HYBRID ACO AND SVM ALGORITHM FOR PATTERN CLASSIFICATION

HIBA BASIM ALWAN

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2013

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Pengoptimum Koloni Semut (ACO) adalah algoritma metaheuristik yang boleh digunakan untuk menyelesai pelbagai masalah pengoptimuman kombinasi. Halatuju baru bagi ACO adalah untuk mengoptimumkan pembolehubah selanjar dan bercampur (diskrit dan selanjar). Mesin Vektor Sokongan (SVM) adalah satu pendekatan klasifikasi corak yang berpunca daripada pendekatan statistik. Walau bagaimanapun, SVM mempunyai dua masalah utama iaitu pemilihan atribut subset dan penalaan parameter. Kebanyakan pendekatan yang berkait dengan penalaan parameter SVM mendiskritkan nilai selanjar parameter dan ini akan memberi kesan negatif kepada prestasi klasifikasi. Tesis ini melaporkan empat algoritma untuk menala parameter SVM dan memilih atribut subset yang meningkatkan prestasi klasifikasi SVM dengan saiz attribute subset yang lebih kecil. Ini boleh dicapai dengan melaksanakan proses pemilihan subset dan penalaan parameter SVM secara serentak. Penghibridan algoritma ACO dan teknik SVM telah dicadangkan. Dua kelompok algoritma pertama iaitu algoritma ACO_R-SVM dan $IACO_R$-SVM akan menala parameter SVM manakala dua algoritma kedua iaitu algoritma ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM boleh melaksanakan penalaan parameter SVM dan pemilihan atribut subset secara serentak. Sepuluh dataset penanda aras dari University California, Irvine, telah digunakan dalam eksperimen untuk mengesahkan prestasi algoritma yang dicadangkan. Dapatan eksperimen daripada algoritma yang dicadangkan adalah lebih baik berbanding pendekatan lain dari segi ketepatan klasifikasi dan saiz subset atribut. Purata ketepatan klasifikasi bagi algoritma ACO_R-SVM, $IACO_R$-SVM, ACO_{MV-R} dan $IACO_{MV-R}$ adalah 94.73%, 95.86%, 97.37% dan 98.1%. Purata saiz atribut subset adalah lapan bagi algoritma ACO_R-SVM dan $IACO_R$-SVM dan empat bagi algoritma ACO_{MV-R} dan $IACO_{MV-R}$. Dapatan kajian ini turut menyumbang kepada halatuju baru bagi ACO yang boleh digunakan untuk pembolehubah ACO yang selanjar dan bercampur.

Kata kunci: Pengoptimum koloni semut selanjar, Pengoptimum koloni semut bercampur, Mesin vektor sokongan, Penalaan parameter SVM, Pemilihan subset atribut.

Abstract

Ant Colony Optimization (ACO) is a metaheuristic algorithm that can be used to solve a variety of combinatorial optimization problems. A new direction for ACO is to optimize continuous and mixed (discrete and continuous) variables. Support Vector Machine (SVM) is a pattern classification approach originated from statistical approaches. However, SVM suffers two main problems which include feature subset selection and parameter tuning. Most approaches related to tuning SVM parameters discretize the continuous value of the parameters which will give a negative effect on the classification performance. This study presents four algorithms for tuning the SVM parameters and selecting feature subset which improved SVM classification accuracy with smaller size of feature subset. This is achieved by performing the SVM parameters' tuning and feature subset selection processes simultaneously. Hybridization algorithms between ACO and SVM techniques were proposed. The first two algorithms, ACO_R-SVM and $IACO_R$-SVM, tune the SVM parameters while the second two algorithms, ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM, tune the SVM parameters and select the feature subset simultaneously. Ten benchmark datasets from University of California, Irvine, were used in the experiments to validate the performance of the proposed algorithms. Experimental results obtained from the proposed algorithms are better when compared with other approaches in terms of classification accuracy and size of the feature subset. The average classification accuracies for the ACO_R-SVM, $IACO_R$-SVM, ACO_{MV-R} and $IACO_{MV-R}$ algorithms are 94.73%, 95.86%, 97.37% and 98.1% respectively. The average size of feature subset is eight for the ACO_R-SVM and $IACO_R$-SVM algorithms and four for the ACO_{MV-R} and $IACO_{MV-R}$ algorithms. This study contributes to a new direction for ACO that can deal with continuous and mixed-variable ACO.

Keywords: Continuous ant colony optimization, Mixed-variable ant colony optimization, Support vector machine, Tuning SVM parameters, Feature subset selection.

Acknowledgement

When I write this acknowledgment, my tears are falling down. Some memories can never be deleted and I remember the first time I decided to complete my studies in a foreign country; when I placed my feet on Malaysian soil I thought I had made the biggest mistake of my whole life; I felt so afraid of studying in a foreign country with strange people. In that moment, I realized I had left my home country to live in a foreign land. But, when I met Malaysian people with their beautiful smiles, and everything smells so fresh, so exciting, I thought it may well be a good journey.

After three years spent in Malaysia, I am filled with confidence that I made the correct decision to complete my studies here and I feel fortunate to have spent three years of my life in this amazing country with so many wonderful and incredible people. Sometimes, life can move too quickly; we all need to rest. Reflecting back you happily discover that the journey you have taken has brought you to a place with a magnificent view; you are now standing on a mountain top with a vista stretching towards a horizon of green trees and mountains. You smile into the distance realizing you are no longer looking back but are viewing new, sweet dreams. Which direction to take? You do not know. There are the memories, the experience, the wisdom, and the smiles. Gather them to you and start walking.

After three years as a PhD student at UUM, I am grateful to the many inspirational people who have helped shape my future, guiding me at times of difficulty, laughing with me in times of discovery. I have many beautiful memories, painted like a vivid picture in my mind, of the days spent with such amazing people. So, I will take this moment, take a deep breath and recollect all the wonderful memories from these past few years and walk down the grassy slopes of the mountain to discover what lies ahead in the continuation of my journey. Now, let me start my acknowledgments:

First and foremost, I would to thank “ALLAH” for being the source of my spiritual strength and giving me the power to finish my studies, for helping and guiding me through all these years and especially in my PhD study period; I aspire to become one of the good mumineen in this world. ALLAH, I am forever grateful to You.

I would like to express my deepest gratitude to my supervisor, Prof. Dr. Ku Ruhana Ku-Mahamud, for giving me the opportunity to study under her direction. I deeply appreciate her views, comments and tireless encouragement. Through her guidance, probing questions and sharing of knowledge, which I will treasure always, I have gained a wealth of knowledge. You broke the barrier of fear for studying in a foreign country, you made my PhD study easier and simpler.

Last but not least, my thanks go my parents who have encouraged me, given me the skills to step courageously into the future with confidence and taught me the importance of being true to myself and working hard, Most importantly, they have walked beside me during my journey, giving love and support. They embody, for me, the value of "family". They always have and always will provide words of wisdom, encouragement and prayers. I thank my parents for inspiring optimism and supporting me unconditionally. To them, I dedicate this thesis. ALLAH, please keep both of them for me.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

NN	Neural Network
ACO	Ant Colony Optimization
SVM	Support Vector Machine
OSH	Optimal Separating Hyperplane
OAO	One-Against-One
OAA	One-Against-All
FS	Feature Selection
TSP	Travelling Salesman Problem
TS	Tabu Search
SA	Simulated Annealing
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
ACO_R	continuous Ant Colony Optimization
ACO_{MV}	mixed-variable Ant Colony Optimization
$IACO_R$	Incremental continuous Ant Colony Optimization
$IACO_R$-LS	Incremental continuous Ant Colony Optimization with Local Search
CSO	Cat Swarm Optimization
CSA	Clonal Selection Algorithm
GSA	Gravitational Search Algorithm
RBF	Radial Basis Function
OAO	One-Against-One
UCI	University of California, Irvine
k-NN	k-Nearest Neighbor
BEOBDW	Binary Encoded Output Based Data Weighting
ANNs	Artificial Neural Networks
MLR	Multiple Linear Regression
RLS	Regularized Least Squares
LSVM	Linear Support Vector Machine
LibSVM	Library SVM
CV	Cross Validation

LibLinear	Library Linear
LOOCV	Leave-One-Out Cross-Validation
EMG	ElectroMyoGraphy
MAV	Moving Average
RMS	Root Mean Square
VAR	Variance
SD	Standard Deviation
ZC	Zero-Crossing
SSC	Slop Sign Change
WL	Waveform Length
FCM	Fuzzy C-Means
PCA	Principle Component Analysis
AFS	Axiomatic Fuzzy Set
MFAM	Modified Fuzzy Ant Miner
OMFAM	Optimizing Modified Fuzzy Ant Miner
FAM	Fuzzy Ant Miner
BPNN	Back Propagation Neural Network
MSE	Mean Square Error
∆ZCC	delta Zero Crossing Counts
GMM	Gaussian Mixture Model
EM	Expectation Maximization
RBFNN	Radial Basis Function Neural Network
DT	Decision Tree
MLP	Multi-Layer Perceptron
PBC	Pattern Based Classification
SFLS	Shortest Feature Line Segment
NFL	Nearest Feature Line
NN	Nearest Neighbor
RNFLS	Refined Nearest Feature Line Segment
EMD	Empirical Model Decomposition
WSVM	Wavelet Support Vector Machine
ARs	Association Rules
DE	Differential Evolution

CVC	Classification Via Clustering
DPSO	Dynamic Particle Swarm Optimization
GS	Grid Search
CPSO	Chained Particle Swarm Optimization
DMS	Dynamic Model Selection
KCRF	Kernel Condition Random Field
PCA	Principle Component Analysis
LKFSVM	Linear Kernel Function Support Vector Machine
GSVM	Gauss kernel Support Vector Machine
BBDE	Bare Bones Differential Evolution
SVM-SA	Support Vector Machine and Simulated Annealing
PWC	Pair-Wise Coupling
R-M- bound	Radius Margin bound
UD	Uniform Design
ES	Empirical Set
RFE	Recursive Feature Elimination
EEG	Electroencephalography
IAPS	International Affective Picture System
SSDOI	Scalp Spectral Dynamics Of Interest
MI	Mutual Information
AV	Artificial Variables
CHB	Chronic Hepatitis B
CIR	CIRrhosis
HCC	HepatoCellular Carcinoma
BR	Bootstrap Resampling
VF	Ventricular Fibrillation
ECG	ElectroCardioGram
VT	Ventricular Tachycardia
RADAG	Reordering Adaptive Directed Acyclic Graph
NPLs	Nonperforming Loans
IFSFS	Improved F-score and Sequential Forward Search
SFS	Sequential Forward Search
SVML	SVM with linear kernel

TF-IDF	Term Frequency-Inverse Document Frequency
mRMR	minimum Redundancy-Maximum-Relevance
ACS	Ant Colony System
ARD	Automatic Relevance Determination
SVM-FuzCoC	Fuzzy Complementary of Criterion
LS-SVM	Least Square SVM
KFFS	Kernel F-score Feature Selection
Lin	Linear
LMANN	Levenberg-Marquart Artificial Neural Network
AS	Ant System
EAS	Elitist Ant System
IG	Information Gain
NGL	Ng, Goh, and Low
GSS	Galavotti, Sebastiani, and Simi
OR	Odds Ratio
ICA	Immune Clonal Algorithm
MBPSO	Modified Binary Particle Swarm Optimization
RGSA	Real value GSA
BGSA	Binary value GSA
SVR	Support Vector Regression
kp-SVM	kernel-penalized SVM
ACO-S	Ant Colony Optimization-Selection
BP	Back Propagation
WT	Wavelet Transform
ACOFS	Ant Colony Optimization for Feature Selection
AS_{rank}	Rank-based Ant System
MMAS	Max-Min Ant System
SBS	Sequential Backward Selection
MFB	Mel-Filter Bank
LPR	Linear Predictive Reflection
RF	Random Forest
CACO	Continuous Ant Colony Optimization
API	pachycondyla apicalis

