromone trail evaporation rate ($0<\rho<1$). The parameter ρ is used to prevent unlimited accumulation of the pheromone trails and enables the algorithm to "forget" previously made bad decisions. On arcs that are not selected by the ants, the associated pheromone strength declines exponentially with the number of iterations. $\Delta\tau_{ij}^{k}$ is the quantity per unit of length of the trail substance that is laid on edge (i, j) by the kth ant.

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if ant } k \text{ uses edge}(i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where L_k denotes the tour length, and Q is a predefined constant.

3. Proposed methodology

The ACO is designed to optimize the feature subset and the classifier parameters; an ant's solution represents a combination of the feature subset and the classifier parameters, C and γ, based on the radial basis function (RBF) kernel of the SVM classifier. The classification accuracy and the feature weights of the constructed SVM classifier are used to design the pheromone update strategy. Based on the pheromone table and measured relative feature importance, the transition probability is calculated to select a solution path for an ant. Accordingly, this scheme tightly combines the ACO with the SVM via a wrapper-based feature selection procedure. In implementing the proposed scheme, this work uses the RBF kernel function (defined by Eq. (4)) for the SVM classifier because the RBF kernel can analyze high-dimensional data and requires that only two parameters (C and γ) be defined [33,34]. However, other kernel parameters may also be optimized in a similar way.

Since ACO is originally designed for discrete combinatorial optimization, the proposed ant optimization algorithm also searches in a discrete search space. Therefore, the continuously valued C and γ are discretized into discrete-valued sets Ω_C and Ω_γ, respectively. In feature selection, the feature set is represented as Ω_F, whose size is determined by the number of features in a dataset. The SVM parameters, Ω_C and Ω_γ, and the feature Ω_F are thus combined to form a search space with a maximum of $(\Omega_C) \times (\Omega_\gamma) \times (2^{(\Omega_F)} - 1)$ possible combinations, where # represents the size of a set (cardinal number). An ant selects (visits) one element (city) from the search space Ω_C, one element from the search space Ω_γ, and several elements from the search space Ω_F. Thus, an ant's solution comprises three parts, C, γ and the selected features.

Fig. 1 displays an example of a traversal path for a particular ant. An ant randomly selects a start C at time step $t = 0$; travels to γ at time step $t = 1$; travels to the first feature at time step $t = 2$, and travels to other features at time steps from $t = 3$ to $t = 2$+the predefined size of the feature subset. Notably, the ants all have different numbers of selected features, so they may not need to visit every feature in the search space Ω_F as they travel from feature to feature. An ant stops visiting the feature space when it reaches a predefined selected feature size. The visited features in the feature mask are labeled "1", while the cities that are never visited are labeled "0". The input features of the SVM model are filtered through the feature mask that is represented as a series of binary values—"1" represents the selection of the feature, while "0" indicates that the feature is not selected. For example, given a feature size of five, a traversal path of $[\underline{19.5}, \underline{0.24}, \underline{1, 0, 1, 0, 1}]$ represents $C = 19.5$, $\gamma = 0.24$, and feature #1, #3, and #5 are selected (#2 and #4 are not selected).

The main steps in Fig. 2 are implemented to establish the proposed ACO-based feature selection and the optimization of the parameters (ACO–SVM). The main steps are as follows: (1) initializing the pheromone tables and system parameters, (2) calculating the feature importance, (3) constructing a solution for C, γ and the selected features, (4) establishing the SVM classifier model for each ant, and (5) updating the pheromone trails. These steps are described in detail in the following subsections.

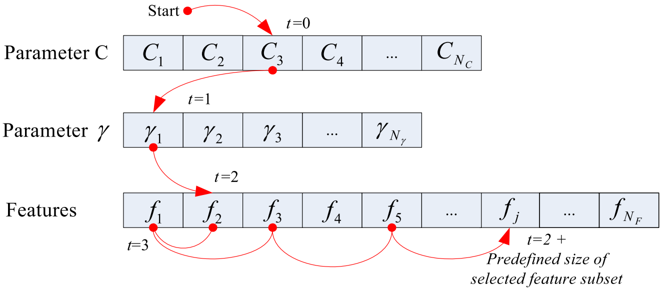


Fig. 1. Example of traversal path for a certain ant.

3.1. Initialization

For an ant that is establishing a solution path from parameter C to parameter γ, from parameter γ to the first element in the feature set, and from feature to feature, three pheromone tables are required to design the transition probabilities.

(1) The pheromone table for designing the probability of transition along the path from parameter C to parameter γ is $PHERO^{C\to\gamma}$.
(2) The pheromone table for designing the probability of transition along the path from parameter γ to the first element of the feature set is $PHERO^{\gamma\to F}$.
(3) The pheromone table for designing the probability of transition along the path from feature to feature is $PHERO^{F\to F}$.

As stated in the preceding section, the resolution for searching parameters C and γ should be determined by dividing the C and γ into search points—Ω_C and Ω_γ, respectively. Given that $(\Omega_C) = N_C$, $(\Omega_\gamma) = N_\gamma$, and $(\Omega_F) = N_F$, the dimensions of $PHERO^{C\to\gamma}$, $PHERO^{\gamma\to F}$ and $PHERO^{F\to F}$ are $N_C \times N_\gamma$, $N_\gamma \times N_F$ and $N_F \times N_F$, respectively. These three pheromone tables are initialized to be a positive constant c.

$$\tau_{ij}(0) = c \tag{8}$$

Like the pheromone tables, some important system parameters must be initialized as follows:

(1) The number of ants, *NumberOfAnts*, the pheromone evaporation rate ρ, and the elite coefficient e on the pheromone update (defined by Eq. (16) in Section 3.5) are initialized.
(2) The number of selected features for each ant is randomly generated according to a uniform distribution $Uniform(1, N_F)$ from 1 to N_F before a prespecified iteration, *ThresholdIterationOfNormalDistribution*, after which a normal distribution $Normal(\mu, \sigma)$ is applied to a specification proportion of ants (half of the ants herein), while the uniform distribution is still applied to the other proportion of ants. In $Normal(\mu, \sigma)$, the number of best ants in the specification of the normal distribution is *NumberOfEliteAntsDefiningNormal*; and the mean μ and deviation σ are determined by calculating the mean and deviation of the size of the selected feature subset from these best ants. Accordingly, the size of the selected feature subset tends to converge to a stable size, and can search globally and randomly the search space. Hence, the convergence and randomness of the feature size are balanced in the final stage of the search.

