Solving SVM model selection problem using ACO_R and $IACO_R$

HIBA BASIM ALWAN[1], KU RUHANA KU-MAHAMUD[2]
[1 and 2] School of Computing, College of Art and Sciences
University Utara Malaysia
06010 Sintok, Kedah
MALAYSIA
hiba81basim@yahoo.com[1] ruhana@uum.edu.my[2]

Abstract: -Ant Colony Optimization (ACO) has been used to solve Support Vector Machine (SVM) model selection problem. ACO originally deals with discrete optimization problem. In applying ACO for optimizing SVM parameters which are continuous variables, there is a need to discretize the continuously value into discrete values. This discretize process would result in loss of some information and hence affect the classification accuracy. In order to enhance SVM performance and solving the discretization problem, this study proposes two algorithms to optimize SVM parameters using Continuous ACO (ACO_R) and Incremental Continuous Ant Colony Optimization ($IACO_R$) without the need to discretize continuous value for SVM parameters. Eight datasets from UCI were used to evaluate the credibility of the proposed integrated algorithm in terms of classification accuracy and size of features subset. Promising results were obtained when compared to grid search technique, $GA_{with\ feature\ chromosome}$-SVM, PSO-SVM, and GA-SVM. Results have also shown that $IACO_R$-SVM is better than ACO_R-SVM in terms of classification accuracy.

Key-Words: -Support Vector Machine, Continuous Ant Colony Optimization, Incremental Continuous Ant Colony Optimization, Model Selection.

1 Introduction

Data classification process consists of: training and testing and this can be undertaken in a two stage procedure. First, the training data are used to build the classifier (model for classification) and subsequently, the classifier will be tested using the test data. The execution of the classification procedure is determined by the precision of the distinguishing function for the particular problem to which it is applied. A distinguishing function is improved to minimize the misclassification percentage, using the few present examples of input and output vector pairs, which are known as the training data group. This distinguishing function is then utilized to classify new examples into pre-defined categories and to test the precision of the classification [1].Pattern classification is an important area in machine learning and artificial intelligence. It 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 [2].

Support Vector Machine (SVM) has been introduced by Vapnik and it has obtained wide recognition due to many interesting features and practical implementations. The SVM was improved based on the concept of structural risk minimization

which means, it decreases the trade-off between the experimental mistake and the difficulty that arises from classification of estimating functions to prevent over fitting. The concept of structural risk minimization plans the data into high dimensional domains via kernel functions by using kernel tricks [3].Polynomial, Radial Base Function (RBF) and sigmoid kernel function are three examples of kernel functions that can be applied in SVM. RBF is the more popular kernel function because of its capability to manage high dimensional data [4], good performance in major cases [5] and it only needs one parameter which is the kernel parameter gamma (γ) [6].However, the disadvantage of RBF is that it will fail if it deals with large numbers of features [7].Two problems in SVM classifier that influence the classification accuracy are: tuning SVM parameters, and selecting an optimal feature subset to be given to the SVM classifier. These problems affect each other and in turn will affect on SVM's performance [6].There is no regular methodology that accepts advance approximation of optimal values for SVM parameters. In present classification work, obtaining good values for these parameters is not easy. It requires either an exhaustive search through the space of hyper variables or an optimization approach that searches simply a bounded sub-group of the potential values.

Currently, almost all SVM research chooses these variables experimentally via searching a bounded number of values and preserving those that supply the lowest amount of mistakes. This approach needs a grid search through the area of variable values and requires identifying the range of executable solution and best sampling step. This is a difficult task because best sampling steps change from kernel to kernel and grid ranges may not be simple to identify without advanced knowledge of the problem. Furthermore, when a hyper parameter exceeds two of the manual prototypes chosen, it may become intractable [8].

ACO algorithms were applied to tune SVM parameters. These algorithms work through repetitive creation procedures where each procedure directs a dependent heuristic by intelligently mixing various ideas for exploring and exploiting the seek space. The learning fashions are utilized to construct information to efficiently obtain near optimal solutions. Solutions that are built using ACO seek to find the shortest way to the origin of food via pheromones [9].