CIAC	Continuous Interacting Ant Colony
HCIAC	Hybrid Continuous Interacting Ant Colony
DHCIAC	Dynamic Hybrid Continuous Interacting Ant Colony
ACA	Ant Colony Algorithm
ACBACO	Adaptive and Commutative Binary Ant Colony Optimization
BACO	Binary Ant Colony Optimization
COAC	Continuous Orthogonal Ant Colony
OS	Orthogonal Scheme
OSACO	Orthogonal Scheme Ant Colony Optimization
CACO-DE	Continuous ACO-Direct Encoding
BAS	Binary Ant System
CACS	Continuous Ant Colony System
DACO	Direct Ant Colony Optimization
DACA	Dynamic Ant Colony Algorithm
MACACO	Multivariate Ant Colony Algorithm for Continuous Optimization
PDF	Probability Density Function
ACO$_R$-LM	ACO$_R$-Levenberg Marquardt
IEEE	Institute of Electrical and Electronics Engineers
ACO$_{MV-O}$	mixed-variable Ant Colony Optimization for Ordering variables
ACO$_{MV-C}$	mixed-variable Ant Colony Optimization for Categorical variables
PDS	Pressure Vessel Design
CSD	Coil Spring Design
DTIS	Designing Thermal Insulation System
M-IACO$_R$	Modifies Improved continuous Ant Colony Optimization
SamACO	Sampling Ant Colony Optimization
DNA	Deoxyribonucleic acid
EI	exon/introns boundaries
IE	introns/exon boundaries
F-score	Fisher score
SVC	Support Vector Classification

CHAPTER ONE

INTRODUCTION

Classification is a supervised learning approach which is a significant field of research involving labeling an object to one of a group of classes, related to features of that object (Qian, Chen & Cai, 2011; Khashei, Hamadani & Bijari, 2012; Khashei, Hamadani & Bijari, 2011; Tsai et al., 2011; Cheng et al., 2010; Liu, Liu & Zhang, 2010; Mastrogiannis, Boutsinas & Giannikos, 2009; Tseng & Lee, 2009; and Uney & Turkay, 2006) and it is considered one of the basic difficulties in a numerous decision making processes. Many decision making processes are examples of classification difficulty or can be simply transformed into classification difficulty, for example, prognosis processes, diagnosis processes, and pattern recognition (Orkcu & Bal, 2011). Data classification process consists of: training and testing and this is 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 (Uney & Turkay, 2006; and Cheng et al., 2010). 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, regarding 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 (Qian, Chen & Cai, 2012; Khashei, Hamadani & Bijari, 2012; Khashei, Hamadani & Bijari, 2011; Tsai et al., 2011; Cheng et al., 2010; Liu, Liu & Zhang, 2010; Mastrogiannis, Boutsinas & Giannikos, 2009; Tseng & Lee, 2009; and Uney & Turkay, 2006). The majority of the available research is centred on enhancing the classification accuracy by utilizing statistical approaches (Tseng & Lee, 2009).

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 (Wang et. al, 2012). In other words, it aims to classify input features into pre-determined groups consisting of classes of patterns (Thomas & Oommen, 2013 and Sivagaminathan & Ramakrishnan, 2007) and it requires comprehending the classification function. The classification function allocates a present input pattern often appearing as a vector of attribute values to a finite group of classes (Kanan, Faez & Taheri, 2007; and Basiri, Aghaee & Aghdam, 2008). Certain pattern classification techniques allow the input data to include many features but, in reality, only a few of them are relevant to the classification. In certain circumstances, it is not suitable to choose a large number of features, as this may be difficult to calculate or may be incomplete (Vieira, Sousa & Runkler, 2007). Therefore, choosing few and related features has its benefits, minimizing both the calculation effort and complexity of the approach as the generalization capability may be enhanced (Maldonado, Weber & Basak, 2011). In order to classify features, classification methods seek to determine the most pertinent features. Less pertinent features may be translated as noise which would decrease the accuracy of the classification. Good classification is usually achieved by using classifiers that are constructed from data or heuristic information (Vieira, Sousa & Runkler, 2007). Approaches such as a

Neural Network (NN) (Khashei, Hamadani & Bijari, 2012 and Ahsan, Ibrahimy & Khalifa, 2012), Ant Colony Optimization (ACO) (Otero, Freitas & Johnson, 2012), Fuzzy (Verma & Yadava, 2012; and Wang et al., 2012), Support Vector Machine (SVM) (Sarafrazi & Pour, 2013; Fu et al., 2013; Lázaro-Gredilla, Gómez-Verdejo & Parrado-Hernández, 2012; Bhadra, Bandyopadhyay & Manlik, 2012; Kapp, Sabourin & Maupin, 2012; Lin et al., 2012; Atienza et al., 2012; Feki, Ishak & Feki, 2012; Xie & Wang, 2011; Zhao et al., 2011; and Huang, 2009), and hybrid approach (Kabir, Shahjahan & Murase, 2012; Phokharatkul & Phaiboon, 2011; and Zhang et al., 2010) have been used to classify patterns.

SVM is an excellent classifier built on a statistical learning approach; however, it is not able to avoid the influence of huge numbers of unrelated or redundant features on the classification results. Therefore, selecting a few numbers of suitable features would result in obtaining good classification accuracy (Liu & Zhang, 2009). The main concept of SVM is to obtain the Optimal Separating Hyperplane (OSH) between the positive and negative samples. This can be completed 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 (Vapnik & Vashist, 2009; Qi, Tian & Shi, 2013; Fu et al., 2013; Khalid & Razzaq, 2012; Khashei, Hamadani & Bijari, 2012; Tasi et al., 2011; Li & Tan, 2010; Al-Naami et al., 2010; Cortes & Vapnik, 1995). Margin means the distance of the closest training sample to the hyperplane (Bajla et. al, 2009; Zhang & Mao, 2009; Liu & Yuan, 2009; Wang & Chen, 2007; Zhang, Pham & Zhang, 2007; Choi & Noh, 2004; and Ren et al., 2004). SVM is initially designed to classify a binary classes problem (Zhang et al., 2009), later, SVM is expanded to classify multi-class problems through utilizing decomposition approaches like One-Against-One (OAO) and One-Against-All (OAA) (Tasi et al., 2011). Nevertheless, there are two problems in the SVM classifier that influence the classification accuracy: selecting the optimal feature subset; and, tuning SVM parameters to be used in the SVM classifier. These two problems are essential because they affect each other (Huang & Wang, 2006; Pham et al., 2007; Lin et al., 2008; Huang & Dun, 2008; Lin & Chien, 2009; Ding & Li, 2009; and Zhao et al., 2011). Other SVM problems include selecting suitable kernel functions (Tasi et al., 2011).

The selection of features to be used for classification could affect the classification accuracy function, required dataset training, and applicable costs related with the classification (Sivagaminathan & Ramakrishnan, 2007). Feature Selection (FS) is the process of determining a subset of fields in the database, minimizing the number of fields that appear during data classification (Sivagaminathan & Ramakrishnan, 2007; Huang & Wang, 2006; and Huang, 2009). The main idea behind FS is to select a subset of input variables by deleting features that contain less or no information (Vieira, Sousa & Runkler, 2007). FS aims to decrease the dimension of the initial features group by determining the unauthentic features which would eventually supply the best performance under a certain classification dataset (Kabir, Shahjahan & Murase, 2009 and Kabir, Shahjahan & Murase, 2012), and to delete unrelated, unneeded, or noisy features while preserving the richness of the instructive ones (Abd-Alsabour, 2010; and Sanz et al., 2002). FS may be considered as an optimization problem which seeks potential features subsets which ultimately determines the optimal one (Abd-Alsabour, 2010). Exhaustive search is the easiest way to identify the best subsets. A FS problem is a hard combinatorial process in machine learning but it has high practical importance (Frohlich, Chapelle & Scholkopf, 2003). Recently, research works have been centered on solving FS problems using ACO. However, in FS problems, there is no previous information about the features to be selected that can be used to guide the search procedure such as in the Traveling Salesman Problem (TSP) where the distance between cities can be used to help search (Abd-Alsabour, 2010; Vieira, Sousa & Runkler, 2007; and Jensen, 2006). The goal of feature selection built on ACO is to obtain, if possible, the minimum number of features and to find high classification accuracy with minimum calculation efforts. This problem must be expressed as a TSP so that ACO can be used. Nodes or cities in TSP express features in the feature group source and the edge between nodes express the selection of the next feature. The search for the optimal feature subset is then changed into the problem of obtaining the path with the smallest number of nodes utilizing certain conditions, like classification accuracy (Huang et al., 2012; Nemati et al., 2009; Kanan & Faez, 2008; Khushaba et al., 2008a; Khushaba et al., 2008b; and Al-Ani, 2005).

In order to find optimal feature subset for SVM, optimization techniques such as Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA), Particle

Swarm Optimization (PSO), and ACO can be used to select suitable features (Kanan, Faez & Taheri, 2007; and Basiri, Aghaee & Aghdam, 2008). While for tuning the SVM parameters value, approaches such as trial and error, grid, cross validation, generalization error estimation and gradient descent and, also, global stochastic optimization approaches such as GA, PSO, SA, and ACO may be utilized (Qiu et al., 2011 and Zhang, Chen & He, 2010).

ACO is a metaheuristic for hard discrete optimization problems initially suggested at the beginning of 1990s. ACO is based on the behavior of real ants in collecting food. Ants when seeking for food, firstly investigate the region bordering their nest in a non-systematic way. When an ant detects a food source, it evaluates quantity and quality of the food and transports some of it to the nest. Throughout the return trip, the ant deposits pheromones on the land and the amount of pheromone depends 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 (Dorigo & Stützle, 2003; Blum & Dorigo, 2004; and Blum & Dorigo, 2005). In order to solve the optimization problem, ACO will repeat the following two steps (Dorigo & Blum, 2005): nominee solutions are built using a pheromone and the nominee solutions are used to update the pheromone values in order to gain high quality solutions.

While ACO was first presented to solve discrete optimization problems, it is modified to solve continuous and mixed optimization problems. However, some research has expanded ACO, applying it to continuous and mixed-variables optimization problems. One of the most interesting ACOs for continuous variables and mixed-variables is Socha's work which is called continuous ACO (ACO_R) and mixed-variable ACO (ACO_{MV}) respectively (Socha, 2004; Blum, 2005; Dorigo & Blum, 2005; Socha & Blum, 2006; Dorigo, Birattari & Stutzle, 2006b; Castro, 2007; Socha, 2008; Socha & Dorigo, 2008; Dorigo & Stutzle, 2010; Liao, 2011; and Liao et al., 2011). ACO_R was later modified by Liao (2011) and introduces two new algorithms called Incremental ACO_R ($IACO_R$) and Incremental ACO_R with Local Search ($IACO_R$-LS) (Liao, 2011; and Liao et al. 2011). Additionally, Liao (2011) suggests modifying ACO_{MV} regarding his two new algorithms.

All four ACO variants follow the same classical ACO framework, apart from the discrete probability used to build the ant solution which is replaced by continuous probability

1.1 Problem Statement

SVM has been used to solve classification problems with acceptable accuracy while simultaneously optimizing both feature subset selection and SVM parameters. The simultaneous operation is produced by hybridizing ACO (Huang, 2009), Simulated Annealing (SA) (Lin et al., 2008), Particle Swarm Optimization (PSO) (Huang & Dun, 2008 and Lin et al., 2008), Bess algorithms (Pham et al., 2007), Cat Swarm Optimization (CSO) (Lin & Chien, 2009), Clonal Selection Algorithm (CSA) (Ding & Li, 2009), Genetic Algorithm (GA) (Huang & Wang, 2006 and Zhao et al., 2011) or Gravitational Search Algorithm (GSA) with SVM (Sarafrazi & Pour, 2013). However, when ACO or other approaches such as GA (Huang & Wang, 2006 and Zhao et al., 2011), PSO (Lin et al., 2008), CSO (Lin & Chien, 2009) and CSA (Ding & Li, 2009) are hybridized in SVM, the continuous values of SVM parameters will have to be converted into discrete values before an optimization process is carried out (Zavar et al., 2011; Blondin & Saad, 2010; and Huang, 2009). The average classification accuracy for the studies is 92.27. This conversion process has resulted in the loss of some information and will affect the classification accuracy (Huang, 2009), because it will restrict the details at which hopeful regions of the seek domain can be investigated (Blondin & Saad, 2010). To overcome this problem, new ways to take continuous values of SVM parameters without converting to discrete forms will be researched.