3.2. Calculation of feature importance

F-score [35] is a simple measurement which is used to evaluate the discrimination ability of a feature. Eq. (9) defines the F-score of the ith feature. The numerator specifies the discrimination among the categories of the target variable, and the denominator indicates the discrimination within each category. A larger F-score corresponds to a greater likelihood that this feature is discriminative [35].

$$FSCR_i = \sum_{c=1}^{v} (\bar{x}_i^{(c)} - \bar{x}_i)^2 \Bigg/ \sum_{c=1}^{v} \left\{ \frac{1}{N_i^{(c)} - 1} \sum_{j=1}^{N_i^{(c)}} (x_{i,j}^{(c)} - \bar{x}_i^{(c)})^2 \right\},$$
$$i \in \{1, 2, \ldots, N_F\} \tag{9}$$

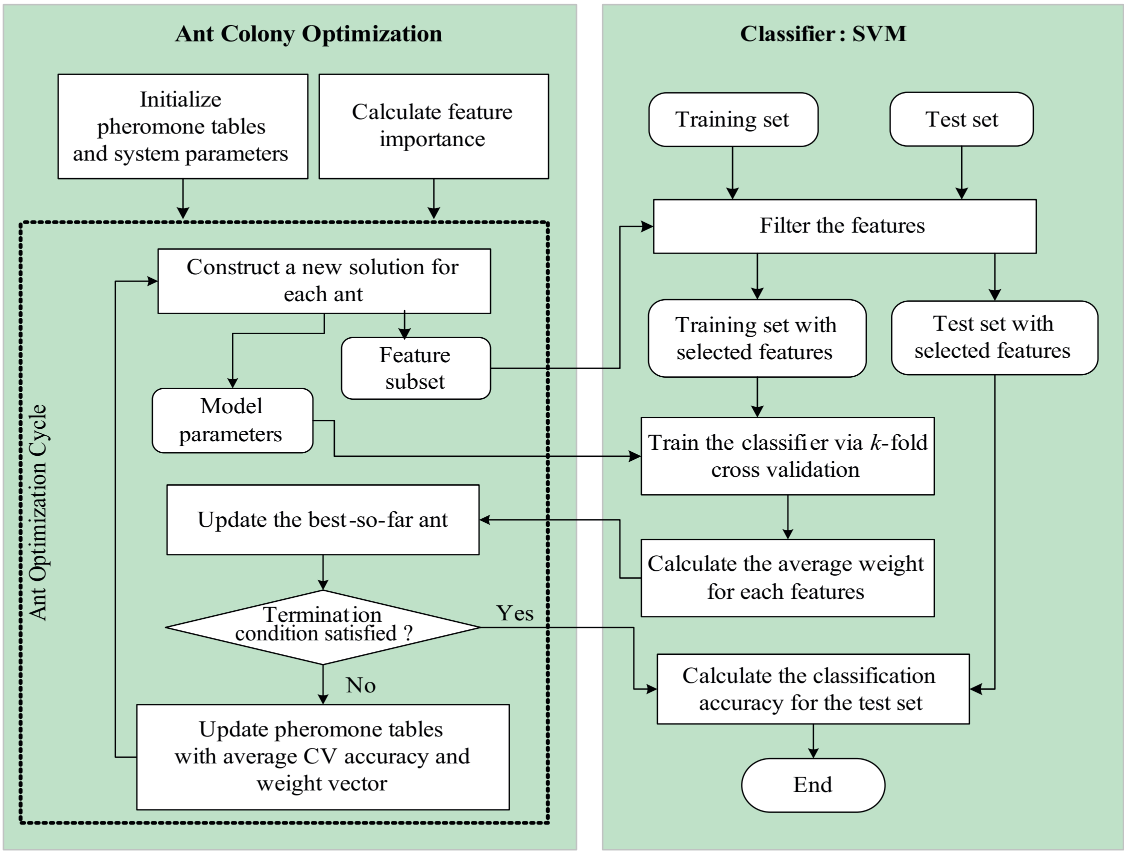

Ant Colony Optimization
Initialize pheromone tables and system parameters
Calculate feature importance
Ant Optimization Cycle
Construct a new solution for each ant
Feature subset
Model parameters
Update the best-so-far ant
Termination condition satisfied ?
Yes
No
Update pheromone tables with average CV accuracy and weight vector
Classifier: SVM
Training set
Test set
Filter the features
Training set with selected features
Test set with selected features
Train the classifier via k-fold cross validation
Calculate the average weight for each features
Calculate the classification accuracy for the test set
End

Fig. 2. System architectures of proposed ACO-based feature selection and optimization of parameters.

where v is the number of categories of target variable; N_F the number of features; $N_i^{(c)}$ the number of samples of the ith feature with categorical value c, $c \in \{1,2,\ldots,v\}$, $i \in \{1,2,\ldots,N_F\}$; $x_{i,j}^{(c)}$ the jth training sample for the ith feature with categorical value c, $j \in \{1,2,\ldots,N_i^{(c)}\}$; $\overline{x}_i$ the mean of the ith feature; $\overline{x}_i^{(c)}$ the mean of the ith feature with categorical value c.

3.3. Construction of a solution: C, γ and feature subset

This step constructs a solution of C, γ and selected feature subset for each ant. An ant travels from C, through γ to selected features. The mechanism by which an ant selects the next city is defined as,

```
if a_generated_random_number ≤ $\varphi$, then
   choose the next city with random,
else
   choose the next city with ij.
```

where *a_generated_random_number* is a random number between zero and one, φ is a predefined threshold value between zero and one, and $PROB_{ij}$ is the state transition probability from city i to city j.

Three transition probabilities with various pheromone tables and heuristic information are designed to obtain a solution.

(1) *Transition from C to γ*. The transition probability from city i (C) to city j (γ) for the kth ant is as follows:

$$PROB_{ij}^{k}\ (t=1) = \frac{PHERO_{ij}^{C\to\gamma}}{\sum_{i=1}^{N_C}\ \sum_{j=1}^{N_\gamma}\ PHERO_{ij}^{C\to\gamma}} \tag{10}$$

(2) *Transition from γ to the first element in the feature set*. The transition probability from city i (γ) to city j (feature) for the kth ant to choose an element in the feature set is as follows:

$$PROB_{ij}^{k}\ (t=2) = \frac{[PHERO_{ij}^{\gamma\to F}\ ]^{\alpha}[FSCR_j]^{\beta}}{\sum_{j=1}^{N_F}\ [PHERO_{ij}^{\gamma\to F}\ ]^{\alpha}[FSCR_j]^{\beta}} \tag{11}$$

In the above equation, the heuristic information η is j, and ants are more likely to select features with a higher F-score.

