Feature selection and model selection algorithm using incremental mixed variable ant colony optimization for support vector machine classifier

Hiba Basim Alwan and Ku Ruhana Ku-Mahamud

***Abstract*—** Support Vector Machine (SVM) is a present day classification approach originated from statistical approaches. Two main problems that influence the performance of SVM are selecting feature subset and SVM model selection. In order to enhance SVM performance, these problems must be solved simultaneously because error produced from the feature subset selection phase will affect the values of the SVM parameters and resulted in low classification accuracy. Most approaches related with solving SVM model selection problem will discretize the continuous value of SVM parameters which will influence its performance. Incremental Mixed Variable Ant Colony Optimization (IACO$_{MV}$) has the ability to solve SVM model selection problem without discretising the continuous values and simultaneously solve the two problems. This paper presents an algorithm that integrates IACO$_{MV}$ and SVM. Ten datasets from UCI were used to evaluate the performance of the proposed algorithm. Results showed that the proposed algorithm can enhance the classification accuracy with small number of features.

Keywords— Incremental Mixed Variable Ant Colony Optimization, Support Vector Machine, Features Selection, Parameter Optimization, Pattern Classification.

I. INTRODUCTION

CLASSIFICATION is a supervised learning approach which is a significant field of research that involved labelling an object to one of a group of classes, related to features of that object and it is considered one of the basic difficulties in many decision making processes [1]. Many decision making processes are examples of classification problem and examples of these problems are prognosis processes, diagnosis processes and pattern recognition [2]. The majority of recent researches centred on enhancing classification accuracy by utilizing statistical approaches. Pattern classification attaches the input samples into one of a present number of groups through an approach. The approach is found through learning the training data group [3]. Certain pattern classification approaches allow the input data to include many features but in reality, only a few of them are relevant to classification. In certain circumstances, it is not suitable to choose a large number of features, as this may be difficult in calculating or it might be incomplete [4]. Therefore, choosing few and related features has its benefits where it minimizes both the calculation effort and complexity of the approach as the generalization capability may be enhanced [5].

Feature selection (FS) is a process of determining a subset of fields in database and it minimizes the number of fields that appears during data classification [6]. The main idea behind FS is to select a subset of input variables by deleting features that contain less or no information [4]. FS aims to decrease the dimension of the initial features group by determining the unauthentic features which would eventually supply the best performance under certain classification dataset [7], and to delete unrelated, unneeded, or noisy features while preserving the richness of the instructive ones. FS may be considered as an optimization problem which looks out for potential feature subsets which ultimately determines the optimal one [8].

Support Vector Machine (SVM) represents supervised machine learning approaches [31] and it is an excellent classifier built on statistical learning approach, but it is not able to avoid the influence of huge number of unrelated or redundant features on the classification results. Therefore, selecting a few numbers of suitable features would result in obtaining good classification accuracy [9]. The main concept of SVM is to obtain the Optimal Separating Hyperplane (OSH) between the positive and negative samples. This can be done through maximizing the margin between two parallel hyperplanes. Finding this plane, SVM can then forecast the classification of unlabeled sample through asking on which side of the separating plan the sample lies [10].

An algorithm that is based on IACO$_R$ and ACO$_{MV}$ to optimize SVM mixed variables has been proposed in [28]. The work presented in [28] is an extension of [13] that can simultaneously optimize SVM parameters and features subset. IACO$_{MV}$ is known to have the ability to optimize discrete and continuous variables [12]. Selecting the optimal feature subset and tuning SVM parameters to be used in SVM classifier are two problems in SVM classifier that influences the classification accuracy. These problems affect each other [11].

Hiba Basim Alwan is with the School of Computing, College of Art and Science, University Utara Malaysia, 06010 Sintok, Kedah, Malaysia, (e-mail: hiba81basim@yahoo.com).

Ku Ruhana Ku-Mahamud is a Professor with the School of Computing, College of Art and Science, University Utara Malaysia, 06010 Sintok, Kedah, Malaysia, (e-mail: ruhana@uum.edu.my).

This paper presents an algorithm that integrates IACO$_{MV}$ and SVM that can be used to overcome the above mentioned problems. The rest of the paper is organized as follows. Section 2 presents a brief introduction to SVM while Section 3 presents the concept of IACO$_{MV}$. Section 4 reviews several literatures on simultaneously optimize SVM parameters and features subset and Section 5 describes the proposed algorithm. Section 6 presents the findings and concluding remarks and future works are presented in Section 7.

II. Support Vector Machine

For binary class classification problem, given *M* training examples where each example is represented through a pair of (x_i, y_i) where $i = 1, \ldots, M$, $x_i \in R^M$ corresponds to the feature group for the i^{th} example, and $y_i \in \{+1, -1\}$ denoted the class label, SVM need to find the optimal hyperplane that will classify each pattern x_i into the correct class y_i. If the patterns are linearly separable, the following expressions can be used to give the parameters *w* and *b* of the hyperplane [14]:

$$\langle w.x_i\rangle + b \geq +1 \, for \, y_i = +1 \tag{1}$$

$$\langle w.x_i\rangle + b \leq -1 \, for \, y_i = -1 \tag{2}$$

Gathering inequalities (1) and (2) gives:

$$y_i(\langle w.x_i\rangle + b) - 1 \geq 0 \; \forall i = 1, \ldots, M \tag{3}$$

The SVM obtains the optimal hyperplane through solving the following minimization problem [11]:

$$min_{w,b} \frac{1}{2} w^T w \tag{4}$$

Subject to$y_i\langle w.x_i\rangle + b) - 1 \geq 0$ (5)