ACO algorithms deal with discrete and continuous variables. However, an ACO deal with continuous variable is considered as a modern research field. ACO_R uses probability density function instead of discrete probability distribution to determine the direction that an ant should follow [10]. The main disadvantages of ACO_R are the stagnation problem and the application gap with the-state-of-art continuous solvers. To overcome these disadvantages, $IACO_R$ Algorithm is introduced [11]. In this study, ACO_R and $IACO_R$ are used to solve SVM model selection problem. The two proposed algorithms presented here are different from previous proposed algorithm presented in [24] in which these two proposed algorithms deals with tuning SVM parameters. The work in [24] simultaneously tunes SVM parameters and optimizes feature subset selections. The rest of the paper is organized as follows. Section 2 presents a brief introduction to SVMwhile Section 3 presents the concept of ACO, ACO_R and $IACO_R$. Section 4 reviews several literatures on tuning SVM parameters and Section 5 describes the proposed algorithms. Section 6 presents the findings and concluding remarks and future works are presented in Section 7.

2 Support Vector Machine

For binary class classification problem, given M training examples where each example is represented through a tuple (x_i, y_i) where $i = 1, ...,$

M, $x_i \in R^M$ corresponds to the feature group for the ith example, and $y_i \in \{+1, -1\}$ denoted the class label, SVM needed to solve the following equation [12]:

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

Subject to:

$$y_i(w^T \phi(x_i) + b) + \xi_i - 1, \xi_i \geq 0, i = 1, ..., M$$

the mapping function that maps the training instances from input space to higher dimensional feature space. Often Eq. (1) is solved through solving the following dual problem [13]:

$$min_\alpha F(\alpha) = \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \qquad (2)$$

Subject to

$$0 \leq \alpha_i \leq C, i = 1, ..., M \text{and} y^T \alpha = 0$$

where $\{\alpha_i\}_{i=1}^{i}$ is Lagrange multipliers, e is the vector of all ones and Q is an M by M positive semi-definite matrix. The $(i, j)^{th}$ element of Q is given through:

$$Q_{i,j} = y_i y_j K(x_i, x_j) \qquad (3)$$
$$K(x_i, x_j) \equiv \phi^T(x_i) \phi(x_j) \qquad (4)$$

where $K(x_i, x_j)$ is the kernel function, Radial Basis Function (RBF) is a common one and as follow:

$$K(x_i, x) = exp^{(-\gamma \|x_i - x\|^2)} \qquad (5)$$
$$w = \sum_{i=1}^{M} \alpha_i y_i \phi(x_i) \qquad (6)$$

where w is the weight vector. The classification decision function is:

$$sgn(w^T \phi(x) + b) = sgn(\sum_{i=1}^{M} \alpha_i y_i K(x_i, x) + b) \qquad (7)$$

For multi-class classification problem, there are two major techniques, which are One-Against-One (OAO) and One-Against-All (OAA). In the OAO technique, thenumbers of binary SVM classifier are built using the following equation [12]:

$$C_2^v = \frac{v(v-1)}{2} \qquad (8)$$

SVM binary classifiers are built and each classifier is trained utilizing two classes. A new x is classified into the majority class that is voted on through all of the decision functions. While in OAA technique, v binary SVM decision functions are built for a v-class problem. The jth $(j = 1, 2,..., v)$ decision function is trained through 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 x is classified into the class that has the largest decision function.

3 Ant Colony Optimization

ACO is a metaheuristic approach for hard discrete optimization problems that was initially introduced in the beginning of 1990s. ACO is based on the behavior 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. In order to solve optimization problem, ACO will repeat the following two steps: 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 [14].

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 interesting ACO for continuous variables and mixed variables is conducted by [10] which is called continuous ACO (ACO_R) and mixed-variable ACO (ACO_{MV}). ACO_R is later modified [11] and two new algorithms called Incremental ACO_R ($IACO_R$) and Incremental ACO_R with Local Search ($IACO_R$-LS) were 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. Adopting either $IACO_R$ or $IACO_R$-LS to modifying mixed-variable Ant Colony Optimization (ACO_{MV}) introduced in [10] to optimize mixed variable problem was suggested by [11].