(3) *Transition from feature to feature in the feature set*. An ant stops visiting the feature in feature set Ω_F when the selected feature subset reaches the predefined size. Given the feasible features visited by the kth ant from feature i at time step t, I_i^k, the transition probability from city i (feature) to city j (feature) is as follows:

$$PROB_{ij}^{k}\ (t) = \begin{cases} \dfrac{[PHERO_{ij}^{F\to F}\ ]^{\alpha}[FSCR_j]^{\beta}}{\sum_{j\in I_i^k}[PHERO_{ij}^{F\to F}\ ]^{\alpha}[FSCR_j]^{\beta}} & \text{if } j \in I_i^k \\ 0 & \text{otherwise} \end{cases}$$

$$t = 3,4,\ldots, \textit{predefined size of feature subset} + 2 \tag{12}$$

The heuristic information η in Eqs. (11) and (12), is $FSCR_j$. However, if the dataset contains cost information for feature j, $Cost_j$, then the cost factor can be incorporated into heuristic information in Eq. (13). If the dataset does not contain feature cost information, then $Cost_j$ may be just set to a constant, such as "1".

$$\eta_{ij} = FSCR_j/Cost_j \tag{13}$$

3.4. Construction of classifier model

Based on the ant's solution of C, γ and the selected feature subset, the solution quality in terms of classification accuracy is obtained by developing SVM models using K-fold cross validation (CV) with the training set. That is, the training set is divided into K subsets, each of which acts as an independent holdout test set for the model that is trained with the remaining $K-1$ subsets. The test accuracy (*Test_Accuracy*) that measures the percentage of examples that are correctly classified is calculated for the K folds, and then the CV accuracy (*CVACC*) is determined by averaging the K test accuracies, as defined in the following equation:

$$CVACC = \frac{\sum_i Test\_Accuracy_i}{K} \quad i = 1, 2, \ldots, K \tag{14}$$

The advantages of cross validation are that all of the test sets are independent and the reliability of the results can be improved [36]. This study employs CV accuracy with $K = 5$ to update the pheromones in the following step. Along with the CV accuracy, the weight vector of the decision function for the SVM classifier defined in Eq. (2) is used to update the pheromones. The weight is obtained by building the SVM model with the whole training set based on the ant's solution of C, γ and the selected feature subset. A binary or multiple SVM models are constructed depending on whether the target variable is of the binary.

- For a binary-class problem, the weight is directly obtained from the constructed SVM model, and it is further scaled into the range zero to one.
- For a multi-class problem, $C_2^v = v(v-1)/2$ binary SVM models are constructed using the one-against-one approach for a v-class problem. Thus, the average weight of a feature can be calculated by averaging the weights of these binary SVM models, and it is further scaled into the range zero to one.

$$Weight_j = \frac{\sum_l Weight_j^l}{C_2^v} \quad l = 1, 2, \ldots, C_2^v \tag{15}$$

where $Weight_j^l$ represents the weight value of the jth feature in the lth SVM model.

3.5. Update of pheromone trails

After all of the ants' solution qualities have been calculated, the best-so-far ant is updated. If the predefined maximum iteration is reached, then the optimal model is evaluated using the test set and the ACO optimization cycle terminated; otherwise the pheromone update is made as follows. The pheromone trails are updated by evaporating all of the pheromone trails at a constant pheromone evaporation rate, and then depositing pheromone on the trail that the ant has crossed in its tour. An elitist strategy [32] in which additional pheromone, $\Delta\tau_{ij}^{Best}$, is deposited by the best-so-far ant is defined as follows:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{NumberOfAnts} \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{Best} \tag{16}$$

where e is the weight factor of the pheromone that is deposited by the ant *Best* (best-so-far).

The proposed scheme utilizes the following performance of an SVM classifier to update the pheromone tables.

- CV accuracy of the SVM classifier, which indicates the quality of an ant's solution.
- Weight vector of the SVM classifier, which indicates the relative importance of features used to construct the SVM model.

Since the CV accuracy specifies the goodness of an ant's solution, and the weight vector specifies the relative importance of the feature used to construct the SVM model, the two performances are properly used to update the pheromone trails. The advantage of this approach is that both the overall classification accuracy and the information on feature importance that is provided by the classifier can be considered simultaneously. This approach differs from previous feature selection methods [13,14,37].

The following three pheromone tables are updated in this step:

(1) Update pheromone table $PHERO^{C\to\gamma}$

$$\Delta\tau_{ij}^k = \begin{cases} CVACC^k & \text{if ant } k \text{ uses edge } (i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where $CVACC^k$ is the CV accuracy for the ant k.

(2) Update pheromone table $PHERO^{\gamma\to F}$

$$\Delta\tau_{ij}^k = \begin{cases} CVACC^k \times Weight_j^k & \text{if ant } k \text{ uses edge } (i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where $Weight_j^k$ is the weight value for the feature j in the ant k's SVM model.

(3) Update pheromone table $PHERO^{F\to F}$

$$\Delta\tau_{ij}^k = \begin{cases} CVACC^k \times Weight_i^k \times Weight_j^k & \text{if ant } k \text{ uses edge } (i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $Weight_i^k$ and $Weight_j^k$ are the weights of the features i and j in ant k's SVM model.

3.6. Computational complexity analysis

The runtime of the proposed ACO–SVM model that combines ACO and SVM can be computed according to the main steps of the wrapper-based model as follows:

$$N_{Iterations} \times (N_{Ants} \times (T_{PathGenerating} + T_{ClassifierTraining}) + T_{PheromoneUpdating}) \tag{20}$$

where $N_{Iterations}$ is the number of iterations; N_{Ants} the number of ants; $T_{PathGenerating}$, $T_{ClassifierTraining}$ and $T_{PheromoneUpdating}$ are the runtimes for an ant to generate a traversal path, train SVM classifier, and update pheromone tables, respectively.

(1) $T_{PathGenerating}$: In the worst case, each ant selects all the features. After each feature and C, γ are traversed, this will result in $(N_F + 2)$ steps per ant, where N_F is the number of features.

(2) $T_{ClassifierTraining}$: The runtime of SVM training is usually analyzed as the required runtime to obtain an accurate solution to the dual of the SVM optimization problem [6]. The runtime scales cubically with the size of the training set by using traditional optimization techniques. For example, a general interior point method on the dual of the SVM problem would take $O(m^3)$ arithmetic operations per iterations [38,39], where m is the

size of the training set. Therefore, many modern SVM solvers use "decomposition techniques" to avoid a cubic dependence on the size of training samples [39].

(3) $T_{PheromoneUpdating}$: The runtime of pheromone update is proportional to the dimensions of the three pheromone tables: $N_C \times N_\gamma$, $N_\gamma \times N_F$ and $N_F \times N_F$.