To solve this quadratic optimization problem one must obtain the saddle point of the Lagrangian function [14]:

$$L_P(w,b,\alpha) = \frac{1}{2} w^T.w - \sum_{i=1}^{M}(\alpha_i y_i(\langle w.x_i\rangle + b) - 1) \tag{6}$$

where α_i represents Lagrange multipliers; $\alpha_i > 0$. The saddle point can be located by minimizing the Lagrangian function L_p with respect to the primal variable *w* and *b* and maximizing L_p with respect to the non-negative dual variable α_i. The following equations are produced after differentiating Eq. (6) with respect to *w* and *b* 14]:

$$\frac{\partial}{\partial w} L_P = 0, w = \sum_{i=1}^{M} \alpha_i x_i y_i \tag{7}$$

$$\frac{\partial}{\partial b} L_P = 0, \; \sum_{i=1}^{M} \alpha_i y_i \tag{8}$$

Substituting Eqs. (7) and (8) into Eq. (6) yields the dual Lagrangian L_D to be maximized [14]:

$$Max_{\alpha_i} L_D(\alpha_i) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j \langle x_i.x_j\rangle \tag{9}$$

Subject to$\alpha_i > 0$, $i = 1, \ldots, M$, and $\sum_{i=1}^{M} \alpha_i y_i = 0$

As mentioned previously, to obtain the optimal hyperplane, maximizing the dual Lagrangian $L_D(\alpha_i)$ with respect to non-negative α_i is needed. This quadratic optimization problem can be solved by utilizing a standard optimization program. When the optimal values α_i^* of α_i have been computed, the optimal decision hyperplane is given by [21]:

$$f(x, \alpha_i^*, b^*) = \sum_{i=1}^{M} y_i \alpha_i^* \langle x_i.x\rangle + b^* \tag{10}$$

For non-zero α_i^*, b^* can be obtained from the Kuhn-Tucker condition:

$$y_i(x_i w^* + b^*) - 1 = 0 \, for \, i = 1, \ldots, M \tag{11}$$

where, utilizing Eq. (7) [21]:

$$w^* = \sum_{i=1}^{M} \alpha_i^* y_i x_i \tag{12}$$

Note that vectors x_i for which Eq. (11) holds are called support vectors.

In non-separable cases, the goal is to build a hyperplane that will generate the smallest number of classification mistakes. Slack variables $\xi_i \geq 0, i = 1, \ldots, M$ are introduced in Inequalities Eq. (1) and Eq. (2) such that [21]:

$$\langle w.x_i\rangle + b \geq +1 - \xi_i \, for \, y_i = +1 \tag{13}$$

$$\langle w.x_i\rangle + b \leq -1 + \xi_i \, for \, y_i = -1 \tag{14}$$

ξ_i relax the constraints on the location of the data relative to the hyperplane. The optimization problem becomes [20]:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{M} \xi_i \tag{15}$$

Subject to $y_i(\langle w.x_i\rangle + b) + \xi_i - 1, \xi_i \geq 0, i = 1, \ldots, M$
where *C* represent the penalty of misclassifying the training instances in other words, it is a weight representing the trade-off between misclassifying certain points and correctly classifying others.

Again, the Lagrangian method can be utilized to solve the above optimization problem. The Lagrangian $L_D(\alpha_i)$ is to be maximized, i.e. [14]:

$$Max_{\alpha_i} L_D(\alpha_i) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j \langle x_i.x_j\rangle \tag{16}$$

Subject to$0 \leq \alpha_i \leq C$, $i = 1, \ldots, M$, and $\sum_{i=1}^{M} \alpha_i y_i = 0$

Note that Eq. (16) is the same as Eq. (9) for the case of linearly separable data, except that α_i is now bounded by *C*. The optimum hyperplane can be found as described previously once the values α_i have been computed.

In most cases, the data are not linearly separable, and are consequently mapped to a higher dimensional feature space. Therefore, if the data cannot be classified clearly in the current dimensional space, then the SVM will map them to a higher dimensional space for classification [21]. Input data are mapped to a higher dimensional feature space by plotting a nonlinear curve utilizing kernel function $\phi(x_i) = k\,(.\;,\;.)$. When applied to two points x_i and x_j, $k\,(x_i, x_j)$, is a generalized form of the inner product in Eq. (9). The Lagrangian maximization problem becomes [14]:

$$Max_{\alpha_i} L_D(\alpha_i) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j k\langle x_i.x_j\rangle \tag{17}$$

Subject to$0 \leq \alpha_i \leq C$, $i = 1, \ldots, M$, and $\sum_{i=1}^{M} \alpha_i y_i = 0$. The kernel function used in this study is Radial Basis Function (RBF) as shown below:

$$K(x_i, x) = exp^{(-\gamma\|x_i - x\|^2)} \tag{18}$$

The classification decision function formula for testing data becomes [14]:

$$sgn(w^T\phi(x) + b) = sgn(\sum_{i=1}^{M} \alpha_i \, y_i K(x_i, x) + b) \tag{19}$$

For multi class SVM, One-Against-One (OAO) strategy is used here in this paper. OAO constructs number of binary SVM classifier where each one is trained on data from two classes. The number of binary SVM classifier is compute according to the following formula [20]:

$$C_2^v = \frac{v(v-1)}{2} \tag{20}$$

In classification, a voting strategy was used where each binary classification is tested against the point and a class is assigned. The point will later be assigned to the class with the higher occurrence. In the case of two classes having identical votes, the class that appears first in the array of the stored class names will be chosen [15].

III. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a metaheuristic approach for hard discrete optimization problems that was initially introduced in the beginning of 1990s. The word heuristic comes from Greek and means to know, to find, to discover, or to guide an investigation [30]. ACO is based on the behaviour of real ants in collecting food. Ant when seeking for food will initially investigate the region bordering their nest in an unstructured way. When ant detects a food source, it evaluates the quantity and quality of the food and holds some of it to the nest. Throughout the return trip, the ant deposits pheromone on the path and the amount of pheromone depend on the amount and quality of the food. This pheromone will be used to lead other ants to the food source. This situation will help other ants to find the shortest paths between their nest and food sources [16]. In order to solve optimization problem, ACO will repeat the following two steps [17]: nominee solutions are built using the pheromone and the nominee solutions are used to update the pheromone values in order to get high quality solutions.