3.1 Continuous ant colony optimization

Continuous ant colony optimization (ACO_R) introduced by [10] and it used Probability Density Function (PDF) instead of Discrete Probability Distribution to determine the direction that an ant should follow; Gaussian function, an example of PDF is one of the most popular as it uses 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 nonsystematic 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. These values are then used to dynamically create PDF. The solution will be established by each ant. For each ant toestablish a solution path, a solution archive is needed to design the transition probabilities. The weight vector, w is computed for each solution stored in solution archive as follows:

$$w_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \qquad (9)$$

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 and as follows:

$$p_l = \frac{w_l}{\sum_{r=1}^{k} w_r} \qquad (10)$$

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 initializing constructs the transition probabilities for ants. An outline of ACO_R is given in Figure 1 [11]:

Figure 1: ACO_R algorithm

Input: k, m, D, q, ξ, and termination criterion
Output: The best solution found
Initialize and evaluate k solutions
    T = Sort (S_1, …,S_k)
    While Termination criterion is not satisfied do
for$l$ = 1 to m do
for$i$ = 1 to D do
        Select Gaussian g_j^i according to weights
        Sample Gaussian g_j^i with parameters μ_j^i, σ_j^i
end for
store and evaluate newly generated solution
end for
    T = Best (Sort (S_1, …,S_{k+m}), k)
End while

3.2 Incremental continuous ant colony

algorithm

Incremental continuous ant colony optimization (IACO$_R$) was proposed by [11] to enhanced ACO$_R$. It starts with a small archive defined by the *InitArhiveSize*.IACO$_R$ starts with randomly initializing the solution archive. This solution archive will be filled with solutions which were generated randomly. IACO$_R$ also characterizes a strategy different from the oneutilized in ACO$_R$ for choosing the solution that directs the creation of new solutions. The new procedure builds on parameter *p ∈ [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.

If the newly created solution is better than the directing solution, it replaced it in the archive.This replacement mechanism is different 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 every growth iteration until a maximum archive size, defined by *MaxArchiveSize*, is reached.A parameter *Growth* monitors the percentage at which the archive grows. Fast growth percentage support seeks diversification while slow growth supports intensification. Each time a new solution is appended,it is initialized utilizing information frombest solution in the archive. First, a new solution S_{new} is created fully in an arbitrary way, and then it is moved in direction of the best solution in the archive S_{best} utilizing the following formula:

$$\overline{S}_{new} = S_{new} + rand(0, 1)(S_{best} - S_{new}) \quad (11)$$

where*rand(0, 1)* is an arbitrary number within the range [0, 1).

IACO$_R$ involve an algorithm-level diversification strategy for fighting stagnation. The strategy includes in 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

---

Input: *p*, *InitArhiveSize*, *Growth*, *MaxArchiveSize*, *MaxStagIter*, no. of ants, and Termination criterion
Output: Optimal Value for *C* and *γ*
k= InitArhiveSize
initialize*k* solutions and evaluate it
while Termination criterion not satisfied do
    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
    if current iterations are multiple of *Growth*&*k<MaxArchiveSize* then
        Initialize new solution using Eq. (4.4)
        Add new solution to the archive
        k + +
    end
    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

outline of IACO$_R$ is given in Figure 2.

4 Tuning support vector machine parameter

Using GA to optimize SVM variables was proposed by [15]. The regularization parameter C and kernel parameters are dynamically optimized through GA. In their work, they used unconnected time strings for each worked trading interval instead of utilizing single time strings to model each day's price profile. From their experiments they concluded that their model supplies better predicting with sensible levels of accuracy and stability. A grid-based ACO technique was introduced by [16]to select variables C and RBF kernel σ automatically for SVM instead of choosing variables unsystematically through human skill to minimize generalization mistakes and generalization execution which may be enhanced concurrently. Their work provides high accuracy and less calculation time compared with other methods such as grid algorithm and cross validation approach. RBF kernel is utilized to enhance the accuracy of SVM. However, one dataset is used to evaluate the performance of the proposed technique. ACO was also used by [17]to optimize both SVM parameters, C and σ kernel function parameters in continuous fields. Both parameters C and σ are divided into a number of sub-intervals. In each sub-interval, one point is chosen unsystematically to be the location of artificial ants. Before starting each loop, advance knowledge and heuristic information are modified. In every loop, the transition probability of each ant is predetermined. The ant will move to the next interval if the state transition rule is met, otherwise, the ant will search for optimal variables within local intervals. Their results showed a very promising hybrid SVM model for forecasting share price in terms of accuracy and generalization ability. Utilizing PSO and grid search in a dynamic environment to optimize SVM parameters was introduced by [18]. The authors examined their approach on fourteen datasets and compared their work with various approaches. The results show that their approach outperforms the classical methods in terms of model complexity and calculation time. Bare Bones Differential Evolution (BBDE) to optimize SVM parameters was proposed by [19]. It deletes the monitor variables of PSO and substitutes the static Differential Evolution (DE) DE monitor variables with dynamically alternating variables to generate a general parameter-free, self-adaptive, optimization algorithm. Four University California, Irvin (UCI) datasets were used to test the performance of the proposed method and results were compared with grid algorithm. Results showed that BBDE-SVM and DE-SVM take shorter time to compute and produced higher classification accuracy. This implies that the parameters chosen by BBDE and DE are better than grid search. A hybrid method based on support vector machine and simulated annealing (SVM-SA) [20] has been used to diagnose hepatitis. SA was used to find the optimal value for SVM parameters. Promising results were obtained and the authors suggested employing some feature selection techniques and other learning approach to maximize the precision of their approach. A hybridized algorithm between GA and SVM to tune its parameters which are the regularization parameter C and RBF kernel function parameter has been proposed by [21]. These two parameters were encoded as a real value chromosomes and the fitness value for each chromosome were computed in terms of chromosome's classification accuracy. The authors show that the hybridization between GA and SVM can improve the classification accuracy and convergence speed. The authors showed that hybridization between GA and SVM can improve the classification accuracy and convergence speed. A study by [9] was on the use of ACO and grid search to solve SVM model selection problem. The authors presented a novel ACO pheromone model and divided the ranges of RBF and C parameters into a number of grids and let the ants select the best combination of RBF and C parameters. The proposed approach was feasible and efficient to optimize the SVM parameter and produce hopeful results in terms of classification accuracy and calculation time.