For the filter-based approach, feature subsets are properly selected using a filter approach and then the classifier is trained. In the feature selection stage, filter-based approach requires some evaluations or measures (such as F-scores and information measures) to discern good features from bad ones. In the classifier construction stage, SVM model parameters (C and γ) are properly optimized by ACO. The runtime of the filter-based ACO model (expressed as FACO–SVM) is as follows:

$$T_{FeatureFiltering} + N_{Iterations} \times (N_{Ants} \times (T_{PathGenerating} + T_{ClassifierTraining}) + T_{PheromoneUpdating}) \tag{21}$$

The runtime of feature filtering ($T_{FeatureFiltering}$) is proportional to the number of features, N_F. The runtime of path generation ($T_{PathGenerating}$) is a two-step traversal time for an ant to traverse from C to γ. This filter-based approach only updates one pheromone table ($PHERO^{C\to\gamma}$) and its runtime of pheromone update is proportional to the dimension of the pheromone table, $N_C \times N_\gamma$.

We can notice that the runtime of wrapper-based approach is slightly more complicated than that of the filter-based approach, and both runtimes are proportional to the number of iterations and the number of ants. Since the proposed wrapper-based ACO–SVM model does not dramatically increase the computational complexity, it is proper to simultaneously optimize the feature subset and SVM parameters in a single stage using the proposed wrapper-based approach.

4. Experiments and results

A simulated dataset and three UCI datasets were experimentally studied in the experiments. All of the input variables were scaled during the data preprocessing stage to prevent attributes with greater numerical ranges from dominating those in smaller numerical ranges and to reduce the difficulty of the calculation. Generally, each feature can be linearly scaled to the [0, 1] range using the following formula:

$$x' = (x - \min_i)/(\max_i - \min_i) \tag{22}$$

where x is the original value; x' is the scaled value, and $\max_i$ and $\min_i$ are the maximum and minimum values of feature i, respectively.

The system parameter settings were as follows. $NumberOfAnts = 20$, $\rho = 0.2$, $e = 1$, $\varphi = 0.2$, $c = 5000$, $\alpha = 1$ and $\beta = 1$. The search ranges for C and γ were $C \in [2^{-1}, 2^{12}]$, and $\gamma \in [2^{-12}, 2^{2}]$. In this experiment, C and γ were divided into many discrete search points with discretization interval lengths of 0.25, which determined the resolution of the search. The number of elite ants that defines the normal distribution of the size of the selected feature subset, *NumberOfEliteAntsDefiningNormal* was set to 10. The threshold iteration of normal distribution, *ThresholdIterationOfNormalDistribution*, was set to 120.

The proposed ACO–SVM was compared with the popular grid search algorithm (Grid–SVM) in searching for the SVM model parameters without feature selection [33–35,40,41]. In the grid search approach, pairs of (C,γ) were tried in the training set, and the best parameter (C,γ) that yields the best average cross validation accuracy (with $K = 5$, the same base as in the ACO–SVM) was then chosen as the predictor to measure the accuracy of the test set. The search ranges and search resolution were also set to the same basis in both ACO–SVM and Grid–SVM.

The implementation platform was implemented in Matlab 6.0, which is a general mathematical development tool. The Libsvm (version 2.82) originally designed by [42] was used as the SVM classifier. The empirical evaluation was performed using an Intel Pentium IV CPU, running at 2.8 GHz with 512 MB RAM.

4.1. Simulated dataset

A simulated data set modified from Punch et al. [43] was tested to demonstrate the classificatory accuracy and determine whether the proposed model can correctly select the relevant features. Eight target classes must be classified in this data set. The data set has 15 features of which only five (f2, f2, f6, f8, and f10) are relevant to the eight classes. Table 1 represents a template from which as many examples as required can be generated. The expression "0.0–1.0" represents uniformly distributed random values generated in the range of zero to one. The expression 0.2+[f4] represents a generation expression, where [f4] means that the present value of the f4 field is used in the calculation. Thus features f2, f4, f6, f8 and f10 are fields that can be utilized simultaneously to distinguish the eight classes A–H, while other 10 features are uniform random noise that is generated in the range [0.0, 1.0]. Each class comprises 200 cases; therefore, 1600 cases were generated in the data set.

For the ACO–SVM, the training CV accuracy and the test accuracy of the best-so-far ant at each iteration were computed and recorded. The test accuracy curve was plotted and training was terminated at the iteration in which the test accuracy was stable during the training process, as displayed in Fig. 3. Therefore, the predefined maximum iteration herein was set to approximately 320 to avoid overtraining and to save training time. The best ant with the best C, γ and feature subset was obtained at the predefined maximum iteration.

To obtain a more reliable result, 10 runs were conducted by 10-fold cross-validation with the simulated dataset. Table 2 presents the average and variance of the accuracy of the training and test set. The average training and test accuracy were 89.19% and 89.13% for the Grid–SVM, and 97.19% and 96.92% for the ACO–SVM. The non-parametric Wilcoxon signed rank test [44] with two related (dependent) samples was performed to compare the performance of the two approaches. With respect to test accuracy and training accuracy, ACO–SVM significantly outperforms the grid search approach with a significance level of 0.005 based on the Wilcoxon signed rank test.

For ACO–SVM, the size of the features and the ratio of the correct features (*CFRatio*) are illustrated. *CFRatio* is defined

Table 1
Template for generating a data set that has 15 features and eight classes.

Class	F2	F4	F6	F8	F10	Others
A	0.0–0.1	0.0–0.1	0.2+[f4]	$0.5 \times [f4]^2$	$0.5+5 \times [f4]^2$	0.0–1.0
B	0.0–0.1	0.0–0.1	0.2+[f4]	$0.5 \times [f4]^2$	$1.0+25 \times [f4]^2$	0.0–1.0
C	0.0–0.1	0.1–0.2	0.3−[f4]	$0.5 \times [f4]^2$	$0.5+5 \times [f4]^2$	0.0–1.0
D	0.0–0.1	0.1–0.2	0.3−[f4]	$0.5 \times [f4]^2$	$1.0-25 \times [f4]^2$	0.0–1.0
E	0.1–0.2	0.0–0.1	0.2+[f4]	$0.5 \times [f4]^2$	$0.5+5 \times [f4]^2$	0.0–1.0
F	0.1–0.2	0.0–0.1	0.2+[f4]	$0.5 \times [f4]^2$	$1.0+25 \times [f4]^2$	0.0–1.0
G	0.1–0.2	0.1–0.2	0.3−[f4]	$0.5 \times [f4]^2$	$0.5+5 \times [f4]^2$	0.0–1.0
H	0.1–0.2	0.1–0.2	0.3−[f4]	$0.5 \times [f4]^2$	$1.0-25 \times [f4]^2$	0.0–1.0