ACO which was firstly presented to solve discrete optimization problem, has now been modified to solve continuous and mixed optimization problems. Several researches have focused on the expansion of ACO for continuous and mixed-parameter optimization problems. One of the most interested ACO for continuous variables and mixed variables is Socha work which is called continuous ACO (ACO_R) and mixed-variable ACO (ACO_MV) respectively [12] and [17]. ACO_R is later modified by [12] and two new algorithms called Incremental ACO_R (IACO_R) and Incremental ACO_R with Local Search (IACO_R-LS) have been introduced. All four ACO variants follow the same classical ACO framework except that the discrete probability used to build ant's solution was replaced by continuous probability. [12] suggested adopting either IACO_R or IACO_R-LS to optimizing continuous variable to optimise in the ACO_MV introduced in [18].

A. *Ant Colony Optimization for Continuous Variable*

Probability Density Function (PDF) has been used in Ant Colony Optimization for continuous variable (ACO_R) introduced by [18]. Distribution to determine the direction that an ant should follow. Gaussian function is one of the most popular PDF as it adopts a very simple manner for data sampling. For each built solution, a density function is generated from a set of *k* solutions that the technique preserves at all times. In order to maintain this set, the set is filled with unstructured or not systematic solutions at the beginning. This is similar to initializing pheromone value in a discrete ACO approach. Then, at each loop, the group of created *m* solutions is appended to the set and the equivalent number of worst solutions is deleted from the set to preserve just the best *k* solutions of the *k* + *m* solutions that are available. This work is similar to pheromone modification in discrete ACO. The goal is to influence the searching procedure to gain the best solution. Pheromone information is kept in a table when ACO for discrete combinatorial optimization is used. During each loop, when selecting a component to be appended to the current partial solution, an ant utilizes part of the values from that table as a discrete probability distribution. In contrast to the situation of continuous optimization, the selection that the ant makes is not limited to a finite group. Therefore, it is difficult to express the pheromone in the table structure. Instead of using a table, ACO_R uses solution archive to preserve the route for a number of solutions. Solution archive contains values of solution variables and objective functions. The solution will be established through each ant. In establishing a solution path, the solution archive is needed to design the transition probabilities. The weight vector, *w* is computed for each solution stored in solution archive as follows:

$$w_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \tag{21}$$

where *k* is the size of solution archive, and *q* is the algorithm's parameter to control diversification of search process. The weights values are also stored in solution archive. Once this step is completed, the sampling procedure is made through two phases. Phase one involves choosing one of the weight vectors according to its probability as follows:

$$p_l = \frac{w_l}{\sum_{r=1}^{k} w_r} \tag{22}$$

The second phase involves sampling selecting weight (*w*) via a random number generator that is able to generate random numbers according to a parameterized normal distribution. This initialization constructs the transition probabilities for ants [18]. An outline of ACO_R is given in Figure 1 [12]:

```
Input: $k, m, D, q, \xi$, and termination criterion
Output: The best solution found
Initialize and evaluate $k$ solutions
//Sort solutions and store them in the archive
$T$ = Sort ($S_1, \ldots, S_k$)
While Termination criterion is not satisfied do
  //Generate $m$ new solutions
  for $l$ = 1 to $m$ do
    //Construct solution
    for $i$ = 1 to $D$ do
      Select Gaussian $g_j^i$ according to weights
      Sample Gaussian $g_j^i$ with parameters $\mu_j^i, \sigma_j^i$
    end for
    store and evaluate newly generated solution
  end for
  //Sort solutions and select the best $k$
  $T$ = Best (Sort ($S_1, \ldots, S_{k+m}$), $k$)
End while
```

Figure 1: ACO_R Algorithm

B. *Mixed Variable Ant Colony Optimization*

Ant Colony Optimization for Mixed Variable (ACO_MV) was also introduced by [18] which extends Ant Colony Optimization for continuous variable (ACO_R). Continuous variables are treated as in the original ACO_R while discrete variables are treated differently. In standard ACO, solutions are constructed from solution components using a probabilistic rule based on the pheromone values. However, there is no static pheromone value, but only a solution archive in ACO_MV. As in standard ACO, the construction of solutions for discrete variables is done by choosing the components, that is, the values for each of the discrete decision variables. However, since the static pheromone values of standard ACO are replaced by the solution archive, the actual probabilistic rule used has to be modified. Similarly to the case of continuous variables, each ant constructs the discrete part of the solution incrementally. For each discrete variable, each ant will chose one of the solution components C^i based on a probability. The probability of choosing the l^{th} value is given by:

$$o_l^i = \frac{w_l}{\sum_{r=1}^{c} w_r} \qquad (23)$$

where w_l is the weight associated with the l^{th} available value. It is calculated based on the weights w and some additional parameters:

$$w_l = \frac{w_{jl}}{u_l^i} + \frac{q}{\eta} \qquad (24)$$

The final weight, $w_{l,}$ is hence a sum of two components. The weight, w_{jl}, is calculated according to Eq. 21, where the j_l is the index of the highest quality solution that uses value v_l^i for the i^{th} variable. In turn, u_l^i is the number of solutions using value v_l^i for the i^{th} variable in the archive. Therefore, the more popular the value v_l^i is, the lower is its final weight. The second component is a fixed value (i.e., it does not depend on the value v_l^i chosen): η is the number of values v_l^i from the c_i available ones that are unused by the solutions in the archive, and q is the same parameter of the algorithm that was used in Eq. 21.