5 Proposed algorithm

This study constructs ACO$_R$ and IACO$_R$ to optimize SVM classifier parameters. An ant's solution is used to represent a combination of 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 archive. Based on the solution archive, the transition probability is computed to choose a solution path for an ant. The outline of the proposed ACO$_R$-SVM algorithm is given in Figure 3 while the outline of the proposed IACO$_R$-SVM algorithm is given in Figure 4:

Input: Size of solution archive (k), no. of ants (m), q range of C, range of γ, and termination criterion
Output: Optimal value for SVM parameters (C and γ) and classification accuracy
Begin
       Initialize k solutions
       call SVM algorithm to evaluate k solutions
       $T = $ Sort ($S_1, \ldots, S_k$)
       while termination criterion is not satisfied do
             for $i = 1$ to m do
                    select S according to its weight
                    sample selected S
                    store newly generated solutions
                    call SVM algorithm to evaluate newly generated solutions
             end
             $T = $ Best (Sort $S_1, \ldots S_k + m$), k)
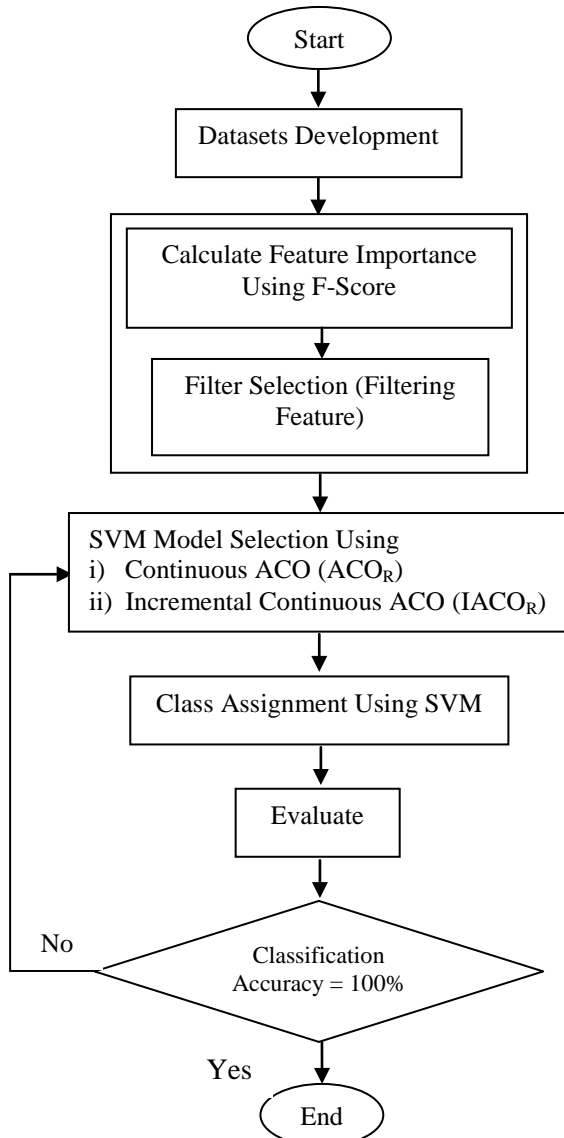   end
End