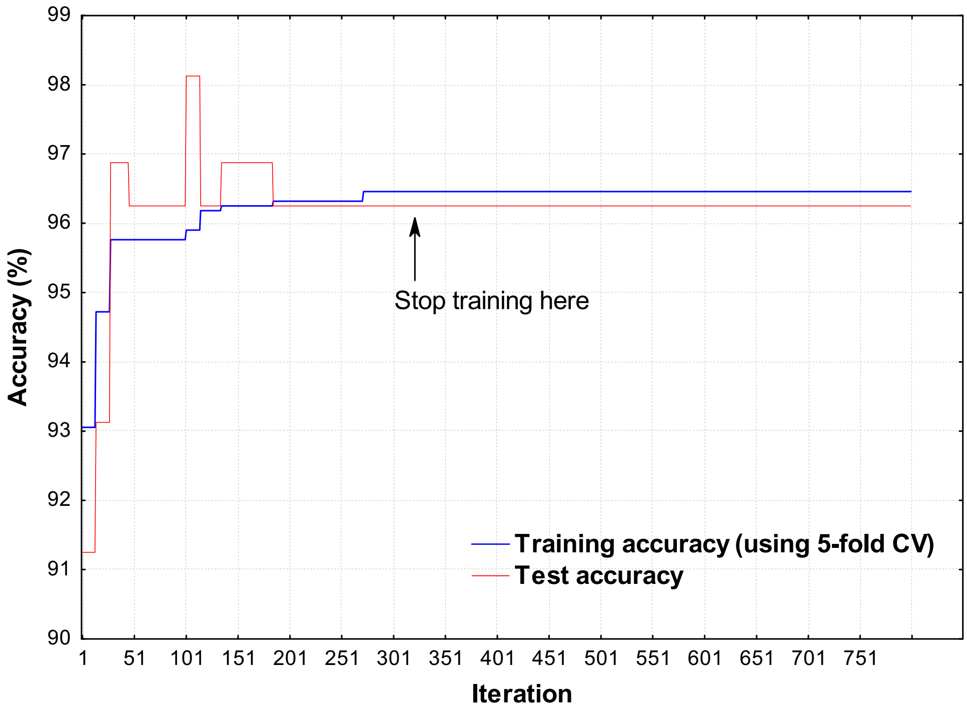


Fig. 3. Training and test accuracy plotted according to one of the 10 folds of the simulated dataset.

Table 2
Summary of results obtained from the simulated dataset using ACO–SVM (at iteration 320) and Grid–SVM.

Fold	Grid–SVM			ACO–SVM				
	Training acc. (%)	Test acc. (%)	Training time (min)	Feature size	Correct feature ratio	Training acc. (%)	Test acc. (%)	Training time (min)
#1	88.40	89.38	99.3	5	1.0	96.25	96.94	772.8
#2	89.58	89.38	120.6	4	0.8	98.13	97.15	778.8
#3	90.14	90.00	96.0	5	1.0	97.50	96.81	756.9
#4	88.40	89.38	119.8	5	1.0	98.13	97.08	566.3
#5	88.75	92.50	121.5	5	1.0	98.75	97.22	579.7
#6	89.10	91.25	117.0	4	0.8	96.25	97.29	418.8
#7	90.28	78.75	112.8	4	0.8	98.13	97.01	571.7
#8	88.33	90.63	125.5	5	1.0	95.63	96.88	405.1
#9	89.31	91.25	127.1	5	1.0	96.88	96.39	577.1
#10	89.65	88.75	112.0	5	1.0	96.25	96.46	391.6
Avg.	89.19	89.13	115.15	4.70	0.94	97.19	96.92	581.88
Dev.	0.681	3.625	9.888	0.458	0.092	1.017	0.288	141.523

as follows:

$$CFRatio = \frac{(F_{\text{discriminated}} \cap F_{\text{selected}})}{(F_{\text{discriminated}} \cup F_{\text{selected}})} \tag{23}$$

In *CFRatio*, # denotes the size of a set; F_{selected} represents the feature subset selected by the ACO–SVM, and $F_{\text{discriminated}}$ denotes the correct discriminating features (f2, f4, f6, f8, and f10 in this experiment). The *CFRatio* (≤ 1) indicates the accuracy of feature selection, such that a *CFRatio* of "1" indicates that the correct discriminating features are chosen. ACO–SVM had a high *CFRatio* of 0.94, revealing that the developed approach almost selected the correct discriminating features using this simulated dataset. With the discriminating features, ACO–SVM had an average number of selected features of 4.7, and the frequency of the selected features in the ten runs is as shown in Table 3.

The training time with ACO–SVM exceeded that with Grid–SVM, since ACO–SVM performed feature selection whereas Grid–SVM did not. Nevertheless, with its ability to select features and optimize parameters, ACO–SVM improved classification accuracy with fewer input features with only 5.05 times the training time of the Grid–SVM. Moreover, with fewer and appropriate input features, the execution time will be lower in the model usage stage.

Table 3
Frequency of selected features in 10-fold cross validation using simulated dataset.

Feature#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Frequency	0	10	0	9	0	10	0	8	0	10	0	0	0	0	0

4.2. UCI datasets

Three real world datasets, Splice dataset, Image segmentation dataset, and Diabetes dataset, available from the UCI Repository of

Machine Learning Databases [45], were adopted to demonstrate and evaluate the proposed ACO–SVM hybrid system.

4.2.1. Splice dataset

The Splice dataset has 60 input features, 1000 instances, and a target attribute with binary classes. Table 4 summarizes the results from five runs using 5-fold CV with the Splice dataset at the predefined maximum iteration of 350. The ACO–SVM had an average test accuracy of 94.64%, which exceeded that (91.31%) of the Grid–SVM. The ACO–SVM significantly outperformed the grid search approach in terms of test accuracy at the 0.043 significance level according to the Wilcoxon signed rank test. ACO–SVM had a relatively small average number of selected features, seven and its most important features were 28, 29, 30, 31, 32 and 34, which were revealed by the frequency of the selected features of 5-fold cross validation, as presented in Table 5. The training time of the ACO–SVM was 3.46 times that of the Grid–SVM.

4.2.2. Image segmentation dataset

The image segmentation dataset comprises 18 input features, 2310 instances and a target attribute with seven classes. Table 6 summarizes the results from five runs via 5-fold CV with the image segmentation dataset at the predefined maximum iteration, 250. ACO–SVM had an average test accuracy of 94.76%, which was slightly lower than the 94.85% for the Grid–SVM; however, these two average test accuracies did not differ significantly at the 0.72 significance level, based on the Wilcoxon signed rank test. Hence, the ACO–SVM was as good as Grid–SVM, according to the test accuracy in the image segmentation dataset. ACO–SVM had an average number of selected features of 13.20, and all of these features were important, except for 3, 5, 6 and 8, as revealed the frequency of selected features of 5-fold cross validation, as indicated in Table 7. The training time for the ACO–SVM was 4.48 times that of the Grid–SVM.

Table 4
Results concerning Splice dataset obtained using ACO–SVM (at iteration of 350) and Grid–SVM.