C. *Incremental Continuous Ant Colony Optimization*

Incremental continuous Ant Colony Optimization (IACO$_R$) is built on enhanced ACO$_R$ [12]. It starts with a small size for solution archive defined by a parameter *InitArhiveSize*. This solution archive will be filled with initial solutions which are randomly generated. IACO$_R$ also characterizes a strategy alternate from the one utilized in ACO$_R$ for choosing the solution that directs the creation of new solutions. The new procedure build on probability parameter $p \in [0, 1]$, which monitors the probability of utilizing just the best solution in the archive as a directing solution. With a probability $1 - p$, all the solutions in the archive are utilized to create new solutions. Once a directing solution is chosen, and a new one is created exactly the same way as in ACO$_R$, they are compared according to their objective function. The newly created solution will replace the directing solution in the archive if it is better. This replacement mechanism is alternate from the one utilized in ACO$_R$ in which all solutions in the archive and all the newly created ones compete. A new solution is appended to them in every growth iterations until a maximum archives sizes, defines by *MaxArchiveSize*, is reached. A parameter *Growth* monitor the percentage at which the archives grows. Fast growth percentage support seeks diversification while slow growth support intensification. Each time a new solution is appended, it is initialized by using the information from the best solution in the archives. First, a new solution S_{new} is created fully in an arbitrary way and then it is moved in the direction of the best solution in the archives S_{best} utilizing the following formula:

$$\bar{S}_{new} = S_{new} + rand(0,1)(S_{best} - S_{new}) \qquad (25)$$

where *rand(0, 1)* is an arbitrary umber in range [0, 1). IACO$_R$ involve an algorithm-level diversification strategy to avoid stagnation. The strategy includes restarting the algorithm and initializing the new initial archive with the best-so-far solution. The restart condition is the number of successive iterations, *MaxStagIter*, with a relative solution improvement lower than a certain threshold. An outline of IACO$_R$ is given in Figure 2.

```
Input:   $p$, InitArhiveSize, Growth, MaxArchiveSize,
         MaxStagIter, no. of ants, and Termination criterion
Output: Optimal Value for $C$ and $\gamma$
$k$ = InitArhiveSize
initialize $k$ solutions and evaluate it
while Termination criterion not satisfied do
  // Generate new solutions
  if rand (0,1) < $p$ then
    for $i$ = 1 to no. of ants do
      Select best solution
      Sample best selected solution
      if Newly generated solution is better than $S_{best}$ then
        Substitute newly generated solution for $S_{best}$
      end
    end
  else
    for $j$ = 1 to $k$ do
      Select $S$ according to its weight
      Sample selected $S$
      Store and evaluate newly generate solutions
      if Newly generated solution is better than $S_j$ then
        Substitute newly generated solution for $S_j$
      end
    end
  end
  // Archive Growth
  if current iterations are multiple of Growth & $k$ <
     MaxArchiveSize then
    Initialize new solution
    Add new solution to the archive
    $k++$
  end
  // Restart Mechanism
  if # (number) of iterations without improving $S_{best}$ =
    MaxStagIter then
    Re-initialize $T$ (solution archive) but keeping $S_{best}$
  end
end
```

Figure 2: IACO$_R$ Algorithm

D. *Incremental Mixed Variable Ant Colony Optimization*

Incremental mixed variable ant colony optimization (IACO$_{MV}$) is based on [12] suggestion to enhance ACO$_{MV}$. This variant is similar to ACO$_{MV}$ except for the part related with optimizing continuous variable. In incremental mixed variable ant colony optimization, IACO$_R$ is used to optimize continuous variable instead of using ACO$_R$ as in ACO$_{MV}$. In optimizing discrete variable, incremental mixed variable ant colony optimization utilizes the same procedure that has been used in ACO$_{MV}$.

IV. SIMULTANEOUS OPTIMIZATION FEATURE SUBSET AND PARAMETERS FOR SUPPORT VECTOR MACHINE

There are ten similar works that suggested using hybrid systems to enhance classification accuracy by using few and suitable feature subsets [11], [14], [19], and [20]-[26]. In all the ten works, they optimize feature subset and SVM parameters (C and γ RBF kernel) simultaneously. SVM is then used to measure the quality of the solution for all the hybrid systems. However, what differs is what the hybrid system is based on.

[11] and [19] proposed a hybrid system which is based on GA and SVM. GA was used to select suitable feature simultaneously with optimize SVM parameter which were represented in the encoded chromosomes. [21] and [22] on the other hand, chose to use a hybrid system which is based on Particle Swarm Optimization (PSO) and SVM. In [22], discrete and continuous PSO values are integrated to simultaneously select suitable feature and optimize SVM parameter while [23] used SA to simultaneously optimize model selection and features subset selection. [23] adopt continuous version of SA which is Hide-and-Seek SA to optimize the continuous values of SVM parameters and the features are represented as discrete values. The authors for this paper did not explain how discrete values for features were handle when continuous version of SA were used to optimize SVM parameters.

[14] utilized Bees algorithm to simultaneously choose best combination of feature subset and SVM parameters values for the process of classifying faults in wood layer pieces. [20] has proposed a hybrid system which is based on ACO and SVM. The classical ACO was adopted to simultaneously select suitable feature and optimize SVM parameter. [24] proposed a hybrid system which is based on Clonal Selection Algorithm (CSA) and SVM where CSA is used to select suitable feature and optimize SVM parameter simultaneously. A hybrid system based on Cat Swarm Optimization (CSO) and SVM was proposed in [25] where CSO is used to select suitable feature and optimize SVM parameter simultaneously. Two versions of Gravitational Search Algorithm (GSA) which are real value GSA (RGSA) to optimized the real value of SVM parameters and binary (discrete) value GSA (BGSA) to select features subset have been reported in [26]. GSA is considering as swarm based meta heuristic seek approach built on gravity’s law and motion and it is derived from the Newtonian gravity. Binary classification problem was only tested while multi class classification problems were ignored.

In conclusions, all these ten works gave good results for classification accuracy using small number of selected features. [11] and [20] suggested applying their work on Support Vector Regression (SVR), because SVR accuracy counts mainly on SVR parameters and selected feature subset. [11] and [20]-[22] suggested in using other types of kernel function than RBF. [19], [21], [22], and [25] suggested applying their works on other real world problem and finally [20] suggested the use of continuous ACO to optimize the continuous value of SVM parameters.

V. PROPOSED HYBRID ALGORITHM

The proposed algorithm has adopted the IACO_{MV} to optimize features subset selection and SVM classifier parameters. An ant’s solution is used to represent a combination of features subset and the classifier parameters, C and γ, based on the Radial Basis Function (RBF) kernel of the SVM classifier. The classification accuracy of the built SVM classifier is utilized to direct the updating of solution archives and pheromone table. Based on the solution archive, the transition probability is computed to choose a solution path for an ant. In implementing the proposed scheme, this study utilizes the RBF kernel function for SVM classifier because of its capability to manage high dimensional data [27], good performance in a major cases, and it only needs to use one parameter 32], which is kernel parameter gamma (γ) [11], [27], and [28]. The overall process for hybridize IACO_{MV} and SVM (IACO_{MV}-SVM) is as depicted in Figure 3. The main steps are (1) initializing solution archive, pheromone table, and algorithm parameters, (2) solution construction for features subset and C and γ parameters (3) establishing SVM classifier model, and (4) updating solution archives and pheromone table.