Figure 3: Proposed ACO_R–SVM algorithm

Input: p, *InitArhiveSize*, *Growth*, *MaxArchiveSize*, *MaxStagIter*, no. of ants, and Termination criterion
Output: Optimal Value for C and γ
$k = $ *InitArhiveSize*
initialize k solutions
call SVM algorithm to evaluate k solutions
while Termination criterion not satisfied do
       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 generated solutions
                    if Newly generated solution is better than S_j then
                          Substitute newly generated solution for S_j
                    end
             end
       end
       if current iterations are multiple of *Growth&k<MaxArchiveSize* then
             Initialize new solution using Eq. (4.4)
             Add new solution to the archive
             $k + +$
       end
       if # (number) of iterations without improving $S_{best} = $ *MaxStagIter* then
             Re-initialize T (solution archive) but keeping S_{best}
       end
end

Figure 4: Proposed $IACO_R$–SVM algorithm

The overall process to integrate ACO_R and SVM (ACO_R-SVM) and $IACO_R$ and SVM ($IACO_R$-SVM) is as depicted in Figure 5. The main steps are (1) selecting feature subset (2) initializing solution archive and algorithm parameters, (3) solution

```
            ( Start )
               |
               v
    [ Datasets Development ]
               |
               v
  [ Calculate Feature Importance
         Using F-Score ]
               |
               v
   [ Filter Selection (Filtering
            Feature) ]
               |
               v
  [ SVM Model Selection Using
    i)  Continuous ACO (ACO_R)
    ii) Incremental Continuous ACO (IACO_R) ]  <--+
               |                                  |
               v                                  |
  [ Class Assignment Using SVM ]                  |
               |                                  |
               v                                  | No
         [ Evaluate ]                             |
               |                                  |
               v                                  |
       < Classification  > ---------------------- +
       < Accuracy = 100% >
               |
              Yes
               |
               v
            ( End )
```

Figure 5: Integrated ACO_R/$IACO_R$ and SVM algorithm

In the features subset selection step, F-score is used as a measurement to determine the importance of the features. This measurement is used to judge the favoritism capability of a feature. High value of F-score indicates the most favorable feature. The calculation of F-score is as follows [12]:

$$F - Score_i = \frac{\sum_{c-1}^{v}\left(\overline{x}_i^{(c)} - \overline{x}_i\right)^2}{\sum_{c-1}^{v}\left(\frac{1}{N_i^{(c)}-1}\sum_{j=1}^{N_i^{(c)}}\left(x_{i,j}^{(c)} - \overline{x}_i^{(c)}\right)^2\right)}, i = 1, 2, ..., N_f \quad (12)$$

where 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 the ith feature with categorical value c, $c \in \{1, 2, ..., v\}$, $\overline{x}_{i,j}^{(c)}$ is the jth training sample for the ith feature with categorical value c, $j \in \{1, 2, ..., N_i^{(c)}\}$, $\overline{x}_i$ is the ith feature, and $\overline{x}_i^{(c)}$ is the ith feature with categorical value c.

After computing the F-score for each feature in the dataset, the average F-score is computed and is considered as the threshold for choosing features in the feature subset. Features with F-scores equal to or greater than the threshold are chosen and put in the feature subset and this subset is presented to the SVM.

In the initialization step, for each ant establishing a solution path for parameter C and parameter γ, two solution archives are needed to design the transition probabilities for C and for γ. The range value for C and γ are sampling according to random parameter k which is the size of solutions archives. The weight vector, w is then computed for each sample for C and γ according to Eq. (9).

Once this step is completed, the sampling procedure is made through two phases. Phase one involves choosing one of the weight vectors according to Eq. (10), while 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 initializing constructs the transition probabilities. Like the solution archives, 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, C range is $\in [2^{-1}, 2^{12}]$ and $\gamma \in [2^{-12}, 2^2]$.

The third step relates to solution construction where each ant builds its own solution. This solution is a combination of C and γ. In order to construct the solution, two transition probabilities with various solutions archives are needed. These transitions are computed according to Eq. (9) and Eq. (10).