Fold	Grid–SVM		ACO–SVM		
	Test acc.	Training time	Feature size	Test acc.	Training time
#1	89.12	886.8	6	93.38	3478.9
#2	91.17	875.9	7	94.48	3568.3
#3	92.11	948.6	8	95.11	2821.5
#4	92.27	932.9	7	95.43	2678.6
#5	91.86	752.2	7	94.84	2672.2
Avg.	91.31	879.28	7.00	94.65	3043.90
Dev.	1.158	69.134	0.632	0.708	396.305

Table 5
Frequency of selected features in 5-fold cross validation of Splice dataset.

Feature#	16	21	25	28	29	30	31	32	33	34	35	others
Frequency	1	1	1	5	5	5	5	5	1	5	1	0

4.2.3. Diabetes dataset

The Diabetes dataset consists of eight input features, 760 instances and a target attribute with binary classes. Table 8 summarizes the results from 10 runs using 10-fold CV at the predefined maximum iteration, 300. ACO–SVM had an average test accuracy of 76.28%, which was slightly less than the 76.68% of the Grid–SVM; however, these two average test accuracies did not differ significantly at the 0.62 significance level based on the Wilcoxon signed rank test. Accordingly, the ACO–SVM was as good as Grid–SVM in terms of the test accuracy with the Diabetes dataset. ACO–SVM had an average number of selected features of 5.40, and its most important features were 2, 6, 7 and 8, which can be found in the frequency of selected features of 10-fold cross-validation as shown in Table 9. The training time for the ACO–SVM was 2.06 times that of the Grid–SVM.

4.3. Comparison with filter-based feature selection approach

We conducted a performance comparison between the proposed wrapper-based ACO (ACO–SVM) and the filter-based ACO (FACO–SVM). The FACO–SVM was performed N_F times to find a proper feature subset as follows. The feature importance measured by the F-score (Eq. (9)) for each feature was calculated and sorted. For the possible number of features $f, f \in \{1, 2, \ldots, N_F\}$, where N_F is the total number of features in a data set, perform the following two steps: keep the first f features according to the sorted F-scores and train the FACO–SVM based on the selected feature subset.

For the simulated dataset with 15 features, we trained 15 models to find the best test accuracy and feature subset for each

Table 7
Frequency of selected features in 5-fold cross validation experiment on image segmentation dataset.

Feature#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Frequency	5	5	0	3	0	2	4	2	4	5	4	4	4	5	5	5	4	5

Table 6
Results concerning image segmentation dataset obtained using ACO–SVM (at iteration of 250) and Grid–SVM.

Fold	Grid–SVM		ACO–SVM		
	Test acc. (%)	Training time (min)	Feature size	Test acc. (%)	Training time (min)
#1	97.40	151.8	15	97.40	662.7
#2	97.62	147.7	15	97.84	677.7
#3	93.29	145.8	12	93.94	663.9
#4	95.24	146.9	11	94.37	627.9
#5	90.69	138.4	13	90.26	641.0
Avg.	94.85	146.11	13.20	94.76	654.64
Dev.	2.610	4.384	1.600	2.739	17.796

Table 8
Results concerning Diabetes dataset obtained using ACO–SVM (at iteration of 300) and Grid–SVM.

Fold	Grid–SVM		ACO–SVM		
	Test acc. (%)	Training time (min)	Feature size	Test acc. (%)	Training time (min)
#1	73.68	109.47	5	72.37	233.28
#2	77.92	119.86	5	81.82	239.60
#3	77.92	111.93	7	75.33	214.25
#4	81.82	118.50	5	80.52	228.83
#5	81.82	102.64	5	77.92	270.22
#6	76.62	120.86	7	76.62	289.04
#7	77.92	124.27	5	75.33	234.15
#8	74.03	112.26	6	79.22	225.91
#9	76.62	118.13	4	76.62	221.01
#10	68.42	117.28	5	67.11	220.10
Avg.	76.68	115.52	5.40	76.28	237.64
Dev.	3.764	6.073	0.917	4.019	22.550

Table 9
Frequency of selected features in 10-fold cross validation experiment on Diabetes dataset.

Feature#	1	2	3	4	5	6	7	8
Frequency	2	10	6	1	5	10	10	10

fold as shown in Fig. 4. The detailed performance on the test and training accuracy, feature subset, and training time are shown in Table 10. We can notice that high test accuracies were obtained when a proper size of feature subset was found (e.g., size of 4, 5, or 6). Table 11 summarizes the average performance for both approaches, ACO–SVM and FACO–SVM, using the simulated dataset and the three UCI datasets (the average performance of ACO–SVM are from Tables 2, 4, 6 and 8). For the simulated dataset, both approaches had similar best test accuracy, best training accuracy, and size of the best feature subset; the training time for the ACO-SVM was longer than the average training time (as well as the single runtime) for the FACO-SVM; however, the total training time for the ACO–SVM was significantly shorter than that of the FACO–SVM. For the three UCI datasets, both approaches maintained similar test accuracy; nevertheless, the proposed ACO–SVM had a smaller size of best feature subset and a shorter total runtime than the FACO–SVM, especially for the dataset with a large number of features.

4.4. Discussion of experimental results

The results obtained using simulated dataset demonstrate that the proposed ACO–SVM successfully selected a correct feature subset with a high classification accuracy of 96.92%, which was similar to the value 97% reported elsewhere [43] obtained using a GA-based K-nearest-neighbor (GA-KNN) feature weighting approach. The results of the three UCI datasets showed that the proposed ACO–SVM maintains competitive classification accuracy but had a smaller feature subset than the Grid–SVM, which performed grid search on the SVM parameters without feature selection. Compared with the filter-based FACO–SVM, the proposed wrapper-based ACO–SVM maintained competitive classification accuracy and had a shorter total training time and a smaller size of feature subset, especially for the dataset with a large number of features.

The training time of the wrapper-based ACO–SVM can be reduced by starting a rough search on the SVM parameters, and conducting a fine search by increasing the search resolution of the SVM parameters following a specified iteration. Some parameters (ρ, e, φ, c, α and β) of the ACO system may also influence the experimental results; this study reports preliminary results with a satisfactory outcome after some possible system parameters were considered.

5. Conclusions

Feature selection is an important issue in the construction of classification systems. The number of input features in a classifier should be limited to ensure good predictive power without an excessively computationally intensive model. With a smaller feature set, the classification decision is more easily explained. This work investigated a novel hybrid ACO-based model that hybridized the ant colony optimization and support vector machines to maintain the classification accuracy with a small and suitable feature subset. The merit of our proposed approach is that the overall classification accuracy and the information on the feature importance that is provided by the classifier can be tightly integrated together into the ACO algorithm. This work is novel, since no empirical research has been conducted on the hybrid ACO–SVM classification system to find simultaneously an optimal feature subset and model parameters.