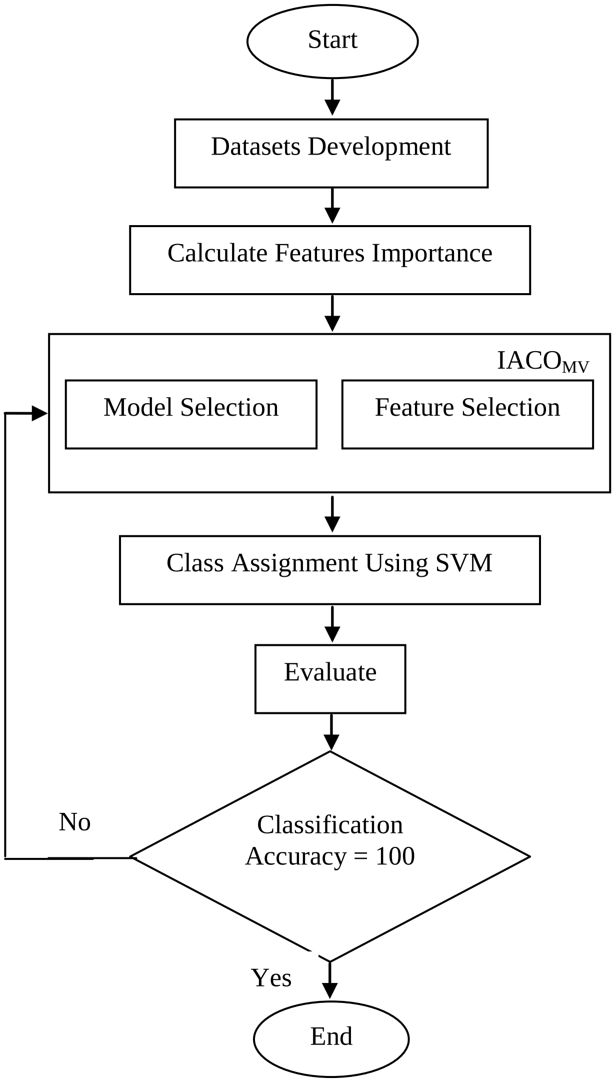
Start
Datasets Development
Calculate Features Importance
IACO_MV
Model Selection
Feature Selection
Class Assignment Using SVM
Evaluate
Classification Accuracy = 100
No
Yes
End

Figure 3: The proposed approach’s flowchart

F-score is used as a measurement to determine the importance of feature. This measurement is used to judge the favouritism capability of a feature. High value of F-score indicates favourable feature. The calculation of F-score is as follow [20]:

$$F - Score_i = \frac{\sum_{c-1}^{v}\left(\bar{x}_i^{(c)} - \bar{x}_i\right)^2}{\sum_{c-1}^{v}\left(\frac{1}{N_i^{(c)}-1}\sum_{j=1}^{N_i^{(c)}}\left(x_{i,j}^{(c)} - \bar{x}_i^{(c)}\right)^2\right)} \tag{26}$$

where $i = 1, 2, \ldots, N_f$, v is the number of categories of target variable, N_f is the number of features, $N_i^{(c)}$ is the number of samples of theith feature with categorical value *c*, *c* ∈ *{1, 2, …, v}*, $\bar{x}_{i,j}^{(c)}$ is the *j*th training sample for the *i*th feature with categorical value *c*, *j* ∈ *{1, 2, …,* $N_i^{(c)}$*}*, $\bar{x}_i$ is the ith feature, and $\bar{x}_i^{(c)}$ is the *i*th feature with categorical value *c*.

In the initialization step, each ant established a solution path for parameter *C*, parameter *γ*, and features subset. Three solution archives are needed to design the transition probabilities for first feature in the features subset, *C* and *γ*, and one pheromone table. The range for *C* and *γ* values will be sampled according to random parameter *k* which is the initial archive size of solutions archives for *C* and for *γ*, while the size for solution archive for features will be equal to number of features. The weight vector, *w* is then computed for each sample for *C* and *γ* using Eq. (21). While for features are computed using:

$$w_f = \frac{w_l}{u} + \frac{q}{\eta} \tag{27}$$

where *u* is the number of how many feature$_i$ is selected, η is the number of none selected features, and *q* in both equations is the algorithm's parameter to control diversification of search process. These values will be stored in solution archives. Once this step is completed, the sampling procedure will be constructed for *C* and *γ* in two phases. Phase one involves choosing one of the weight vectors using a probability calculated according to Eq. (22).

The second phase involves sampling selecting *w* via a random number generator that is able to generate random numbers according to a parameterized normal distribution. This initialization will construct the transition probabilities.

The probability transition that is used to move from *γ* to the first feature in features subset is as follow:

$$p_f = \frac{w_{f_i} * F-Score_{f_i}}{\sum_{i=1}^{n} w_{f_i} * F-Score_f} \tag{28}$$

In order to select other features that construct the features subset, the following probability transition is used:

$$Prob_{ij}^k = \begin{cases} \frac{(Prob_{ij})^{\alpha}(F-Score_j)^{\beta}}{\sum_{j \in I_i^k}(Prob_{ij})^{\alpha}(F-Score_j)^{\beta}} & if\ j \in I_i^k \\ 0 & otherwise \end{cases} \tag{29}$$

Like the solution archives and pheromone table, some important system parameters must be initialized as follows: the number of ants = 2, *q* = 0.1, *initial archive size* = 10, *Growth* = 5, *maximum archive size* = 15, *MaxStagIter* = 2, number of runs = 10, *α* = 1, *β* =2, *C* range ∈ [2^{-1}, 2^{12}] and *γ* ∈ [2^{-12}, 2^{2}].