When ACO is used to select feature subset, the features would be represented as discrete graph nodes, while SVM parameters are naturally continuous (Huang, 2009). To overcome this problem, new ways to accept mixed-variables (continuous and discrete) (Sarafrazi & Pour, 2013) of SVM mixed-variables will also be researched.

In a bid to overcome the limitation of working with discrete values, an algorithm that can handle both discrete or continuous values, and even mixed (discrete and continuous) values with the ability to perform the simultaneous optimization process for both feature subset selection and tuning SVM parameters has to be proposed. The classification accuracy can be enhanced by selecting suitable feature subset and tuning SVM parameters based on feedback supplied by ACO. Hence, the research questions posited center on finding high classification accuracy by answering the following:

i. How to optimize SVM parameters?

ii. How to select the suitable feature subset for the SVM?

iii. How to hybridize ACO with the SVM?

iv. Can the proposed algorithms enhance the performance of the SVM in terms of classification accuracy and size of feature subset selection?

1.2 Objective

The main objective of this research is to propose new algorithms that simultaneously optimize both feature subset selection and tune SVM parameters based on hybridized ACO and SVM for pattern classification. The following specific research objectives are to be fulfilled:

i. To propose enhanced pattern classification algorithms based on ACO-SVM that can handle continuous SVM parameters.

ii. To propose techniques for feature subset selection in non inductive and inductive learning approaches.

iii. To propose enhanced pattern classification algorithms based on ACO-SVM that can simultaneously optimize feature subset and SVM parameters.

iv. To evaluate the performance of the proposed enhanced pattern classification algorithms.

1.3 Significance of the Research

The approach in this research sets a new direction for ACO by extending the established standard ACO that handles mainly discrete values to now handle both discrete or continuous and mixed-variables. This new direction will then be used in this research to simultaneously optimize feature subset selection and tuning parameters for SVM using ACO. This research suggests intelligent algorithms that hybridize four new algorithms; standard ACO for discrete variables and its extension for continuous and mixed-variables hybrid with SVM. The outcome of this study is a novel algorithmic optimization that can notably increase the classification accuracy with small numbers of selected features as compared to the existing classification approaches.

1.4 Scope, Assumption and Limitations of the Research

This thesis focuses on using ACO and its variants (ACO_R, $IACO_R$, ACO_{MV-R}, and $IACO_{MV-R}$) to identify the suitable input feature subset and value for SVM parameters. Also, in this research, ACO and its variants will be hybridized with SVM through a wrapper-based feature selection approach. This is done because it gives the ability to hybridize the classification accuracy and necessary features introduced to the classifier from ACO and its approach to variants. It is also done because feature subset selection via the wrapper approach is dependent on the inductive learning approach. Beside the wrapper approach, a filter approach through using Fisher (F)-score technique is also utilized in this thesis in order to selecting suitable feature subset to introduce to SVM. The kernel function that will be used in this study is the Radial Basis Function (RBF) because it requires a few parameters and has been proven to produce good results in many problems. A binary SVM classifier as well as a multi-class SVM classifier through utilizing OAO is used in this thesis.

All experiments were carried out utilizing ten benchmark datasets from the University of California, Irvine (UCI) Repository of Machine Learning's benchmark datasets from various fields. The proposed algorithms were compared with the GSA-SVM (Sarafrazi & Pour, 2013), $\text{GA}_{\text{with feature chromosome}}$-SVM and Grid search (Zhao et al., 2011), ACO-SVM (Huang, 2009), CSO-SVM (Lin & Chien, 2009), CSA-SVM (Ding & Li, 2009), PSO-SVM (Lin et al., 2008), SA-SVM (Lin et al., 2008), and GA-SVM (Huang & Wang, 2006) in terms of classification accuracy and feature subset size.

This thesis deals with solving two problems of classical SVM. These are: selecting suitable feature subset; and tuning SVM parameters. Other SVM problems related to selecting suitable kernel function are not the concern of this thesis. Also other variants of SVM regarding classification or regression are not considered in this thesis.

1.5 Thesis Organization

This thesis has six chapters, including the introductory chapter, which covers the background information related to the problem that this thesis attempts to solve it. Chapter Two covers the literature review of related studies on pattern classification and its approaches. The second part of this chapter focuses on the SVM and its problems. Meanwhile, the third part presents the studies used by ACO to select feature subset as well as extending the classical ACO to handle continuous and mixed-variables.

Chapter Three presents the methodology that has been used in conducting this research. It is divided into five sections. The first section outlines the research framework that has been used in this thesis. The second section presents the datasets' development that has been used in this thesis which is, in turn, divided into four subsections. The first subsection presents a description of the datasets that have been used; the second subsection is related to cleaning datasets while the third subsection describes the needed transformation for some datasets; and the last subsection relates to how the datasets are scaled. The third section presents the methodology that has been used to tune SVM parameters; and the fourth section presents the methodology that has been used to simultaneously tune SVM parameters and select feature subset. Finally the summary of the chapter is given in fifth section.

Chapter Four presents two new hybridized algorithms of ACO_R and $IACO_R$ with SVM. These two approaches are used only to tune SVM parameters in order to obtain optimal values for SVM parameters to use it to classify patterns. This chapter compares the results with other hybrid SVM approaches in terms of classification accuracy and feature subset size.

Chapter Five presents another two new hybridized algorithms of ACO_{MV-R} and $IACO_{MV-R}$ with SVM. Differing from the two algorithms proposed in Chapter Four these approaches, instead of only optimizing SVM parameters, also simultaneously optimize feature subset selection and tuning SVM parameters. This chapter compares the results with other hybrid SVM approaches in terms of classification accuracy and feature subset size.

Finally, Chapter Six gives the concluding remarks on the four proposed hybrid algorithms. The concluding remarks consist of the description of features, capabilities and weaknesses of the four proposed approaches. This chapter also presents some recommendations as guidelines for further research for hybrid ACO and SVM for pattern classification.

CHAPTER TWO

LITERATURE REVIEW

This chapter reviews the literature related to the research field considered in the thesis. Section 2.1 presents the concept of pattern classification and its techniques, while Section 2.2 discusses the SVM and the solution for its problems. Section 2.3 gives the variations of the implementation of ACO for feature subset selection, and continuous and mixed-variables optimization problems. Finally, Section 2.4 summarises this chapter.

2.1 Pattern Classification Technique

Pattern classification is a significant topic whose objective is to utilize the training data to discover the planning relationship among the input and a number of classes. It is broadly exploited in a variety of subjects involving pattern recognition, decision making and artificial intelligence. At the present time, with classification difficulties becoming increasingly complicated, it may appear a good idea but it is more difficult to achieve a reliable classification execution (Acharya et al., 2010; Zhang & Li 2010; and Reed, Reed & Dascalu, 2010).

Pattern classification systems are improved to distinguish, classify, describe, or group patterns or items, which are often characterized through a number of measurements (i.e. feature vectors). After the pattern classification systems are built, it is an extremely important concern to assess the systems’ executions (Tsai, 2010).

Particularly, the main goal of pattern classification systems is to accomplish the automated assign pattern into one of a limited group of classes with the smallest percentage of misclassification. To achieve this goal, the new pattern classification research generally involves two cases: the build-up of strong single pattern classification algorithms or hybrid classifiers.

When a difficult classification crisis is faced, the majority of single classification algorithms will be deficient in obtaining a high classification accuracy percentage, stability of system and strength. To overcome these faults, some hybrid classifiers are proposed. These hybrid classifiers are possible and helpful in accomplishing high classification accuracy percentages. Hybrid algorithms research is presently growing in the pattern classification machine learning and decision sciences (Zhang et al., 2010). In the literature, much research deals with pattern classification, some of these are as follows.

A recent study by Theresa and Raj (2013) names “Fuzzy based genetic neural networks for the classification of murder cases using Trapezoidal and Lagrange Interpolation Membership Functions” to classify murder cases to help judges about to pass sentence in murder cases and as a learning tool for law students. The authors propose two classification systems: fuzzy neural network with random weights and fuzzy neural network with genetic algorithm based weights. Both systems, built on three layered artificial neural network with 27-3-3 architecture, utilize Lagrange interpolation and Trapezoidal membership functions. In this study, the feature value converts into fuzzy data based on: Gaussian, Lagrange and Trapezoidal membership functions used as input and output of a neural network which is trained using BackPropagation. 399 samples were used to evaluate the proposed classification systems, collected from district courts, high courts and Supreme Court decisions and law experts in India. The results demonstrate that fuzzy neural networks with random weights and genetic algorithm based weights utilizing trapezoidal membership function better than Lagrange interpolation and Gaussian membership functions.

Another recent study introduced by Tsai et al. (2013) called “Genetic algorithms in feature and instance selection” tries to classify patterns through not only selecting features but also through selecting instances by utilizing genetic algorithms. Five different strategies are used to build Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel function and k-Nearest Neighbor (k-NN) classifiers with different parameter values were tested. These strategies are: baseline, feature selection, instance selection, feature selection then instance selection and, finally, instance selection then feature selection. Eight University of California, Irvine (UCI) datasets varying from small scale to large scale are used to evaluate the proposed approach. Each dataset is divided into training and testing subsets and 10-fold cross validation is used to evaluate the classifier. The experimental results show that feature selection, then instance selection strategy, is better than other strategies in terms of classification accuracy for small scale datasets; while for large scale datasets, feature selection then instance selection and instance selection then feature selection strategies are better than other strategies. The authors suggest to compare baseline strategy with other algorithms that simultaneously select feature and instance and to test the execution of combining different features and instance selection algorithms as a hybrid approach.

Recent research was conducted on pattern classification using k-NN classifier, called "Data weighting method on the basis of binary encoded output to solve multi-class pattern classification problems" by Polat (2013). In this research, the author attempts to increase the classification accuracy and to simplify the complexity of datasets by data pre-processing through utilizing a new data weighting strategy named Binary Encoded Output Based Data Weighting (BEOBDW) to represent the output labels of datasets in only binary form, then the datasets are weighted utilizing relationships between datasets' features and these two binary codes. Subsequently, the k-NN classifier is used as to classify datasets and different values of *k* are used to decide upon the best *k* value in the classification of datasets. To test the proposed approach, five UCI datasets are used and the experimental results show that the proposed approach is very efficient and has robust discrimination ability in the classification of datasets. Unfortunately, the author does not compare his results with other exiting approaches, but he uses many performance measurements to evaluate his proposed approach.

A study based on Artificial Neural networks (ANNs) and Multiple Linear Regression (MLR) prototypes was presented by Khashei, Hamadani and Bijari (2012). MLR is utilized to summarize linear elements of features in new features then ANNs are utilized for data modelling and classification by utilizing original features and produced linear features through multiple linear regression. The authors evaluated their study on six benchmark synthetic and real life binary and multi-class datasets. One-versus-one, one-versus-all, and one-versus-rest approaches are utilized to solve multi-class problems, and the experiments' results show that one-versus-all was better than the other two approaches. The result of all experiments demonstrates the success of the proposed approach compared with classical multilayer perceptions, linear discriminant analysis, quadratic discriminant analysis, kNN, and SVM.

Fu et al., (2012) present a study called "Efficient nonlinear classification via low-rank regularized least squares". In their study, they utilize low-rank formulation of Regularized Least Squares (RLS) for large scale nonlinear classification and the results are compared with kernel and Linear SVM (LSVM) on the UCI repository benchmark datasets as well as on synthetic datasets. Both kernel SVM and LSVM were implemented using Library SVM (LibSVM) software with five-fold Cross Validation (CV) in order to choose regularization variable *C*, while RLS were implemented using Library Linear (LibLinear) software with Leave-One-Out Cross-Validation (LOOCV) to choose gamma variables. Additionally, a one-versus-all approach was used for multi-class datasets. The experiment results show that low-rank RLS accomplished comparable application. The authors suggest performing their approach on music annotation classification problems.