A classifier model is constructed in step four. Solution is generated by each ant and is evaluated based on the classification accuracy obtained by the SVM model utilizing k-fold Cross Validation (CV) with the training set. In k-fold CV, the training data group is portioned into k subgroups, and the holdout approach is repeated k times. One of the k sub-groups is utilized as the test set and the remaining k-

construction for C and γ, (4) establishing SVM classifier model, and (5) updating solution archive.

1 sub-groups are combined to construct the training group. The average mistakes along with all the *k* trails are calculated. CV accuracy is calculated as follows:

$$CV_{accuracy} = \frac{\sum_i test\_accuracy}{k}, i = 1, 2, \ldots, k \quad (13)$$

Test_accuracy evaluates the percentage of samples that are classified in the correct way to determine *k*-folds and is computed as follows:

$$Test\ Accuracy = \frac{no.of\ correctly\ predicted\ data}{total\ testing\ data} * 100\% \quad (14)$$

The benefits of using CV are (1) each of the test groups is independent and (2) the dependent outcomes can be enhanced [12]

The final step is related to updating solution archives. This modification is completed by appending the newly generated group solutions that gave the best classification accuracy to solution archive and then deleting the exact number of worst solutions. For ACO$_R$-SVM this ensures the size of solution archive does not change while for IACO$_R$-SVM the solution archive continues to grow during the running until it reaches a specific size defined by *maximum archive size*. This procedure guarantees that only good solutions are stored in the archive, and it will efficiently influence the ants in the seek process.

For each iteration, ACO$_R$/IACO$_R$ generates SVM parameters' values and introduces it to SVM and SVM uses these values to classify patterns. The proposed algorithms stops if the classification accuracy or maximum number of iteration satisfies user specification; otherwise, ACO$_R$/IACO$_R$ searches for other optimal values for SVM parameters to work with.

6 Findings

Eight datasets were used in evaluating the proposed ACO$_R$-SVM and IACO$_R$-SVM algorithms. The datasets are Australian, Pima-Indian Diabetes, Heart, Ionosphere, German, Sonar, Iris, and Vehicle datasets available from UCI Repository of Machine Learning Databases [22]. The summary of these datasets is presented in Table 1.

Table 1. Summarization of UCI's datasets repository

Dataset	No. of instances	No. of features	No. of classes	Features' Types
Australian	690	14	2	Categorical, Integer, Real
Diabetes	760	8	2	Integer, Real
Heart	270	13	2	Categorical, Real
Ionosphere	351	34	2	Integer, Real
German	1000	24	2	Categorical, Integer
Sonar	208	60	2	Real
Iris	150	4	3	Real
Vehicle	846	18	4	Integer

These datasets have been utilized as benchmarks to compare the execution of different classification approaches in the papers frequently.

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

$$\overline{x} = \frac{x - min_i}{max_i - min_i} \quad (15)$$

where *x* is the original value, $\overline{x}$ is the scaled value, and max_i and min_i are the maximum and minimum values of feature *i*, respectively [12].

Each dataset was randomly re-arranged and divided into ten approximately equal sized subsets, one subset is a testing set and the remaining are training sets and repeated ten times to enable each fold of data to take a turn as the testing dataset.

The classification accuracy percent of the experiment was calculated by summing the individual accuracies percent for each run of testing and then divided the overall by 10. The performance of the proposed ACO$_R$-SVM and IACO$_R$-SVM was compared against other as well as were compared with the grid search approach which was considered as the basic approach to optimize SVM parameters without the ability to select features subset also the proposed approaches were compared with GA$_{with\ feature\ chromosome}$-SVM [23] and PSO-SVM and GA-SVM [13]. The work presented in [13] only focused on tuning SVM paratemters while [23] focused on simultaneously tuning SVM parameters and optimizes feature subset selection. In this paper, the results of proposed algorithms are compared with [13]. C programming language was used to implement ACO$_R$-SVM and IACO$_R$-SVM. Experiments were performed on an 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 classification accuracy produced by the proposed algorithms compared with four (4) other approaches. It can be seen that the proposed

$IACO_R$-SVM algorithm achieves highest classification accuracy in all datasets except the Iris dataset where ACO_R-SVM recorded the best result. The proposed algorithms was able to tune the SVM parameters without the need to discretize the continuous value, thus eliminate any error or

missing information which leads to better accuracy. It also shows that the proposed algorithms can handle categorical, real and integer values.