The third step is related to solution construction where each ant builds its own solution. This solution will be a combination of C, γ, and features subset. In order to construct the solution, three transition probabilities with various solutions archives and pheromone table are needed. These transitions will be computed according to Eq. (22), Eq. (28), and Eq. (29).

Classifier model will be constructed in step four. Solution is generated by each ant and will be evaluated based on classification accuracy and feature weight obtained by SVM model utilizing *k*-fold Cross Validation (CV) with the training set. In *k*-fold CV, training data group is partitioned into *k* subgroups, and the holdout approach is repeated *k* times. One of the *k* subgroups is utilized as the test set and the remaining *k-1* subgroups are combined to construct the training group. The average errors along with all the k trails are calculated. CV accuracy is calculated as follows:

$$CV_{accuracy} = \frac{\sum_i test\_accuracy}{k}, i = 1, 2, \ldots, k \tag{30}$$

Test accuracy is used to evaluate the percentage of samples that are classified in the right way to determine *k*-folds and it will be compute as follows:

$$Test\ Accuracy = \frac{no.of\ correctly\ predicted\ data}{total\ testing\ data} * 100\% \tag{31}$$

The benefits of using CV are (1) each of the test groups are independent and (2) the dependent outcomes can be enhanced [20].

The final step is related to updating the solution archives and pheromone table. The solution archives' modification will be done as explained in IACO$_R$ This procedure guaranteed that just good solutions are stored in the archive and it will efficiently influence the ants in the seek process. The pheromone table will be modifying as follow:

$$T_{ij}(t+1) = pT_{ij} + \sum_{k=1}^{m} \Delta T_{ij}^k(t) \tag{32}$$

$$\Delta T_{ij}^k = \begin{cases} CVACC^k * Weight_i^k * Weight_j^k & if\ ant\ k\ use\ edge\ (i,j) \\ 0 & otherwise \end{cases} \tag{33}$$

VI. EXPERIMENTAL RESULTS

Ten datasets were used in evaluating the proposed IACO$_{MV}$-SVM algorithm. The datasets are Australian, Pima-Indian Diabetes, Heart, German, Splice, Image Segmentation, Iris, Vehicle, Sonar, and Ionosphere datasets, available from UCI Repository of Machine Learning Databases [29]. The summary of these datasets are presented in Table 1.

Table 1 Summarization of UCI's Datasets Repository

Datasets	No. of Instances	No. of Features	No. of Classes	Features' Type
Australian	690	14	2	Categorical, Integer, Real
German	1000	20	2	Categorical, Integer
Heart	270	13	2	Categorical, Real
Segmentation	2310	19	7	Real
Ionosphere	351	34	2	Integer, Real
Iris	150	4	3	Real
Diabetes	768	8	2	Integer, Real
Sonar	208	60	2	Real
Splice	3190	61	3	Categorical
Vehicle	846	18	4	Integer

All input variables were scaled during data pre-processing phase to avoid features with higher numerical ranges from dominating those in lower numerical ranges and also to reduce the computational effort. The following formula was used to linearly scale each feature to [0, 1] range.

$$\bar{x} = \frac{x-min_i}{max_i-min_i} \tag{34}$$

where x is the original value, $\bar{x}$ is the scaled value, and max_i and min_i are the maximum and minimum values of $feature_i$, respectively [20].

Each dataset is randomly re-arranged and divided into ten approximately equal size subsets, one subset as testing set and the remaining as training sets and repeated ten times. The performance of the proposed IACO$_{MV}$-SVM was compared with GA $_{\text{with feature chromosome}}$-SVM [19].

C programming language has been used to implement IACO$_{MV}$-SVM. Experiments were performed on Intel(R) Core (TM) 2 Duo CPU T5750, running at 2.00 GH$_Z$ with 4.00 GB RAM and 32-bit operating system.

Table 2 shows the optimal values for *C* and *γ* that gave the highest classification accuracy in all ten runs that have been produced by the proposed algorithm as well as *p* value which is algorithm parameter generated randomly and used to monitors the probability of utilizing just best solution in the archive as a directing solution.

Table 2 Optimal Value for *C*, *γ*, and *p*

Dataset	*C*	*γ*	*p*
Australian	416.29	0.75	0.539
German	437.68	0.74	0.506
Heart	439.33	0.76	0.516
Segmentation	820.43	1.15	0.684
Ionosphere	447.5	0.73	0.979
Iris	440.45	0.74	0.945
Diabetes	440.53	0.76	0.280
Sonar	429.79	0.76	0.843
Splice	435.50	0.73	0.801
Vehicle	311.18	0.31	0.472

Table 3 shows the average classification accuracy that produced in all the ten runs. The classification accuracy of classify pattern of the proposed IACO$_{MV}$-SVM algorithm compared with GA $_{\text{with feature chromosome}}$-SVM [19] results. The proposed algorithm classifies patterns with higher accuracy comparing with GA $_{\text{with feature chromosome}}$-SVM [19] in nine datasets while just in one dataset, the Iris dataset, the performance of the proposed algorithm IACO$_{MV}$-SVM and GA $_{\text{with feature chromosome}}$-SVM was similar. The reason for being the proposed algorithm IACO$_{MV}$-SVM is better than GA $_{\text{with feature chromosome}}$-SVM is because the proposed algorithm handles directly the continuous value of SVM parameters without the needs to discretize it. Also, this is because the proposed algorithm IACO$_{MV}$-SVM simultaneously optimizes feature subset selection and model selection for SVM.

Table 3 Classification Accuracy%

Dataset	IACO$_{MV}$-SVM	GA-SVM $_{\text{with feature chromosome}}$
Australian	96.96 ± 0.53	91.59 ± 2.14
German	97.23 ± 0.46	86.10 ± 1.97
Heart	98.01 ± 0.35	95.56 ± 2.34
Segmentation	98.96 ±0.41	98.12
Ionosphere	99.99 ± 0.02	99.43 ± 1.21
Iris	99.98 ± 0.29	100 ± 0
Diabetes	97.22 ± 0.81	83.84 ± 5.14
Sonar	99.99 ± 0.02	99.00 ± 2.11
Splice	98.65 ±0.55	90.53
Vehicle	93.92 ± 0.29	88.24 ± 1.47

Table 4 shows the average selected features subset size that was produced in all the ten runs. The proposed algorithm produced lower average number of selected features when compared with GA $_{\text{with feature chromosome}}$-SVM [19]. The biggest reduction in number of features for IACO$_{MV}$-SVM was 87.14% for Australian dataset while the smallest number of feature reduction was 75% for Diabetes and Iris datasets.