Research was conducted on classification using an Ant Colony Optimization (ACO) called “Inducing decision trees with an ant colony optimization algorithm” by Otero, Freitas and Johnson (2012). The main issue the authors attempt to solve is to maximize classification accuracy with a reasonably sized induced decision tree. Four variants of Ant-Tree-Miner based on discretization and pruning mechanisms were proposed; these variants are a binary entropy established discretization process, two stage tree pruning named Ant-Tree-Miner, and C4.5 mistake established pruning. All these variants were compared against two famous decision tree induction approaches, C4.5 and CART, implemented in Weka as well as against ACO established CART decision tree approaches. The experiments were made on twenty two datasets from UCI with ten-fold cross validation in terms of classification accuracy and size of obtained decision trees; and the results demonstrate that the Ant-Tree-Miner approach is better than other compared approaches, while the Ant-Tree-Miner based binary entropy established discretization process with two stage tree pruning are the best. For future work, the authors suggested using alternative heuristic information measures and utilizing alternative discretization approaches.

A study introduced by Ahsan, Ibrahimy and Khalifa (2012) aimed to classify ElectroMyoGraphy (EMG) signals of hand motion by utilizing many models of ANN. ANN was trained with back-propagation Levenberg-Marquardt with seven statistical time and time frequency based features: Moving Average (MAV), Root Mean Square (RMS), Variance (VAR), Standard Deviation (SD), Zero-Crossing (ZC), Slop Sign Change (SSC) and Waveform Length (WL). The results show that

ANN with ten neurons in hidden layers gives best classification accuracy. The authors suggest testing their approach with EMG signals taken from disabled or aged people because they could have alternate muscle structure and different ways to move hand muscles in all directions (right, left, down, and up).

Another study introduced by Verma and Yadava (2012) called “Fuzzy C-Means Clustering Based Uncertainty Measure for Sample Weighting Boosts Pattern Classification Efficiency” tries to minimize fuzziness in datasets to increase classification accuracy. In order to do this, the authors utilized Fuzzy C-Means (FCM) clustering for boosting the execution of RBF pattern classifier. The boost includes weighting pattern vectors related to fuzziness associated with them as created by the FCM clustering, which are then presented to an RBF classifier. The feature extraction is completed by utilizing Principle Component Analysis (PCA) on the dimensionally auto scaled data and this auto scaling is carried out by means of centring and variance normalization along measurement dimensions. The authors apply their work on fifteen datasets and the results show that FCM is highly reproducible and extremely easy.

A study on the use of Fuzzy Set for Classification by Wang et al., (2012) attempts to mine association rules for classification using fuzzy sets and underlying logic operation produced through Axiomatic Fuzzy Set (AFS). The authors present a new idea of a fuzzy class support to understand mining significant rules from data with imbalanced classes. Moreover, they also propose optimal fuzzy confidence truncation guiding to decrease the existing rule base in order to prevent the complexity of calculating the minimal confidence level. In order to evaluate AFS's accuracy and interpretability (rule size), the authors use their approach to examine twelve UCI datasets with ten-fold cross validation. They compare AFS with rule-based classifiers and with traditional classification approaches involving Bagging, k-NN, SVM, and Naïve Bayes; all the results illustrate that AFS performance is better than other methods in terms of obtaining high accuracy and smaller rule size while maintaining sensible trade-off between accuracy and interpretability.

Another study, presented by Aribarg, Supratid and Lursinsap (2012) titled "Optimizing the modified fuzzy ant miner for efficient medical diagnosis", improves Modified Fuzzy Ant Miner (MFAM) to prevent failures in local optimal issues by using Simulated Annealing (SA) to dynamically obtain optimal fuzzy set variables within MFAM; this will enhance quality of rule building in MFAM which depends on fuzzy set variables. Optimizing MFAM (OMFAM) evaluates six UCI medical datasets and compares them against MFAM, Fuzzy Ant Miner (FAM), and SVM. The results show that OMFAM is the best in terms of accuracy and interpretability. The authors suggest applying other advanced seeking approaches instead of SA.

The goal of a study between Fourier Descriptors and Neural Network (NN), presented by Phokharatkul and Phaiboon (2011), is to control chart pattern classification. The control chart data is portioned into three categories utilizing Pearson's correlation coefficients and then the Fourier Descriptor is utilized to extract control chart features which will be used as input vectors to train Back Propagation Neural Network (BPNN) to classify control chart patterns. The training process will be halted if epochs are reached or when average Mean Square Error (MSE) fails. The authors compare their work with a symbol-sequence histogram and the results show that their system gives reliable verification and high classification accuracy.

A paper titled "Novel Delta Zero Crossing Regression Features for Gait Pattern Classification" was presented by Ibrahim, Sethu and Ambikairajah (2010). In their approach, they utilize linear regression on delta Zero Crossing Counts (ΔZCC) to extract new dynamic features from a waste-mounted accumulated triaxial accelerometer to classify five kinds of walking patterns: flat, slope down, slope up, stairs down, and stairs up gait pattern. They utilized a single piezo-resistive waist-mounted tri-axial accelerometer which can capture both static and dynamic acceleration with in all orthogonal axes x, y, and z from 15 female and 37 males aged between 21 and 65 over a period of three months. They use the Gaussian Mixture Model (GMM) as a classifier and for every class of movement they trained a separate GMM by utilizing the Expectation Maximization (EM) approach, followed by the Bayesian approach to classify the overall individual's gait patterns. The results show that a ΔZCC regression feature is better when compared to the ZCC; also ΔZCC regression features execute well when integrated with filter bank features. As a future work, the authors suggest to examine higher order regression parameters.

Zhang et al. (2010) propose a new ensemble intelligent classifier. In their approach, they use three well-known classifiers which are a BPNN with twelve neurons in hidden layer, one neuron sigmoid output layer with Levenberg-Marquardt as the training method, a standard RBF with ten neurons in a hidden layer and Gaussian RBF as a transfer function, one neuron in the output layer, and finally a SVM with Gaussian kernel function. Each classifier is considered as a group of agents, so the whole structure can be considered as a multi-agent construct. Every agent contains a group of the same kind of intelligent classifiers by selecting alternative setting parameters or alternative sample training groups. The authors evaluate their approach on Monk's problem dataset and compare their approach against single BPNN, single Radial Basis Function Neural Network (RBFNN), single SVM, BPNN ensemble, and RBFNN ensemble. The results show that their approach produces very high classification accuracy.

Two similar studies, one introduced by Santana, Canuto and Silva (2011) called "Bio-inspired meta-heuristic as feature selection in ensemble systems: a comparative analysis" and the other by Santana et al., (2010) called "A comparative analysis of genetic algorithm and ant colony optimization to select attributes for an heterogeneous ensemble of classifiers" try to utilize Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) to select feature subset to introduce to the ensemble system classifier to classify four datasets and utilize 10-fold cross validation to evaluate the performance of the ensemble classifier as well as with and without selected features. The ensemble classifier system comprises of three individual classifiers which are k-NN, Decision Tree (DT), and Multi-Layer Perceptron Neural Network (MLP). In order to construct and define the learning process of the ensemble classifier system, the authors utilize a stacking procedure. Individual diversity and group diversity are two alternative parameters utilized in ensemble classifier systems and the results show that when GA is used selected feature subset is better than others in individual diversity, while PSO supplies better than other in diversity groups.

One-class classification through clustering was introduced by Salama, Hassanien and Fahmy (2010) in their paper titled “Uni-class Pattern-based Classification Model”. In their approach, they present a Pattern Based Classification (PBC) prototype that can extract patterns with likeness through all objects in an exact class. The difficulty lies in the user-defined threshold that arises in the pattern based subspace clustering is built on a classical clustering approach. The new element in this work is that the input training set has assigned class labels. The authors test their approach on two UCI datasets and the Chiba university dataset. The results show effectiveness and capability of PBC in term of classification accuracy compared against the Bayesian network, SVM, Multilayer Neural Network and Decision table. The authors suggest obtaining a suitable missing data imputation for information data (selected feature) and obtaining a method for rule extraction because it is considered as a challenge in the prototype.

Fang and Li (2009) propose an approach titled “An integrated space-time pattern classification approach for individuals’ travel trajectories”. Their approach can integrate space-time to classify individuals’ local and global travel trajectories. The main benefits of their approach are its sensitivity of travel movements and also its capability providing space-time analysis and pattern mining. The authors utilize Equal Interval Standard Deviation, Defined Interval, Natural Breaks and Quintile as classification techniques. Household travel characteristics survey data is utilized to classify and implement their approach. The results show that their approach is certainly appropriate to classify huge amounts of individual travel trajectories; however, it is not able to calculate the likeness between trajectories. Handling the classification and recognition of online-human activities is addressed by the authors in their future work.

Yang, Han and Han (2009) present Shortest Feature Line Segment (SFLS). The main idea of SFLS is that it tries to obtain the shortest feature line segment give geometric relation limitations and does not compute the distance between the query point and the feature line as with Nearest Feature Line (NFL); in this case, it can resolve NFL’s disadvantages which are related to extrapolation, interpolation and computational efforts. SFLS is tested on three UCI datasets and compared against Nearest Neighbor (NN), k-NN, Bayesian NFL, and Refined Nearest Feature Line Segment (RNFLS). The experiment results show that SFLS is an easy and successful classification technique. The authors suggest investigating for a faster and more successful approach for SFLS, modifying SFLS to be, for example, k-SFLS which follows the same concept of k-NN, utilizing other distance calculations in the SFLS approach and, finally, improving SFLS by some way to minimize the cost for computation.

2.2 Support Vector Machine

The SVM has been introduced as a successful statistical learning approach for classification. The original SVM was improved 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 (Yuxia & Hongtao, 2012; Li & Zhou, 2011; Qiu et al., 2011; Jin, Chen & Ma, 2010; Al-Naami et al., 2010; Zhang & Mao, 2009; Kapp, Sabourin & Maupin, 2009; Lu et al., 2009; Liu & Zhang, 2009; and Luo et al., 2008) which means, it decreases the tradeoff between the experimental mistake and the difficulty that arises from classification of estimating functions to prevent over fitting (Ju et al., 2009; and Bajla et al., 2009). The concept of structural risk minimization plans the data into high dimensional domains via kernel functions by using kernel tricks (Wang et al., 2010; Ye et al., 2009; Shieh & Yang, 2008; Lessmann, Stahlobock & Crone, 2006; Kadoury & Levine, 2006; Nguyen et al., 2006; and Vapnik, 1999). There are many types of kernel functions, such as polynomial, RBF, linear, and sigmoid kernel function. Nevertheless, RBF is the most popular kernel function because of its capability to manage high dimensional data (Moustakidis & Theocharis, 2010), good performance in major cases (Zhang et al., 2009), and it only needs to use one parameter, which is kernel parameter gamma (γ) (Huang & Wang, 2006). However, the disadvantage of

RBF is that it will fail if it deals with large numbers of features (Huang & Dun, 2008).

SVM has an extremely good generalization capability and strong theoretical foundation (Liu & Zhang, 2009). Generalization capability can be defined as the ability of SVM to classify unknown data examples correctly through constructed SVM. This is achieved by learning SVM from training examples which is also known as SVM performance (Zhang, Chen & He, 2010). The SVM manipulates the "curse of dimensionality", which means the computational complexity for the SVM training or testing is not affected by the feature space dimensionality (Wang et al., 2010 and Liu & Zhang, 2009).

There are two types of SVM: binary and multi-class. Binary SVM is the core of SVM. It is capable of distinguishing between two classes. Multi-class SVM expands binary SVM by being able to classify three or more classes. The main algorithm involved in multi-class SVM is to divide the classification problem to many binary problems with its own classifier (Harvey, 2009). Two main methods are related to SVM multi-class classification: One-Against-All (OAA) and One-Against-One (OAO). In OAA, a group of binary classifiers are trained to isolate each class from all others. Based on the largest decision value, each data object would be classified to the class. The main advantage of this method is its fast method; unfortunately, it is prone to errors obtained from the marginally unstable training groups. The other method, OAO, is a sequence of classifiers that is implemented on a pair of classes with the popular calculated class preserved in each object. Subsequently, a max-min operator is applied to calculate to which class the object will finally be allocated. This method is considered as being more accurate but it is slower than OAO (Tzotsos & Argialas, 2008).