Table 2. Classification accuracy

Dataset	ACO_R-SVM	$IACO_R$-SVM	Grid Search	GA$_{with feature chromosome}$-SVM	PSO-SVM	GA-SVM
Australian	91.99	94.62	84.74	86.81	88.09	88.09
Diabetes	88.00	91.25	76.58	81.97	80.19	82.98
Heart	94.87	96.28	88.15	91.11	-	-
Ionosphere	97.76	98.90	94.29	98.57	97.50	96.61
German	88.72	91.56	78.90	80.80	79.00	84.24
Sonar	98.36	98.92	90.50	95.00	88.32	95.22
Iris	**99.87**	99.86	94.09	96.00	98.00	97.56
Vehicle	93.00	93.20	83.94	84.74	88.71	85.87

Table 3 shows the number of selected feature produced by the proposed algorithms and three (3) other approaches. Smaller number of features is produced in all datasets except the Iris dataset by the proposed ACO_R-SVM algorithm and $IACO_R$-SVM algorithm as compared to other approaches. For the Iris dataset, GA $_{with feature chromosome}$-SVM approach

achieves the smallest feature subset size. Both of the proposed algorithms produce similar feature subset sizebecause both of the proposed algorithms use the same technique to select features. The biggest reduction in number of features generated by ACO_R-SVM is 77.86% and for $IACO_R$-SVM is 76.43% for the Australian dataset.

Table 3. Number of selected features

Datasets	Number of features	ACO_R-SVM	$IACO_R$-SVM	GA$_{with feature chromosome}$-SVM	PSO-SVM	GA-SVM
Australian	14	3.2	3.3	6.7	15	14
Diabetes	8	2.6	2.6	5.1	8	8
Heart	13	6.2	6.0	7.0	-	-
Ionosphere	34	11	10.6	15.4	34	34
German	24	6.2	6.4	11.8	30	24
Sonar	60	21.5	21.7	28.7	60	60
Iris	4	2	2	1.8	4	4
Vehicle	18	8.9	9.3	10.3	18	18

7 Conclusions and future works

Continuous Ant Colony Optimization (ACO_R) and Incremental continuous Ant Colony Optimization ($IACO_R$) as an extension of the Ant Colony Optimization (ACO) algorithm offer the opportunity to deal with continuous optimization problems. This study had proposed enhanced pattern classification algorithms based on ACO_R and $IACO_R$ with SVM. ACO_R and $IACO_R$ act as optimization algorithms in order to tune SVM parameters. The proposed algorithms outperformed GA$_{with feature chromosome}$-SVM, Grid search, PSO-SVM and GA-SVM when comparisons were performed on classification accuracy. This was possible because selection of feature subset is made through filter approach using a threshold technique. Average Fisher (F)-score has

been used as the threshold value to select feature subset.

The differences between the two proposed algorithmsare (i) the size of solution archive in $IACO_R$-SVM grows over time until it reaches a predefined maximum size while in ACO_R-SVM the size of solution archive will be fixed, (ii) the mechanism for selecting a solution that directs the creation of new solutions in $IACO_R$-SVM is different from the mechanism used in ACO_R-SVM, and (iii) a diversification technique was utilized in $IACO_R$-SVM in solving stagnation. This technique includes restarting the algorithm and initializing the new initial solution archive with the best-so-far solution. The restarting condition is the number of successive iterations through a relation solution enhancement lower than a certain threshold.

Possible extensions can focus on the area where $ACO_R/IACO_R$-SVM can simultaneously optimize both SVM parameters and features subset using mixed-variable ACO (ACO_{R-MV}/I ACO_{R-MV}).Other future direction is to apply the proposed algorithms on Support Vector Regression (SVR), because SVR has similar problems as SVM because it is extended from Vector Classification (SVC). This task requires minimal changes to the proposed algorithms by defining other benchmarks datasets. Regression is a data mining task in predicting the value of the target (numerical variable) by building a model based on one or more predictors (numerical and categorical variables).

Another future direction is to use the proposed algorithms in solving dynamic problems. In dynamic optimization, the search space changes with time. This requires a modification of the proposed algorithms to consider the dynamism of search space. Other variants of SVM such as least square SVM can also be used to solve classification problems. It would be accomplished by modifying the SVM's mathematical equation. Future work could also focus on the area where other kernel parameters besides Radial Basis Function can be used or to design a mechanism for using many kernel functions and selecting the most successful kernel function that gives the best classification accuracy.