Table 4 Average Selected Feature Subset Size

Dataset	Number of Features	IACO$_{MV}$-SVM	GA-SVM $_{\text{with feature chromosome}}$
Australian	14	1.8 ± 0.4	5.2 ± 2.15
German	24	3.9 ± 0.3	10.3 ± 1.76
Heart	13	2 ± 0	6.2 ± 1.12
Segmentation	18	3 ± 0	18
Ionosphere	34	6 ± 0	13.9 ± 3.45
Iris	4	1 ± 0	1.2 ± 0.28
Diabetes	8	2 ± 0	3.7 ± 1.26
Sonar	60	12 ± 0	26.4 ± 3.20
Splice	61	9.5 ± 4.18	61
Vehicle	18	2.8 ± 0.4	9.2 ± 1.71

Table 5 shows the best features that were chosen by IACO$_{MV}$-SVM.

Table 5 Features Selection

Datasets								
Australian								
Feature#	2	3	4	5	6	7	8	
Frequencies	2	1	3	6	3	1	2	
German								
Feature#	1	2	3	4	5			
Frequencies	10	10	10	5	4			
Heart								
Feature#	1	2	3	5	6	7	8	13
Frequencies	4	5	3	1	1	2	3	1
Segmentation								
Feature#	1	2	3	5	6	7		
Frequencies	6	4	1	2	2	4		
Feature#	9	10	11	12				
Frequencies	4	4	1	2				
Ionosphere								
Feature#	1	2	3	4	5	6	7	
Frequencies	7	1	10	4	10	8	10	
Feature#	8	9	13					
Frequencies	6	3	1					
Iris								
Feature#	1	2						
Frequencies	9	1						
Diabetes								
Feature#	1	2						
Frequencies	5	5						
Sonar								
Feature#	1	2	3	4	5	6	7	
Frequencies	10	9	8	10	10	5	5	
Feature#	8	9	10	11	12	13	14	
Frequencies	8	9	10	10	10	4	2	
Feature#	20	35	43	44	45	46		
Frequencies	1	1	2	2	2	1		
Splice								
Feature#	1	2	3	4	5	6	7	8
Frequencies	7	9	9	8	6	8	7	5
Feature#	9	10	11	12	13	15	16	17
Frequencies	8	6	7	8	1	1	2	1
Vehicle								
Feature#	1	2	3	5	6	14		
Frequencies	10	5	9	1	2	1		

VII. CONCLUSION

This study has investigated an integrated IACO_{MV} and SVM technique to obtain optimal model parameters and features subset. Experimental results on ten public UCI datasets showed promising performance in terms of test accuracy and features subset size. Possible extensions can focus on the area where other kernel parameters besides RBF, application to other SVM variants and multiclass data.

ACKNOWLEDGMENT

The authors wish to thank the Ministry of Higher Education Malaysia for funding this study under Fundamental Research Grant Scheme, S/O code 12377 and RIMC, Universiti Utara Malaysia, Kedah for the administration of this study.

REFERENCES

[1] Qian Q., Chen S., & Cai W., Simultaneous Clustering and Classification Over Cluster Structure Representation, *Pattern Recognition*, Vol. 45, No. 6, 2011, pp. 2227–2236.

[2] Orkcu H. & Bal H., Comparing Performance of Back Propagation and Genetic Algorithms in the Data Classification, *Expert Systems with Applications*, Vol. 38, No. 4, 2011, pp. 3703-3709.

[3] Wang X., Liu X., Pedrycz W., Zhu X., & Hu G., Mining Axiomatic Fuzzy Association Rules for Classification Problems, *European Journal of Operational Research*, Vol. 218, No. 1, 2012, pp. 202-210.

[4] Vieira S., Sousa J, & Runkler T., Ant Colony Optimization Applied to Feature Selection in Fuzzy Classifiers, *Paper Presented at IEEE International Conference on Fuzzy System*, 2008, pp. 778-788.

[5] Maldonado S., Weber R., & Basak J., Simultaneous Feature Selection and Classification Using Kernel-Penalized Support Vector Machines, *Information Science,* Vol. 181, No. 1, 2011, pp. 115-128.

[6] Sivagaminathan R. & Ramakrishnan S., A Hybrid Approach for Feature Subset Selection Using Neural Networks and Ant Colony Optimization, *Expert Systems with Applications*, Vol. 33, No. 1, 2007, pp. 49-60.

[7] Kabir M., Shahjahan M., & Murase K., An Efficient Feature Selection Using Ant Colony Optimization Algorithm, In C. Leung & J. Chan (Eds.), *Neural Information Processing*, (pp. 242-252), Springer-Verlag: Berlin Heidelberg, 2009.

[8] Abd-Alsabour N., Feature Selection for Classification Using an Ant System Approach, Distributed, Parallel and Biologically Inspired Systems, IFIP Advances in Information and Communication Technology, Vol. 329, 2010, pp 233-241.

[9] Liu W. & Zhang D., Feature Subset Selection Based on Improved Discrete Particle Swarm and Support Vector Machine Algorithm, *Paper presented at the IEEE International Conference of the Information Engineering and Computer Science,* 2009, pp.1-4.

[10] Vapnik V. & Vashist A., A New Learning Paradigm: Learning Using Privileged Information, *Neural Networks: the Official Journal of the International Neural Network Society*, Vol. 22, No.5-6, 2009, pp.544–557.

[11] Huang C. & Wang C., A GA-Based Feature Selection and Parameters Optimization for Support Vector Machines, *Expert System with Applications*, Vol.31, No.2, 2006, pp.231-240.