Classification work of SVM strongly depends on selecting the appropriate kernel function, kernel parameter and suitable value for the regularization parameter *C* (Bhadra, Bandyopadhyay & Maulik, 2012; Zavar et al., 2011; Hric, Chmulik & Jarina, 2011; Jin et al., 2011; Liao, Yang & Ding, 2011; Zhang, Chen & He, 2010; Sun et al., 2010; Zhou & Xu, 2009; Zhang, 2008; Acir, Ozdamar & Guzelis, 2006; Lessmann, Stahlbock & Crone, 2006; and Wu & Li, 2006). Additionally, selecting small numbers and suitable feature subset also has an effect on SVM performance (Feki, Ishak & Feki, 2012; Tan et al., 2012; Hu et al., 2012; Xie & Wang, 2011; Zavar et al., 2011; Nguyen & Torre, 2010; Dunbar et al., 2010; and Liu & Zhang, 2009).

Nonetheless, there are two main strategies to classify patterns using SVM: simultaneous and non-simultaneous. Non-simultaneous strategy may be divided, in turn, into two approaches: optimizing only the value for the SVM parameters or optimizing only feature subset selection. One of the main downsides to the non-simultaneous strategy is that the selecting feature subset and tuning SVM parameters are completed separately and this would greatly influence the classification process as there might be relevant information which is lost during this process (Nguyen &

Torre, 2010). The current research direction has moved towards simultaneously optimizing both feature subset selection and tuning SVM parameters by using optimization algorithms. This approach will increase the classification accuracy because selecting suitable feature subset and values for SVM parameters influence each other and, in turn, will influence classification accuracy (Sarafrazi & Pour, 2013; Zavar et al., 2011; Han, Hou-Jun & Xiucheng, 2011; Zhao et al., 2011; Huang, 2009; Ding & Li, 2009; Lin & Chien, 2009; Huang & Dun, 2008; Pham et al., 2007; Lin et al., 2008; and Huang & Wang, 2006). Techniques such as PSO, ACO, GA, Cat Swarm Optimization (CSO), and Clonal Selection Algorithm (CSA) are used to simultaneously optimize feature subset selection and value for SVM parameters. The following subsections illustrate these strategies.

2.2.1 Approach in Solving Model Selection Problem

The SVM has recently been seen as a stronger approach for resolving difficulties in pattern classification but its execution depends, especially, on the variables chosen for it. Variables chosen for SVM are very complex and difficult to resolve via conventional optimization approaches (Reif, Shafait & Dengel, 2012; Anguita & Ghio, 2011; Xusheng, Wei & Yongxiang, 2010; and Zong, Liu & Don, 2006). The difficulty in selecting the values of SVM variables that decrease the prediction of test error is called the model selection problem (Kapp, Sabourin & Maupin, 2012; Anguita & Ghio, 2011; Glasmachers & Igel, 2010; Zhang, 2008; Chapelle & Vapnik, 2000; and Vapnik & Chapelle, 2000). However, researchers use previous skills in choosing SVM variables. In order to seek suitable variable groups, some implementation conditions of the prototype need to be selected. This point, at which a minimum training group mistake or validation group mistake is achieved, is chosen as the variable group to apply training and thereafter forecasting (Saini, Aggarwal & Kumar, 2010).

Model selection alternates, mainly, between two features: the chosen conditions and the seeking approaches utilized. The chosen conditions – meaning fitness function - is an assessment that directs the seek. Some are especially associated with the SVM formulation, such as: radius margin bound, span bound, and support vector number. Others are traditional, such as cross validation and hold-out assessments (Kapp, Sabourin & Maupin, 2012). Many researchers choose SVM variables experimentally by applying a bounded number of values and preserving the values that provide the minimum test mistake. Manual selection is avoided because it is not accurate and does not provide any assurance on the quality of the result (Imbault & Lebart, 2004).

As mentioned earlier, in order to optimize value for SVM parameters, techniques like trial and error, grid search, generalization error estimation and gradient descent, and evolutionary algorithms may be used to tune SVM parameters. Some disadvantages are associated with a few of these techniques such as: the trial and error technique has its shortcomings in uncounted results; grid search algorithm suffers from calculation complexity; and discretization of the seek domain in constant values is critical to achieve high performance; cross validation has the disadvantage of having long and complex computations; while in the gradient-based technique, choosing a non-suitable beginning point, will pose a problem; also, it needs a differentiable objective function regarding the hyper parameters and the kernel, which requires it to be differentiable; additionally, this technique suffers from multiple local minima in objective functions. Evolutionary algorithms such as ACO can simplify the choice of optimal value processes for SVM parameters (Kapp, Sabourin & Maupin, 2012 and Zhang, Chen & He, 2010).

The variables involved in optimization, *C* and kernel parameters, are necessary in constructing a successful and highly executable SVM prototype. When utilizing SVM, it is difficult to choose parameters in order to obtain high generalization execution. Therefore, investigating better approaches to choose optimal variables is often carried out. The purpose of optimizing variables for SVM is to obtain variables that decrease generalization mistakes (Diosan, Rogozan & Pecuchet, 2012 and Dong et al., 2007). It is seen as difficult because it requires either exhaustive searching through the space of variables or optimizing approaches that discover only a bounded subgroup of the potential values (Imbault & Lebart, 2004). There is also no regular methodology that accepts advance approximation of their optimal values. In fact, in the present classification work, obtaining good values for these parameters is not an easy model selection problem that requires either an exhaustive search through the space of hyper variables or an optimization approach that searches only a bounded sub group of the potential values (Zhang, Chen & He, 2010).

The kernel function parameter indirectly defines non-linear planning from the input domain to a high dimensional feature (attribute) domain where the maximal margin hyperplane is obtained (Zhang, Chen & He, 2010 and Tang, Guo & Gao, 2009); an extremely large value for kernel parameter can be due to over fitting, otherwise, it is due to under fitting; while regularization parameter *C* monitors the tradeoff between the complexity of the machine and the amount of non-separable points (training error). In other words, the *C* parameter monitors the tradeoff between margin maximization and error minimization (Lázaro-Gredilla, Gómez-Verdejo & Parrado-Hernández, 2012; Li, Liu & Gong, 2011; Zhang, Chen & He, 2010; Tang, Guo & Gao, 2009; and Chapelle et al., 2002). If *C* is too large, the assessment of precise percentage is very high in the training stage, but very low in the testing stage. If *C* is too small, the assessment precision is unsatisfied, and the model is useless. Kernel parameter has much greater impact on assessment than *C* (Liu et al., 2013). In the literature, much research deals with solving model selection problems, some of these are discussed below.

A recent paper titled “Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings” introduced by Liu et al. (2013) uses PSO to tune SVM parameters: *C* and wavelet kernel parameter α for multi-fault classification. In this paper, the author uses an empirical model decomposition (EMD) method to extract vibration signals measured from rolling: fourteen time-domain statistical features and thirteen frequency-domain statistical features are extracted from these signals, and then a distance evaluation method is used to select most superior features. In order to evaluate the proposed study, the authors compare SVM with Gaussian RBF kernel. The experimental results show that Wavelet SVM (WSVM) with two types of wavelet function: the Morlet and the Mexican wavelet kernel functions, obtain better classification accuracy than RBF–SVM, but the authors do not mention which wavelet kernel function type is better than others, also, they do not mention what PSO type is used to optimize and deal with the continuous value of SVM parameters. The authors suggest improving their work through utilizing advanced wavelet kernel functions that can adaptively match any curve in quadratic continuous integral domain because the wavelet kernel function used in their study presents as translation and dilation of one fixed function, the characteristics of the wavelet functions are isolated with the processed signals.

Another study on tuning SVM parameters conducted by Abdi and Giveki (2013) titled "Automatic detection of erythemato-squamous diseases using PSO–SVM based on association rules" uses Association Rules (ARs) to select feature subset and then utilizes PSO to tune SVM parameters: C and σ of RBF kernel parameter; they name their approach AR-PSO-SVM. The authors test their approach on an erythemato-squamous diseases dataset taken from UCI and this dataset is considered as a multi-class classification problem; so the author uses a OAA decomposition technique to construct a multi-class SVM classifier. In order to evaluate their approach, the authors also implement a Multi-Layer Perceptron (MLP) classifier utilizing ARs to select feature subset. Both approaches are implemented using MATLAB software and the authors compare their results with AR-MLP and SVM alone. The experimental results show that AR-PSO-SVM performs better in terms of classification accuracy and size of feature subset selection. Also in this paper, the authors do not mention what type of PSO is used to tune the continuous value of SVM parameters. The authors suggest applying their approach on other datasets benchmarks to test the applicability of their approach.

A recent study introduced by Azadeh et al. (2013) aims to classify two types of faults condition of centrifugal pump through six features: flow, temperature, suction pressure, discharge pressure, velocity and vibration, by implementing four methods: SVM alone, SVM with GA, SVM with PSO and feed forward MLP. The authors do not mention how the feature subset is selected. In this study, classical GA and PSO are used to tune SVM parameters: *C* and parameter of four kernel functions named: linear, quadratic, Gaussian, and polynomial. First SVM parameters are generated randomly then GA or PSO optimize this parameter and select the one that produces the best classification accuracy. The average classification accuracy is computed through utilizing a 10-fold cross validation method. The comparison results are carried in two directions: first, the four proposed methods are compared against each other and the results show that SVM-GA and SVM-PSO produce better classification accuracy while the second direction compares the proposed methods against k-NN and Decision Tree; the results show that ANN, SVM-GA and SVM-PSO produce better classification accuracy. The authors suggest applying their approach on multi-class classification problems.

A study presented by Lázaro-Gredilla, Gómez-Verdejo & Parrado-Hernández (2012) called “Low-cost model selection for SVMs using local features” uses a 10-fold CV technique to optimize the *C* parameter while, for optimizing Gaussian kernel width parameter, they present three different methods that combine information about the local construction of each dataset. These methods are, k-NN, nearest enemy, and redundant fast clustering. In their study, they implement the three methods for optimizing Gaussian kernel parameter in Matlab and they use LibSVM for implementing SVM. They evaluate their study on six UCI datasets and compare it with CV. The results show that k-NN finds the best value for kernel parameter for three datasets, nearest enemy obtains best value in only one dataset, and CV obtains the best value in two datasets; whereas, redundant clustering could not find any best value. The authors suggest extending their study to deal with multi-class classification; they also suggest determining the single features of each dataset that support application for each of the three methods in addition to using the information found regarding the local construction of the problem to generate a multi-kernel classifier, where every support vector is labelled to a different value of kernel parameter.

Research conducted on optimizing SVM parameters using Differential Evolution (DE) by Bhadra, Bandyopadhyay and Maulik (2012) attempts to solve how to optimize SVM parameters. In order to solve this issue, three factors are utilized (accuracy, sensitivity, and specificity) to generate fitness function that requires to be maximized; unfortunately, the authors do not explain how to compute these three factors or how the proposed study is to work. In their research, they propose a meta-classification prototype ($SVMDE_{Meta}$) using four types of kernel function which are linear, polynomial, RBF, and sigmoid. $SVMDE_{Meta}$ is evaluated on three real-life datasets: TargetMiner, and two UCI datasets. LibSVM is utilized to implement SVM and the experiments move in two directions; the first relates to comparing $SVMDE_{Meta}$ with $SVMDE_{Lin}$, $SVMDE_{RBF}$, $SVMDE_{Poly}$, and $SVMDE_{Sig}$ and the second relates to comparing $SVMDE_{Meta}$ with ensemble classifier such as Bagging, AdaBoost, LogicBoost, Classification Via Clustering (CVC), and ensemble selection. The results from these two directions show that $SVMDE_{Meta}$ is the most successful compared with other stated methods in terms of accuracy, sensitivity, and specificity.

Kapp, Sabourin and Maupin (2012) propose, in a study to optimize hyper-parameters for SVM by utilizing PSO and grid search in a dynamic environment, alternating among three stages: best solution obtained so far will be utilized; utilizing adapted grid search to seek for a new solution; or perform dynamic optimization procedure. To do this, their framework includes three basic modules; the first module is a change detection module to monitor the intensity of the seek task through addressing how the solutions are obtained by the stages of the framework; the second is an adapted grid search to supply optimum solutions through re-evaluating the knowledge taken from past optimizations applied by the Dynamic Particle Swarm Optimization (DPSO) module which will become the third and last module responsible for obtaining new solutions through re-optimization procedures. The authors examined their approach on fourteen datasets and compared their work with classical Grid Search (GS), 1st Grid Search (1st GS), Chained PSO (CPSO), and Dynamic Model Selection (DMS). The results show that their approach outperforms the classical methods in term of model complexity.