Acknowledgments

References

[1] X.Cheng, J.Xu, J. Pei& J.Liu, Hierarchical Distributed Data Classification in Wireless Sensor Networks,*Computer Communications,*Vol. 33, No. 12, 2010, pp. 1404-1413.

[2] M.Khashei, A. Zeinal& M. Bijari, A Novel Hybrid Classification Model of Artificial Neural Networks and Multiple Linear Regression Models, *Expert Systems with Applications,* Vol. 9, No. 3, 2012, pp. 2606-2620.

[3] V. Vapnik, An Overview of Statistical Learning Theory,*Transactions on Neural Networks,* Vol.

10, No. 5, 1999, pp. 988-999.

[4] X. Li& Z. Yang, Least Squares Density Induced Margin Support Vector Machinefor Pattern Classification,*International Journal of Digital Content Technology and its Applications*Vol. 6, No. 15, 2012, pp. 297-305.

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

[6] H. Zhang, M. Xiang, C. Ma, Q. Huang, W. Li, W. Xie, Y. Wei, S. Yang,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.

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

[8] V. Vapnik&A. Vashist, ANew 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.

[9] N. Ayat, M. Cheriet&C. Suen, Automatic Model Selection for the Optimization of SVM Kernels,*Pattern Recognition,* Vol. 38, No. 10, 2005, pp. 1733-1745.

[10] X. Zhang, X. Chen& Z. He,An ACO-Based Algorithm for Parameter Optimization of Support Vector Machines,*Expert Systems with Applications,* Vol. 37, No. 9, 2010, pp. 6618-6628.

[11] K. Socha,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, 2008.*

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

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

[14] S. Lin, K. Ying, S. Chen& Z. Lee, Particle Swarm Optimizationfor Parameter Determination and Feature Selection of

Support Vector Machines,*Expert System with Applications,* Vol. 35, 2008, pp. 1817-1824.

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

[16] L. Saini, S. Aggarwal& A. Kumar, Parameter Optimization Using Genetic Algorithm for Support Vector Machine-Based Price-Forecasting Model in National Electricity Market,*IET Generation, Transmission & Distribution*, Vol. 4, No. 1, 2008, pp. 36-49.

[17] X. Zhang, X. Chen, Z. Zhang,&Z. He, A Grid–Based ACO Algorithm for Parameters Optimization in Support Vector Machines, *Proceedingsof the IEEE International conference on Granular Computing,* 26-28 August 2008,Hangzhou, 805-808.

[18] X. Fang& T. Bai, Share Price Prediction Using Wavelet Transform and Ant Colony Algorithm for Parameters Optimization in SVM, *Proceedings of the IEEE International Congress on Intelligent Systems,* 19-21 May 2009, Xiamen, 288-292.

[19] M. Kapp, R. Sabourin& P, A Dynamic Model Selection Strategy for Support Vector Machine Classifiers,*Applied Soft Computing*, Vol. 12, No. 8, 2012, pp.2550–2565.

[20] D. Qiu, Y. Li, X. Zhang&B. Gu,Support Vector Machine with Parameter Optimization by Bare Bones Differential Evolution, *Proceedings of the 7th IEEE International Conference of the Natural Computation,* 26-28 July 2011, Shanghai, 263-266.

[21] J. Sartakhti, M. Zangooei&K. Mozafari, Hepatitis Disease Diagnosis Using a Novel Hybrid Method Based on Support Vector Machine and Simulated Annealing (SVM-SA), *Computer Methods and Programs in Biomedicine*, Vol. 108, No. 2, 2012, pp. 570-579.

[22] F. Samadzadegan, A. Soleymani&A. Abbaspour,Evaluation of Genetic Algorithms for Tuning SVM Parameters in Multi-Class Problems, Proceedings of the *11th IEEE International Symposium on Computational Intelligence and Informatic,*18-20 November 2010, Budapest, 323-328.

[23] *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.*

[24] M. Zhao, C. Fu, L. Ji, K. Tang& M. Zhou, Feature Selection and Parameter Optimization

for Support Vector Machines: A New Approach Based on Genetic Algorithm with Feature Chromosomes,*Expert System with Applications, Vol. 38,* 2011, 5197-5204.

[25] H.B.Alwan & K.R. Ku-Mahamud, Feature Selection and Model Selection Algorithm Using Incremental Mixed Variable Ant Colony Optimization for Support Vector Machine Classifier,*International Journal of Mathematics and Computers in Simulation,* Vol. 7(7), 2013, pp. 700-707.