[12] Liao, T., Dorigo, M., & Stutzle, T., Improved Ant Colony Optimization Algorithms for Continuous and Mixed Discrete-Continuous Optimization Problems. Retrieved from http:// http://www.swarmbots.org/~mdorigo/HomePageDorigo/thesis/dea/LiaoMAS.pdf, 2011.

[13] Alwan, H.B & Ku-Mahamud, K.R., Incremental Continuous Ant Colony Optimization Technique for Support Vector Machine Model Selection Problem, *Paper Presented at Proceedings the 17th WSEAS International Conference on Applied Mathematics (AMATH '12),* 2012, pp. 165-170. Montreux, Switzerland.

[14] Pham D., Muhamad Z., Mahmuddin M., Ghanbarzadeh A., Koc E., & Otri S, Using the Bees Algorithm to Optimise a Support Vector Machine for Wood Defect Classification, *Paper presented at Innovative Production Machines and Systems Virtual Conference,* 2007.

[15] Hsu C., Chang C., & Lin C. (2010). A practical guide to support vector classification. Retrieved December 25, 2010, from http://www.csie.ntu.edu.tw.

[16] Dorigo M & Socha K., An Introduction to Ant Colony Optimization, In T. F. Gonzalez (Ed.), Approximation Algorithms and Metaheuristics, Vol. 2006, No. 010, 2006, pp. 1-19.

[17] Blum C., Ant Colony Optimization: Introduction and Recent Trends, *Physics of Life Reviews*, Vol. 2, No. 4, 2005, pp. 353-373.

[18] Socha K., Ant Colony Optimization for Continuous and Mixed-Variables Domain. Doctoral dissertation, Universite' Libre de Bruxelles, 2008, Retrieved from: iridia.ulb.ac.be/~mdorigo/HomePageDorigo/thesis/SochaPhD.pdf.

[19] Zhao M., Fu C., Ji L., Tang K., & Zhou M., Feature Selection and Parameter Optimization for Support Vector Machines: A New Approach Based on Genetic Algorithm with Feature Chromosomes, E*xpert System with Applications*, Vol. 38, No. 5, 2011, pp. 5197-5204.

[20] Huang C., ACO-Based Hybrid Classification System with Feature Subset Selection and Model Parameters Optimization, *Neurocomputing*, Vol.73, No.1-3, 2009, pp.438-448.

[21] Lin S., Ying K., Chen S., & Lee Z., Particle Swarm Optimization for Parameter Determination and Feature Selection of Support Vector Machines, *Expert System with Applications*, Vol. 35, No. 4, 2008, pp. 1817-1824.

[22] Huang C. & Dun J., A Distributed PSO-SVM Hybrid System with Feature Selection and Parameter Optimization, *Applied Soft Computing*, Vol. 8, No. 4, 2008, pp. 1381-1391.

[23] Lin S, Lee Z., Chen S., Tseng T., Parameter Determination of Support Vector Machine and Feature Selection Using Simulated Annealing Approach, *Applied Soft Computing*, Vol. 8, No. 4, 2008, pp. 1505-1512.

[24] *Ding S. & Li S.,* Clonal Selection Algorithm for Feature Selection and Parameters Optimization of Support Vector Machines, *Paper presented at 2nd International Symposium IEEE on knowledge acquisition and modeling*, 2009, pp. 17-20.

[25] Lin K. & Chien H., CSO-Based Feature Selection and Parameter Optimization for Support Vector Machine, *Paper presented at IEEE Pervasive Computing (JCPC)*, 2009, pp. 783-788, Tamsui, Taipei.

[26] Sarafrazi S. & Nezamabadi-Pour H., Facing the Classification of Binary Problems with a GSA-SVM Hybrid System, *Mathematical and Computer Modeling*, in press.

[27] Moustakidis S. & Theocharis J., SVM-FuzCoC: A Novel SVM-Based Feature Selection Method Using a Fuzzy Complementary Criterion, *Pattern Recognition*, Vol. 43, No. 11, 2010, pp. 3712-3729.

[28] Alwan, H.B. & Ku-Mahamud, K.R., Incremental Mixed Variable Ant Colony Optimization for Feature Subset Selection and Model Selection. *Paper Presented at Proceedings the 12th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED '13)*, 2012, pp. 131-136. Cambridge, UK.

[29] *UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine, CA, <http://www.ics.uci.edu/mlearn/MLRepository>, 2012.*

[30] M. Ku & A. Mustafa, New heuristic function in ant colony system for job scheduling in grid computing. *Paper Presented at Proceedings the 17th WSEAS International Conference on Applied Mathematics (AMATH '12)*, 2012, pp. 47-52. Montreux, Switzerland.

[31] Barbu T., SVM-based human cell detection technique using histograms of oriented gradients. *Paper Presented at Proceedings the 17th WSEAS International Conference on Applied Mathematics (AMATH '12),* 2012, pp. 156-160. Montreux, Switzerland.

[32] Zhang H., Xiang M., Ma C., Huang Q., Li W., Xie W., Wei Y. & Yang S., Three-Class Classification Models of LogS and LogP Derived by Using GA-CG-SVM Approach, *Molecular Diversity*, Vol. 13, No. 2, 2009, pp. 261-268.

Hiba Basim Alwan holds a Bachelor in Computer Science from Al-Mansour University College, Baghdad-Iraq in 2004 and a Masters degree in Computer Science from University of Technology, Baghdad-Iraq, in 2006. As an academic, her research interests include Ant Colony Optimization, Genetic Algorithm, Machine Learning, Support Vector Machine, and Classification. She is currently a PhD. student in the School of Computing at the College of Arts and Science, UUM.

Prof. Dr. Ku Ruhana Ku-Mahamud holds a Bachelor in Mathematical Science and a Masters degree in Computing, both from Bradford University, United Kingdom in 1983 and 1986 respectively. Her PhD in Computer Science was obtained from University Pertanian Malaysia in 1994. As an academic, her research interests include computer system performance modeling and swarm intelligence. She is currently a professor in the School of Computing at the College of Arts and Science, UUM.