Another study, conducted by Lei and Qiao (2012), named "Text categorization using SVM with exponent weighted ACO" attempts to find the optimal value for σ kernel and C SVM parameters to enhance performance on classified Chinese text provided by Sogou Labs. The authors utilize exponent weighted ACO to optimize SVM parameters, the reason for which is to avoid local optimization and slow convergence speed in ACO. A Kernel Condition Random Field (KCRF) was utilized as a pre-processing step to segment words, extract features and representation, and then a combination of Principle Component Analysis (PCA) and noise elimination threshold was utilized to select feature subset. A one-to-one strategy was utilized in this study to classify multi-class text and the results compared with standard SVM, Linear Kernel Function SVM (LKF-SVM), AdaBoost, and Naive Bayes; the results show that the proposed study had optimal execution in terms of classification accuracy and efficiency. The authors suggest the use of a mutation procedure in ACO to avoid the local convergence problem as well as to use other swarm optimization techniques.

A study called "Parameter Selection Algorithm for Support Vector Machine" was conducted by Wang and Meng (2011) for solving SVM model selection problems. In this study, the modified PSO is utilized to choose the optimal value for SVM parameters. The modified PSO made through virtue of chaotic motion with sensitive count on initial conditions and ergodicity (CPSO) and the error of *k*-fold CV is utilized as the objective function of PSO. Wang and Meng use SVM with Gauss kernel (GSVM) and SVM with WSVM and, in order to use wavelet kernel, Marr wavelet is utilized to build a translation invariant wavelet kernel. The performance of GSVM and WSVM is compared against BPNN with four neurons in the input layer, ten neurons in the hidden layer, and one neuron in the output layer. The results show that WSVM has fast convergence speed and high generalization capability compared with GSVM; and compared with BPNN, SVM it has simple construction, fast convergence speed with high generalization capability.

Qiu et al., (2011) conducted a study on SVM parameters optimization by using Bare Bones Differential Evolution (BBDE). BBDE is a hybrid of the barebones PSO and DE. It deletes the monitor variables of PSO and substitutes the static DE monitor variables by dynamically alternating variables to generate a general parameter-free, self-adaptive, optimization algorithm. Qiu, Li, Zhang and Gu test their method on four UCI datasets. They run their algorithm 20 times and they compare their results with a grid algorithm. The results show that the testing accuracy is higher the grid search which indicates that the parameters chosen by BBDE and DE are better than the ones chosen by grid search. The testing accuracy of BBDE-SVM is higher than DE-SVM and BBDE-SVM is quicker than DE-SVM, this is because BBDE-SVM has fewer parameters to set than DE-SVM.

Model Parameters Selection for SVM Classification using Particle Swarm Optimization was proposed by Hric, Chmulik and Jarina (2011). In this work, the authors utilize a standard type PSO and each value of the particle is rounded to one decimal place where parameter range is the same as in the grid search model selection. Additional to PSO, GA that works with real parameters to tune SVM parameters is also implemented. They compare their work with a grid search algorithm. In order to evaluate their work, they examine their work on two datasets: a letter recognition database and speech dataset for speaker recognition processes. They utilize a five-fold CV. Both PSO and GA seek for the optimal value of C and RBF kernel SVM parameters in 20 iterations. The results showed that both PSO and GA are faster than grid search in model selection, both techniques present comparable results and have a capability to optimize more than two parameters. The results for the letter dataset show that PSO and GA obtain classification accuracy that is as good as grid search, while in case of the speaker dataset, the results show that PSO can obtain good parameters and obtain classification accuracy five times better than GA.

Liao, Yang and Ding (2011) produced a paper named “Approximate Parameter Tuning of Support Vector Machines” which considers a novel productivity model selection strategy for SVM through kernel matrix approximation. They examine the feasibility and efficiency of model selection on five UCI datasets and the results indicate that the sampling sizes decrease in a certain range, therefore, the alternations of test set accuracy do not need to be taken into consideration closely ignorable.

Also, they compare the 5-fold cross validation accuracy between accurate and approximate optimal models. For future work they suggest building a complete approximate model selection theory which has a close fitting approximation mistake range and can be directly or indirectly performed in the design of adaptive approximate model selection algorithms.

A novel study named “Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA)” proposed by Sartakhti, Zangooei and Mozafari (2011) was carried out. SA is used to find the optimal value for SVM parameters which are C and γ RBF kernel parameter. Firstly, SA parameters are initialized, as are random SVM parameters (C and γ); subsequently regarding these random SVM values, SA is used to seek best neighbours. k-fold CV is utilized to obtain CVs and compare the alternation combination of C and γ. If the combination is satisfied the algorithm will stop and construct the leaner classifiers for Pair-Wise Coupling (PWC) probability guesstimate, otherwise, it must tune them and continue. Apart from hepatitis disease diagnosis, SVM-SA is also applied on a two-spiral dataset and a chain link dataset. The results show that the SVM-SA approach can find very hopeful results in classifying. The authors suggest to expand their work to employ some feature selection techniques and other learning approaches will be utilized in order to maximize the precision of their approach.

GA and SVM was used for a model selection problem proposed by Samadzadegan, Soleymani and Abbaspour (2010). The authors show that the hybridization between GA and SVM can improve the classification accuracy and convergence speed when compared with classical grid search algorithm and SVM. In this study, the authors utilize GA to optimize *C* and RBF kernel function parameter encoded as real value chromosomes; the fitness value for each chromosome is computed with regard to the chromosome's classification accuracy obtained from multi-class decomposition (one-versus-one and one-versus-all). The implementation of GA-SVM was carried out on Java by expanding the LibSVM and JGAP package for implementing the GA part. In order to evaluate this study, the authors examine it on five datasets, three from UCI and two found in LibSVM. The authors suggest applying their study on a binary class SVM, on other kernel functions, and also on optimizing feature subset which are considered to be another problem of SVM.

A paper written by Zhang, Chen and He (2010) is presented, using an ACO and grid search to solve a SVM model selection problem. The author presents a novel ACO pheromone model and divides the ranges of RBF and *C* parameter into a number of grids and allows ants to select the best combination of RBF and *C* parameters. To examine the execution of this approach, the author uses five UCI datasets compared against grid search SVM, 5-fold CV, radius margin bound, span bound as well as with the method proposed by Adankon and Cheriet (2007). The results show that the proposed approach is feasible and efficient to optimize the SVM parameters and produce hopeful results in term of classification accuracy.

Zhou and Xu (2009) present a study named “A SVM Model Selection Method Based on Hybrid Genetic Algorithm and Empirical Error Minimization Criterion”. In the study, the authors integrate GA with a gradient descent method to build hybridized techniques to select the optimal value for RBF and *C* parameters. Their technique first selects the best chromosome as the initial solution, then utilizes a gradient descent approach to seek for optimal kernel parameters for the best chromosome as its iterative solution. After examining a new generation, the new best chromosome was taken from three nominees: the best solution in the past generation, the best solution in the new generation, and the iterative enhanced version of the past best solution. Zhou and Xu evaluate their approach on thirteen datasets and compare their approach with simple GA-SVM, 5-fold CV, Radius Margin Bound (R-M bound), span bound and Adankon’s approach; the results show that this approach finds better results compared with other approaches. Zhou and Xu suggest examining their approach on more real applications and applying it with other kernel function as well as using some heuristic mechanisms to speed up their approach.

Tang, Guo and Gao (2009) propose research called “Efficient Model Selection for Support Vector Machine with Gaussian Kernel Function”. The authors propose a two-phase heuristic parameter selection approach which is able to automatically set the Gaussian kernel parameter and the cost parameter in SVM. Firstly, a new Gaussian kernel parameter selection method is proposed. Subsequently, to choose the cost parameter *C* and Gaussian kernel parameter, the proposed method, integrated with a one-dimension cross validation seek mechanism, develops a complex parameters method. The authors evaluate their approach on eight real world datasets taken from UCI and compare their work with the standard grid search method which has high accuracy in parameter selection, a two stage Uniform Design (UD) schema which has computational efficiency, and also with Empirical Set (ES) method. The results show that the proposed approach finds the highest test accuracy of SVM in six datasets among the eight UCI datasets. The proposed approach reliably obtains a good parameter string for Gaussian kernel SVM in a small number of trail pairs. The authors suggest as future work to apply their work to other types of kernel function as well as to apply it on SVM regression.

A study presented by Fang and Bai (2009) titled “Share price prediction using wavelet transform and ant colony algorithm for parameters optimization in SVM” aims to optimize C and σ kernel function SVM parameters through utilizing ACO. 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 the local interval. Their results show a very promising hybrid SVM model for forecasting share price in terms of accuracy and generalization ability.

Zhou, Zhang and Bai (2008) present a study called “Client classification on credit risk using rough set theory and ACO-based support vector machine”. In their study they firstly discretize the continuous feature through the Average Linkage method and then compute t-tests for each feature. After that rough sets are utilized to select suitable feature subset and ACO is used to tune SVM parameters which are σ of Gaussian kernel and C parameter. In order to evaluate their work, the authors apply it on a Chinese bank credit risk classification problem and compare the result with Fisher, Probit, fix SVM parameters and PBNN with eight input neurons, seven hidden neurons, and one output neuron. The results show that their approach outperformed other approaches in terms of best classification accuracy, the lowest error of Type I (mean the first year before failure, failed firms are classified as non-failures), and Type II (means non-failed firms are classified as failures).

Zhang (2008) proposed a study titled “Evolutionary computation based automatic SVM model selection” aiming to optimize SVM parameters through utilizing evolutionary computation approach and using recollection, accuracy and mistake ratio as an optimization goal. The concept of constructing a kernel prototype is used which is then modified to the data group with the help of evolutionary computation approaches. The modification procedure is directed by the feedback information obtained from SVM execution. Both GA and PSO are used as evolutionary computation approaches to resolve optimization difficulty that occurs due to their robustness and global seeking capability. The results show that PSO produces better results compared with GA when it is used to optimize SVM parameters.

Saini, Aggarwal and Kumar (2010) use GA to optimize SVM variables. The regularization parameter *C* and kernel parameters are dynamically optimized through GA. SVM parameters are encoded into chromosomes and the GA operation including crossover and mutation are applied on these chromosomes to obtain the optimal value for SVM parameters. The authors use 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 conclude that their model supplies better predicting capability with sensible levels of accuracy and stability.

A grid-based ACO technique was introduced by Zhang et al., (2008) to select variables *C* and σ RBF kernel automatically for SVM instead of choosing variables randomly through human skill to minimize generalization mistakes and generalization execution may be enhanced concurrently. The ranges of SVM parameters are divided into many intervals and then ACO search for optimal combination of SVM parameters values to present to SVM. Their work provides high accuracy compared with other methods like an exhausting seeking approach with human experience, grid algorithm and cross validation approach. However, one dataset is used to evaluate the performance of the proposed technique. The authors suggest applying their approach on more datasets to examine the power of their work.

2.2.2 Approach in Solving Feature Subset Selection Problem

A recognized difficulty in classification particularly, and machine learning universally, is to obtain methods to minimize the dimensionality of the feature domain to prevent the danger of over fitting (Guyon et al., 2002) and, as mentioned earlier, SVM is an established optimization approach for data classification problems; additionally, it is widely used for classification in machine learning. SVM can avoid having repeated features. Contemporary datasets mostly involve large features, many of which may be noisy or non-related to the context of the analysis, so eliminating the irrelevant features can minimize SVM complexity, accelerate SVM convergence, and also improve its execution (Feki, Ishak & Feki, 2012; Tan et al., 2012; Hu et al., 2012; Xie & Wang, 2011; Nguyen & Torre, 2010; Dunbar et al., 2010; and Liu & Zhang, 2009). Researches related to Feature Selection (FS) constantly deals with the issue of choosing suitable input features to produce good classification results. Therefore, it has been noticed that successful application using datasets of large numbers of features, would be difficult to manipulate (Wang, 2012; Xiong & Wang, 2008; Shieh & Yang, 2008; Verleysen, Rossi & Francois, 2009; Jensen & Shen, 2004; and Partridge, Wang & Jones, 2001).