Experimental results concerning a simulated dataset revealed that the proposed approach not only optimized the classifier's model parameters and correctly obtained the discriminating feature subset, but also achieved accurate classification accuracy. Experimental results on three public UCI datasets showed promising performance in terms of test accuracy, the size of the selected feature subset and training time. Some possible extensions of this research are as follows:

- This work used the discrete version of ACO, in which the continuous classifier parameter values such as C and γ in the SVM were discretized into several search points. The interval length determined the search resolution of the discrete ACO, impacting on the solution quality and the search time. Although the discrete ACO yielded positive results herein, the continuous ACO [46–49] may be a good alternative for optimizing the continuous classifier parameter values.
- This work showed experimental results with the RBF kernel. Other kernel parameters can also be optimized using the same procedure. Furthermore, the proposed approach can be modified and applied to support vector regression (SVR). The kernel parameters and input features significantly affect the

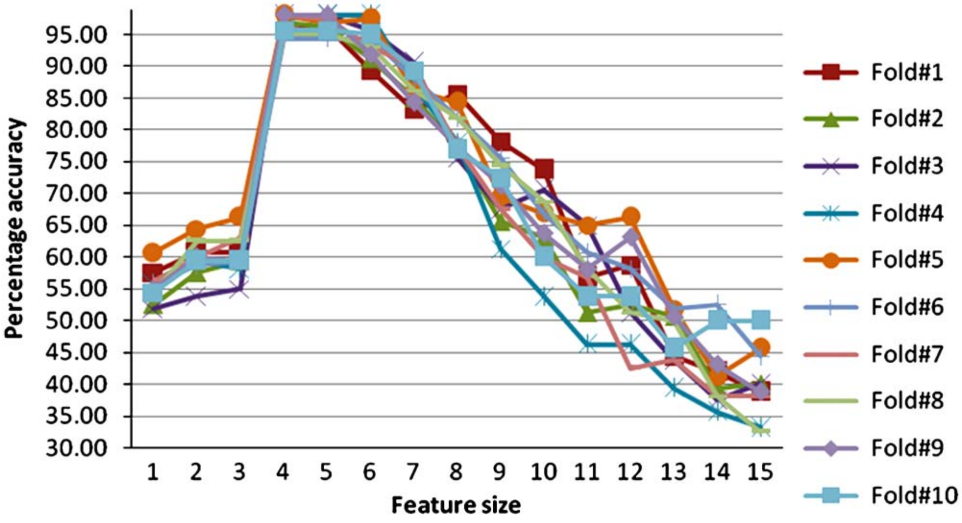


Fig. 4. Best test accuracy and size of feature subset for the simulated dataset obtained using FACO–SVM.

Table 10
Results concerning simulated dataset obtained using FACO–SVM.

Fold	Best feature size[a]	Best test acc. (%)[b]	Average test acc. (%)[c]	Best training acc. (%)[d]	Average training acc. (%)[e]	Total training time (min)[f]	Average training time (min)[g]
#1	4	96.25	68.08	97.22	84.55	1530.0	102.0
#2	4	96.88	65.25	96.94	84.51	1705.5	113.7
#3	5	98.13	66.29	96.88	84.81	1281.0	85.4
#4	6	98.13	63.17	97.22	84.25	1605.0	107.0
#5	4	98.13	70.71	96.67	84.07	1791.0	119.4
#6	6	94.38	69.13	95.69	84.51	1185.0	79.0
#7	4	98.13	65.50	96.88	85.11	1653.0	110.2
#8	5	95.00	66.92	96.46	84.43	1693.5	112.9
#9	5	98.13	67.54	95.97	84.45	1576.5	105.1
#10	5	95.63	67.42	97.15	84.80	1495.5	99.7
Avg.	4.80	96.88	67.00	96.71	84.55	1551.60	103.44
Dev.	0.75	1.40	2.01	0.50	0.28	180.59	12.04

[a] Size of features with the best test accuracy among 15 models (runs) with feature size from one to 15.
[b] Best test accuracy among 15 models.
[c] Average test accuracy of 15 models.
[d] Best training accuracy among 15 models.
[e] Average training accuracy of 15 models.
[f] Total training time for the 15 models.
[g] Average training time for the 15 models.

Table 11
Summary of results obtained from the simulated dataset using ACO–SVM and FACO–SVM.

Dataset	Model	Best feature size	Best test acc. (%)	Average test acc. (%)	Total training time (min)	Average training time (min)
Simulated	ACO–SVM	4.7	96.92	n/a[a]	581.88	n/a
	FACO–SVM	4.8	96.88	67.00	1551.60	103.44
Diabetes	ACO–SVM	5.4	76.68	n/a	237.64	n/a
	FACO–SVM	6.1	75.83	63.57	764.67	95.584
Image	ACO–SVM	13.2	94.76	n/a	654.64	n/a
	FACO–SVM	16.7	94.72	54.64	2415.6	134.2
Splice	ACO–SVM	7.0	94.65	n/a	3043.9	n/a
	FACO–SVM	12.3	93.14	64.73	31219.2	520.32

[a] Not available.

predictive accuracy of the SVR with a kernel function. Similar ACO-based feature selection and parameters optimization procedures can be adopted to improve the SVR accuracy.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments, which helped in improving the quality of this paper. The author would also like to thank Mr. Cheng-Yi Tsai for his support on collecting the experimental data.

References

[1] G.P. Zhang, Neural networks for classification: a survey, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 30 (4) (2000) 451–462.

[2] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems 13 (2) (1998) 44–49.

[3] G. John, R. Kohavi, K. Peger, Irrelevant features and the subset selection problem, in: Proceedings of the Eleventh International Conference on Machine Learning, 1994, pp. 121–129.

[4] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.

[5] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic, Norwell, MA, 1998.

[6] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, NY, USA, 1995.

[7] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Transactions on System, Man, and Cybernetics 26 (1) (1996) 1–13.

[8] M. Dorigo, L.M. Gambardella, Ant colonies for the traveling salesman problem, BioSystems 43 (1997) 73–81.

[9] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.

[10] M. Dorigo, G.D. Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artificial Life 5 (2) (1999) 137–172.

[11] D. Merkle, M. Middendorf, H. Schmeck, Ant colony optimization for resource-constrained project scheduling, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2000, pp. 893–900.

[12] S. Izrailev, D.K. Agrafiotis, Variable selection for QSAR by artificial ant colony systems, SAR and QSAR in Environmental Research 13 (3–4) (2002) 417–423.

[13] A. Al-Ani, Feature subset selection using ant colony optimization, International Journal of Computational Intelligence 2 (1) (2005) 53–58.

[14] R.K. Sivagaminathan, S. Ramakrishnan, A hybrid approach for feature subset selection using neural networks and ant colony optimization, Expert Systems with Applications 33 (1) (2007) 49–60.