A FS algorithm can be seen as three groups, depending on how the classification model is built using machine learning and feature selection search, these are wrapper, filter, and embedded, sometimes called hybrid approaches (Atienza et al., 2012; Unler, Murat & Chinnam, 2011; Brogin & Slanzi, 2010; Moustakidis & Theocharis, 2010; Peng & Jiang, 2010; and Schaffernicht, Stephan & Grob, 2007;

and Weston et al., 2001). Filter and wrapper approaches may be further classified into five main methods: forward selection, backward elimination, forward/backward combination, random choice and, finally, instance based method (Basiri, Aghaee & Aghdam, 2008 and Kanan, Faez & Taheri, 2007).

A filtering method can be used in the preprocessing phase to minimize space dimensionality, over fitting and independency on the classifier. The main idea behind using a filtering approach is to look for related features and delete the non-related features through using statistical measurements (Atienza et al., 2012; Foithong, Pinngern & Attachoo, 2012; Lian, 2012; Unler, Murat & Chinnam, 2011; Peng, Wu Jiang, 2010; Brogin & Slanzi, 2010; Moustakidis & Theocharis, 2010; and Su & Yang, 2008). There are two steps in using a filters approach: In the first step criteria such as information, distance, dependence, or stability are used for feature selection without classifier; in the second step, with the selected features, a classifier would learn from the training data and be examined on the test data (Moustakidis & Theocharis, 2010; Brogin & Slanzi, 2010; and Su & Yang, 2008).

A wrapper approach searches through the area of all potential feature subset by applying the forecasted accuracy based on classification algorithms. These approaches are exhaustively calculated and rely on the classification algorithm, but they also consider feature dependencies (Atienza et al., 2012; Foithong, Pinngern & Attachoo, 2012; Lian, 2012; Unler, Murat & Chinnam, 2011; Peng, Wu & Jiang, 2010; Moustakidis & Theocharis, 2010; Shieh & Yang, 2008; and Liu & Zheng, 2006). A wrapper approach is seen as useful but is very slow to run because the learning approach is called repeatedly. Hence, wrappers do not perform well on large datasets that have many features (Atienza et al., 2012; Foithong, Pinngern & Attachoo, 2012; Lian, 2012; Unler, Murat & Chinnam, 2011; and Hall, 2000).

The quality of features and feature subset size are often used as the terminating condition in wrapper techniques (Moustakidis & Theocharis, 2010). The wrapper approach involves two steps: choosing the good feature subset by utilizing the accuracy of the classifier (on the training data) as a condition to learn and understand; and the second step, a learning classifier is taken from the training data with the best feature subset, and would be examined on the test data (Su & Yang, 2008).

If a comparison is made between filter and wrapper approaches, it is obvious that the filter is able to perform at a much faster rate compared to the wrapper. Therefore, it can be used on large datasets with many features. The generic existence allows them to work with any learner. This is not the same when it comes to the wrapper because the wrapper needs to be re-executed when changing from one learning approach to another. The filter technique is more efficient than the wrapper technique based on calculation. However, a major disadvantage is that an optimal selection of features may not be treated separately from the inductive and representational influences of learning approach, which is used to build the classifier. The wrapper technique on the other hand, includes extra calculation which measures nominated feature subset by running a provided learning approach on the database using each feature subset under restriction (Atienza et al., 2012; Foithong, Pinngern & Attachoo, 2012; Lian, 2012; Unler, Murat & Chinnam, 2011; Wang et al., 2010; and Oliveira et al., 2003).

In the hybrid or embedded approach, the advantage of both feature and wrapper technique is utilized. The optimal feature subset is constructed into the classifier when it is being built. The interaction with classification prototype enhances the calculation complexity and considers feature dependencies (Lian, 2012; Brogin & Slanzi, 2010; and Peng, Wu & Jiang, 2010). In the literature, there much research that deals with using SVM and feature selection, some of these are as follow.

A recent study presented by Hidalgo-Muňoz et al. (2013) titled “Application of SVM-RFE on EEG signals for detecting the most relevant scalp regions linked to affective valence processing” tries to select feature subset for SVM classifiers through using Recursive Feature Elimination (RFE). The authors evaluate their study on brain oscillation dynamics from Electroencephalography (EEG) signals throughout the scalp. EEG signals are recorded during visualization of selected pictures belonging to an International Affective Picture System (IAPS). In order to transfer EEG signal from input domain to topography-time–frequency feature domain, A Morlet wavelet filter is used and SVM-RFE is performed for detecting Scalp Spectral Dynamics Of Interest (SSDOI) in this feature space. In order to evaluate the proposed study, 26 healthy females are included in this study aged between 18-62 years with normal or corrected to normal vision, none of whom had a history of severe medical treatment, either psychological or neurological disorders. Each image is presented three times, the different conditions are counterbalanced along the experiment and trial order is pseudo-random. The classification accuracy of SVM is computed using a LOOCV strategy. The experiment results are compared with the t-test statistical approach utilized to select feature subset and show that SVM-RFE outperforms t-test. The authors of this study do not mention what type of kernel function they use and also they do not mention how they tune SVM parameters.

A recent research conduct by Garde et al. (2013) called “SVM-based feature selection to optimize sensitivity-specificity balance applied to weaning” aims to optimize SVM’s feature subset selection through utilizing a new metric named balance index (B) which measures the alternation between miss-classified data within each data class. RBF kernel function is used and SVM parameters are tuned using grid search algorithm. 10-fold cross validation is used to compute the average classification accuracy of the proposed approach and the proposed approach is evaluated to classify 154 patients’ weaning trials from mechanical ventilation: 94 patients with successful trials were able to maintain spontaneous breathing after 48 hours and 39 patients failed to maintain spontaneous breathing and were reconnected to mechanical ventilation after 30 minutes. The proposed approach compared with quadratic discriminant, classification trees, naive Bayes classifier, SVM and k-NN classifier. The experimental results show that the proposed approach gives good results.

A study presented by Lin et al. (2012) aims to improve SVM performance by choosing suitable features to classify liver disease metabolome datasets. The authors use SVM-RFE as a famous embedded approach for feature selection and propose a Mutual Information (MI)-SVM-RFE approach which initially filters out the noisy features via utilizing Artificial Variables and MI (AV-MI) then uses SVM-RFE to compute the weight of each feature regarding the support vectors of the present learning model and choose the best feature subset. The authors use LibSVM for SVM implementation and C++ for implementing MI-SVM-RFE and SVM-RFE. In order to choose the necessary features regarding liver disease, the authors portion their research into five sub-objects: (1) a binary problem to classify normal set and the disease set, (2) three binary problems to classify every two of Chronic Hepatitis B (CHB), CIRrhosis (CIR), and HepatoCellular Carcinoma (HCC), (3) three class problem to classify CHB, CIR, and HCC simultaneously and utilize a ten-fold cross validation to evaluate the performance of MI-SVM-RFE as well as being compared with the original SVM-RFE. The results show that MI-SVM-RFE outperforms the original SVM-RFE in terms of classification accuracy.

Atienza et al. (2012) present a novel embedded features selection method based on SVM and Bootstrap Resampling (BR) approaches for Ventricular Fibrillation (VF) detection. A group of temporal, spectral, and time-frequency variables selected from the ElectroCardioGram (ECG) signal databases is utilized as the input domain to SVM. The pertinence of input parameters is examined via comparing the detection execution of the complete group of input parameters and reduced subsets by using a corresponding a group strategy of filter approaches, which are, accounting for second order approaches (correlation criterion), mutual information approaches (difference and quotient schema), and maximum separability Fisher criterion. This comparison is accomplished regarding to a nonparametric statistical test based on BR. The authors compare their approach with SVM-RFE and apply it on toy set and ECG signals databases. A five-fold cross validation technique is used to set the SVM parameters. The results demonstrate good application of SVM-BR for detecting VF. The authors suggested extending their approach to improve VF-Ventricular Tachycardia (VT) discrimination and analysing possible discriminatory ECG variables to develop real-time VF detectors.

Another study, presented by Feki, Ishak and Feki (2012), proposes a wrapper features selection approach using Gaussian marginal densities for a Bayesian prototype and ranking score process to examine the contribution of each feature of the prototype which is taken from a multi class SVM. In this study, the authors use OAO and OAA approaches for a multi class SVM and utilize Hamming decoding, Loss decoding and Reordering Adaptive Directed Acyclic Graph (RADAG) to gathers the OAA and OAO hyperplane while for the Gaussian Bayes prototype, the authors utilize two parametrics under both dependence and independence assumptions. The authors evaluate their approach on a toy data and on ten alternation commercial Tunisian banks during a period from 2000 to 2006 to classify seventy instances into three classes of risk regarding the level of Nonperforming Loans (NPLs) utilizing twenty-eight input features. The results show that the Bayesian prototype under independence assumption accomplishes the best result while the loss function decoding method outperforms all other techniques. The authors suggest applying their approach in other studies when the number of features is high compared with the number of instances as well as applying their approach in many other business studies like prediction of issuer credit rating.

A study presented by Xie and Wang (2011) proposes a novel feature selection approach called Improved F-score and Sequential Forward Search (IFSFS) to obtain the optimal feature subset in the procedure of feature selection where F-score is considered as a filter technique while Sequential Forward Search (SFS) is a wrapper technique. In this study, the SVM parameters which are C and γ of RBF kernel function are obtained through a grid search method utilizing four alternation training-testing portions to obtain optimal parameter values. To examine the success of this study, the authors apply their technique to the diagnosis of erythemato-squamous diseases; they build 34 models with alternative feature subset and compare them with other studies; the results show that their approach outperforms others. The authors suggest applying their improved F-score to other fields as a filter technique for discrimination measure.

Liu et al., (2011) undertook a study for wrapper feature subset selection by using SVM with RBF kernel built on RFE (SVM-RBF-RFE). This study extends nonlinear RBF into Maclaurin series then calculates the weight vector from the series regarding the contribution for classification via each feature. After that, they utilize square weight for each feature as a score rank. To evaluate the success of this approach, the authors compare their approach with SVM-RFE and information gained from WEKA which they use to evaluate the feature subset chosen by their approach, SVM with linear kernel (SVML), SVM with RBF kernel, 5-NN and with 10-NN as well as applying it to three UCI datasets and three sets of microarray data using 10-fold cross validation. The results show that a SVM-RBF-RFE approach is very competitive and can identify the most important features; the authors conclude that their approach is very successful on feature selection for SVM with RBF kernel function.

A study called "mr^2 PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification" was conducted by Unler, Murat and Chinnam (2011). They propose a schema between filter and wrapper feature subset selection technique built on PSO for SVM classification. MI is used as a filter technique, while the wrapper technique adapts a discrete PSO approach. In order to evaluate their approach, the authors utilize six UCI datasets, 2-fold cross validation and set SVM parameters which are C to 100 and RBF kernel parameter σ to 2 in addition to utilizing a OAA technique for multi-class SVM. mr^2 PSO is compared with hybrid filter-wrapper technique based on GA, and PSO based wrapper technique in terms of classification accuracy. The results show that the proposed approach is competitive in terms of classification accuracy. The author suggests searching for best parameters' values to enhance classification accuracy, using 10-fold cross validation rather than 2-fold cross validation, applying back tracking at the end of feature selection and arbitrarily re-evaluating the features in the subset, and finally comparing it with features extraction approaches.

A paper written by Garbarine et al. (2011) utilizes three filter techniques based on information theory for features selection that enhance SVM genome classification. These techniques are: (1) a method that gathers Kullback-Leibler Mutual Information and distance information, (2) a text mining method, Term Frequency-Inverse Document Frequency (TF-IDF), (3) minimum Redundancy-Maximum-Relevance (mRMR). These techniques are compared against each other in terms of enhanced SVM classification accuracy and also with non-information theoretic features techniques and applied on 100 bacterial genomes. The authors use N-mers terms which are utilized to construct frequency profiles (how often each word happens in a present sequence) which are then utilized to construct classification models. The results show that N=6 produces better results than both N=3 and N=9 demonstrating that there is a balance between size of features set and classification accuracy. Additionally, TF-IDF performs better on N=9 level for fine resolutions. However, mRMR utilizing N=6 executes well in most taxonomic levels (strain, species, genus, family, order, and phyla) especially on a phyla level.