[15] H.R. Kanan, K. Faez, M. Hosseinzadeh, Face recognition system using ant colony optimization-based selected features. in: IEEE Symposium on Computational Intelligence in Security and Defense Applications, 2007, pp. 57–62.

[16] R. Jensen, Q. Shen, Fuzzy-rough data reduction with ant colony optimization, Fuzzy Sets and Systems 149 (1) (2005) 5–20.

[17] R. Jensen, Q. Shen, Webpage classification with ACO-enhanced fuzzy-rough feature selection, Lecture Notes in Computer Science, vol. 4259, Springer, Berlin, 2006, pp. 147–156.

[18] R.B. Perez, A. Nowe, P. Vrancx, Y.D. Gomez, Y. Caballero, Using ACO and rough set theory to feature selection, WSEAS Transactions on Information Science and Applications 2 (5) (2005) 512–517.

[19] Z. Yan, C. Yuan, Ant colony optimization for feature selection in face recognition, Lecture Notes in Computer Science, vol. 3072, Springer, Berlin, 2004, pp. 221–226.

[20] H.-H. Gao, H.-H. Yang, X.-Y. Wang, Ant colony optimization based network intrusion feature selection and detection, in: International Conference on Machine Learning and Cybernetics, 2005, pp. 3871–3875.

[21] C. Zhang, H. Hu, Ant colony optimization combining with mutual information for feature selection in support vector machines, Lecture Notes in Computer Science, vol. 3809, Springer, Berlin, 2005, pp. 918–921.

[22] S.S. Mohamed, A.M. Youssef, E.F. El-Saadany, M.M.A. Salama, Artificial life feature selection techniques for prostrate cancer diagnosis using TRUS images, Lecture Notes in Computer Science, vol. 3656, Springer, Berlin, 2005, pp. 903–913.

[23] C.-L. Huang, C.-J. Wang, A GA-based attribute selection and parameter optimization for support vector machine, Expert Systems with Applications 31 (2) (2006) 231–240.

[24] H. Fröhlich, O. Chapelle, Feature selection for support vector machines by means of genetic algorithms, in: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, pp. 142–148.

[25] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition, International Journal of Pattern Recognition and Artificial Intelligence 17 (6) (2003) 903–929.

[26] V. Kecman, Learning and Soft Computing, MIT Press, Cambridge, Massachusetts, USA, 2001.

[27] B. Schőlkopf, A.J. Smola, Statistical Learning and Kernel Methods, MIT Press, Cambridge, Massachusetts, USA, 2000.

[28] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 2000.

[29] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, V. Vapnik, Comparison of classifier methods: a case study in handwriting digit recognition, in: International Conference on Pattern Recognition, 1994, pp. 77–87.

[30] S. Knerr, L. Personnaz, G. Dreyfus, Single-layer learning revisited: a stepwise procedure for building and training a neural network, in: F. Fogelman-Soulíe, J. Herault (Eds.), Neurocomputing: Algorithms, Architectures and Application, vol. F68, Springer, Berlin, 1990, pp. 41–50.

[31] U. KreBel, Pairwise classification and support vector machines, in: B. Schölkopf, C.J. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, Massachusetts, USA, 1999, pp. 254–268.

[32] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, Cambridge, Massachusetts, 2004.

[33] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, Available at: 〈http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf〉, January 2008.

[34] H.-T. Lin, C.-J. Lin, A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, Available at: 〈http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf〉, January 2008.

[35] Y.-W. Chen, C.-J. Lin, Combining SVMs with various feature selection strategies, in: I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh (Eds.), Feature Extraction, Foundations and Applications, Studies in Fuzziness and Soft Computing, Springer, Berlin, 2005, pp. 273–282.

[36] S.L. Salzberg, On comparing classifiers: Pitfalls to avoid and a recommended approach, Data Mining and Knowledge Discovery 1 (1997) 317–327.

[37] D. Morariu, L. Vintan, V. Tresp, Evaluating some feature selection methods for an improved SVM classifier, International Journal of Intelligent Technology 1 (4) (2006) 288–298.

[38] S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, Journal of Machine Learning Research 2 (2001) 243–264.

[39] S. Shalev-Shwartz, N. Srebro, SVM optimization: inverse dependence on training set size, in: W.W. Cohen, A. McCallum, S.T. Roweis (Eds.), The 25th International Conference on Machine Learning, Berlin, Springer, 2008, pp. 928–935.

[40] C.W. Hsu, C.J. Lin, A simple decomposition method for support vector machine, Machine Learning 46 (1–3) (2002) 219–314.

[41] S.M. LaValle, M.S. Branicky, S.R. Lindemann, On the relationship between classical grid search and probabilistic roadmaps, International Journal of Robotics Research 23 (7–8) (2004) 673–692.

[42] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, Software available at: 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm〉, January 2008.

[43] W.F. Punch, E.D. Goodman, M. Pei, L. Chia-Shun, P. Hovland, R. Enbody, Further research on feature selection and classification using genetic algorithms, in: S. Forrest (Ed.), Fifth International Conference on Genetic Algorithms, , 1993, pp. 557–564.

[44] W. Conover, Practical Nonparametric Statistics, second ed., Wiley, New York, 1980.

[45] P.M. Murphy, D.W. Aha, UCI Repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 〈http://www.ics.uci.edu/~mlearn/MLRepository.html〉, 2001.

[46] G. Bilchev, I.C. Parmee, The ant colony metaphor for searching continuous design space, in: T. Fogarty (Ed.), Proceeding of the AISB Workshop on Evolutionary Computation, Lecture Notes in Computer Science, vol. 993, Springer, Berlin, 1995, pp. 25–39.

[47] W. Lei, W. Qidi, Ant system algorithm for optimization in continuous space, in: Proceeding of the IEEE International Conference on Control Applications, 2001, pp. 395–400.

[48] L. Chen, J. Shen, L. Qin, J. Fan, A method for solving optimization problem in continuous space using improved ant colony algorithm, in: Y. Shi, W. Xu, Z. Chen (Eds.), CASDMKM 2004, Lecture Notes in Computer Science, vol. 3327, Springer, Berlin, 2004, pp. 61–70.

[49] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185 (3) (2008) 1155–1173.

Cheng-Lung Huang received his M.S. and Ph.D. degrees in Industrial Engineering from the National Chiao-Tung University, Hsinchu, Taiwan, ROC, in 1990 and 1998, respectively. He is currently an associate professor of the Department of Information Management, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan. Now he is also the director of the Laboratory of Business Intelligence and Data Mining at the National Kaohsiung First University of Science and Technology. His research interests include neural networks, kernel methods, swarm intelligence, evolutionary computing and data mining.