A paper titled “hybrid feature selection algorithm based on dynamic weighted ant colony algorithm” introduced by Xiong, Wang and Lin (2010) attempts to enhance SVM performance on classifying eight UCI datasets through utilizing dynamic weighted ACO to select feature subset. The reason for utilizing dynamic weighted ACO was to preserve a good tread-off between the convergence rate and the stagnant pheromone which was calculated and updated based on SVM outputs. The authors implemented their approach in Matlab for programming dynamic weighted ACO and LibSVM for implemented SVM sections. All features are discretized and normalized before carrying out the experiments. The authors compare their results with other approaches and the results show that their approach is the best in terms of classification accuracy and number of selected features. The authors suggest proposing other heuristic functions and pheromone updating mechanisms for the particular field datasets.

A study introduced by Abd-Alsabour and Randall (2010) aims to improve the performance of SVM on classified ten binary datasets taken from statistical and UCI datasets through integrating Ant Colony System (ACS) via a wrapper approach to selecting limited numbers of features. The authors implement their approach in C-classification SVM of package e1071 of the R language and WEKA machine learning tool. SVM performance is computed through utilizing a 5-fold cross validation technique and with two directions of experiments which are with and without selected feature subset. The results are compared with other approaches and demonstrate that the proposed approach, SVM-ACS, was the best compared with others in terms of classification accuracy and size of feature subset. The authors suggest making an investigation into tuning the parameters of their approach also utilizing an ensemble features selection strategy to enhance the overall performance of the proposed approach. Finally, the authors also suggest utilizing high performance calculation approaches such as parallel multi category SVM.

Another study, introduced by Li and Chen (2010) called “Weed identification based on shape features and ant colony optimization algorithm”, tries to minimize classification errors of SVM through selecting suitable feature subset. Two optimization techniques, ACO and GA, are utilized to achieve this goal. These two optimization techniques are integrated with SVM through a wrapper approach. The features are initially extracted from the weed leaves; after that ACO and GA are utilized to choose the optimal feature subset to introduce to SVM and; lastly, this technique is performed on a lab plant image database of a cotton area. The experiment results show that when ACO is integrated with SVM it is better than when GA is integrated with SVM. The authors suggest improvement of high performance approaches built on multi features combination to distinguish between plants and weeds containing substantial types of weeds and crops.

A study conducted by Wang et al., (2010) to select features for ℓ_1-norm soft margin SVM for binary classification set the parameters of the Gaussian Automatic Relevance Determination (ARD) kernel through optimizing kernel polarization; after that the whole features are ranked in descending order of necessity, so if any input features badly in the classification problem its scaling factor is small and potentially can be eliminated without impact on the classification execution. This is done by maximizing kernel polarization with the Gaussian ARD kernels. The authors utilize a classical correlation coefficients approach to evaluate the execution of their work and the optimal values of the SVM parameters which are C and γ RBF kernel function are set a through grid search 10-fold cross validation approach. The authors examine their approach on four UCI datasets and suggest applying their approach on regression and multi-class SVM as well as appending the theoretical verifications of fine execution of their study and comparing it with other techniques.

A study presented by Moustakidis and Theocharis (2010) titled “SVM-FuzCoC: A novel SVM-based feature selection method using a fuzzy complementary criterion” enhances SVM performance by selecting the suitable feature subset by using a forward filter technique. The authors utilize OAA to decomposed multi-class SVM and train a group of binary K-SVM classifiers on each feature to obtain fuzzy membership of each pattern to its class. The entire data is portioned into training-testing data and Fuzzy Complementary of Criterion (SVM-FuzCoC) performed on training data, k-NN classifier is utilized as an evaluation approach. The authors apply their approach on twelve real-world classification problems and group it into three with regard to their calculation difficulties and compare it with eleven well known feature selection approaches in the literature. The results show the success of SVM-FuzCoC in terms of accuracy, dimensional minimization, and calculation effort by comprehensive experimental setting.

Akay (2009) proposes an approach using F-score to select relevant feature subset for SVM to classify a Wisconsin breast cancer dataset from the UCI repository. The F-score is computed for each feature in the training dataset and sorted in descending order; a number of features are then selected that have the highest F-scores to generate feature subset, followed by a grid search with 10-fold cross validation in order to obtain the optimal values for C and γ RBF kernel function and train the selected feature subset with these values of C and γ RBF kernel function and the SVM classifier model is found and utilized to classify test dataset. The author compares his approach with other approaches and the results show that his approach produces the highest classification accuracy. As an extension to his study, the author suggests to apply his approach to more datasets.

Another study “A new feature selection method on classification of medical data sets: kernel F-score feature selection” was conducted by Polat and Gunes (2009) to select feature subset for Least Square SVM (LS-SVM). Their approach proposes a novel feature selection named Kernel F-score Feature Selection (KFFS) which firstly maps the input features to kernel domain through using Linear (Lin) or RBF kernel functions then applies F-score on this domain, followed by the mean value of computed F-score as a threshold to decide which feature will be chosen; if the feature’s F-score value is higher than the mean value, this feature will be chosen, otherwise the feature is discarded. LS-SVM and Levenberg-Marquart Artificial Neural Network (LMANN) are used in this study as a classifiers. In order to evaluate the performance of their approach, the authors apply it on three medical UCI datasets; and they fix the value for σ of RBF kernel function to 100 and C value is computed through a trial and error approach and the best value is 1000. The results show that KFFS generates very successful results compared to F-score features selection. The authors suggest applying their approach on more medical datasets as well as replacing the F-score approach with other approaches.

Maldonado and Weber (2009) present a study titled “A wrapper method for feature selection using Support Vector Machines”. In their study, they propose a wrapper technique to select features based on sequential backward selection. The authors utilize a hold-out approach as a measure to judge which feature to delete at each iteration instead of a measure to build simply on one dataset, gradient-based measure or Fisher correlation score. The authors start their approach by solving a model selection problem using 10-fold cross validation, they apply SVM without features selection in direction to determine the best kernel function; the Gaussian kernel function is the best when compared with the polynomial kernel function. In order to evaluate their approach, the authors apply the study on four datasets and compares it with RFE-SVM, concave feature selection as well as performing a Fisher criterion score filter approach; the results show that their approach was strong according to selected features and improved classification accuracy. The authors suggest applying their approach on other kernel functions and on support vector regression also to apply their approach on weighted SVM.

Another study was presented by Mesleh and Kanaan (2008) titled “Support vector machine text classification system: using ant colony optimization based feature subset selection”, to utilize a combining of Ant System (AS) and Elitist Ant System

(EAS) to select a small number of feature to present to SVM to classify 1445 online Arabic newspaper archives, taken from Al-Jazera, Al-Nahar, Al-hayat, Al-Ahram, and Al-Dostor, into nine categories. In this paper, Chi-square statistic is computed for each feature and utilized as heuristic information for ACO to guide ants to select the best features. Apart from Chi-square, the authors implement five traditional feature subset selection techniques including Information Gain (IG), Ng, Goh, and Low (NGL), Galavotti, Sebastiani, and Simi (GSS), MI and Odds Ratio (OR). The results show that when using Chi-square as heuristic information the result is the best compared to other techniques. The authors suggest comparing their work with others like Max-Min Ant Colony algorithm and also suggest applying their approach on larger datasets. Another suggestion from the authors is to try other statistic approaches as heuristic information for ACO.

2.2.3 Approach in Simultaneously Solving Model Selection and Feature Subset Selection Problem

Twelve similar works (Zhang & Jiao, 2005; Huang & Wang, 2006; Pham et al., 2007; Lin et al., 2008; Huang & Dun, 2008; Lin et al., 2008; Ding & Li, 2009; Lin & Chien, 2009; Huang, 2009; Zhao et al., 2011; Vieira et al., 2013; and Sarafrazi & Pour, 2013) suggest using hybrid systems to enhance classification accuracy by using limited, suitable feature subset. All eleven works optimize feature subset and SVM parameters which are C and γ RBF kernel variable simultaneously. Ultimately, SVM is used to measure the quality of the solution for all hybrid systems. However, what differs is what the hybrid system is based on.

Zhang and Jiao (2005) propose utilizing a hybrid system based on Immune Clonal Algorithm (ICA) and SVM. They utilize ICA to simultaneously choose feature subset and SVM parameters which are represented in the encoded antibody population. Huang and Wang (2006) and Zhao et al., (2011) propose the use of a hybrid system which is based on GA and SVM. They would use GA to select suitable features simultaneously with optimized SVM parameters which are represented in the encoded chromosomes. Huang and Dun (2008), Lin et al., (2008) and Vieira et al. (2013), on the other hand, choose to use a hybrid system which is based on PSO and SVM. In Huang and Dun (2008), they mix discrete PSO with continuous valued PSO to simultaneously select suitable features and optimize SVM parameters, Lin et al., (2008) use only the discrete version of PSO to optimize feature subset selection and SVM parameters, while Vieira et al. (2013) use Modified Binary Particle Swarm Optimization (MBPSO) to simultaneously select suitable features and optimize SVM parameters. Conversely, Lin et al., (2008) use SA to simultaneously optimize model selection and feature subset selection. They use a continuous Hide-and-Seek SA to optimize the continuous values of SVM parameters and they represent the features as discrete values. The authors, for this paper, not explain how they can handle the discrete values for features while they use the continuous version of SA to optimize SVM parameters.

Also, Pham et al., (2007) propose utilizing a Bees algorithm to simultaneously choose the best combination of feature subset and SVM parameters values for the process of classifying faults in wood layer pieces. Huang (2009) decides to use a hybrid system which is based on ACO and SVM. He uses classical ACO to simultaneously select suitable features and optimize SVM parameters. Ding and Li (2009) propose using a hybrid system which is based on a Clonal Selection Algorithm (CSA) and SVM. They would then use CSA to select suitable features simultaneously and optimize SVM parameters, while Lin and Chien (2009) propose the use of a hybrid system which is based on Cat Swarm Optimization (CSO) and SVM. They would then use CSO to select suitable features simultaneously with optimized SVM parameters. Finally, Sarafrazi and Pour (2013) utilize two versions of Gravitational Search Algorithm (GSA) which are Real value GSA (RGSA) to optimize the real value of SVM parameters and Binary (discrete) value GSA (BGSA) to select feature subset. GSA is considered as a swarm based metaheuristic seek approach built on the law of gravity and motion and it is derived from the Newtonian law of universal gravitation. Sarafrazi and Pour (2013) apply their approach only on a binary class classification problem; they did not test multi class classification problems.

All the above works - except Huang and Dun (2008) who apply their work on distributed parallel architecture - examine their approaches on different UCI datasets. From the view point of Huang and Wang (2006), Huang and Dun (2008), and Huang (2009), they consider their works to be a novel and first studies that combine GA, PSO, and ACO with SVM to simultaneously optimize both feature subset selection and model selection.

The general conclusions of all twelve works give good results in the concept of classification accuracy and a few numbers of selected features while the general suggestions of all these works are as follows: Vieira et al. (2013) implement their method with other medical databases; Zhang and Jiao (2005) implement their method on the territory classification in remote sensing images; Huang and Wang (2006) and Huang (2009) suggest applying their work on Support Vector Regression (SVR), because SVR accuracy counts mainly on SVR parameters and selected feature subset; Huang and Wang (2006), Huang (2009), Ding and Lie (2009), and Lin et al., (2008) suggest using other types of kernel function not simply RBF. Lin and Chien (2009), Ding and Lie (2009), Lin et al., (2008), and Zhao et al., (2011) suggest applying their works on other real world problems to test and expand their works. And finally Huang (2009) suggests using continuous ACO to optimize the continuous value of SVM parameters.

A paper titled “Simultaneous feature selection and classification using kernel-penalized Support Vector Machine” is presented by Maldonado, Weber and Basak (2011). The main idea of their approach is to simultaneously optimize kernel function and choose optimal feature subset for classification. The authors utilize dual formula SVM involving a penalization function based on 0-norm approximation and modifying Gaussian kernel utilizing gradient descent approximation for kernel optimization and feature deletion. 10-fold cross validation is utilized to calculate the classification accuracy and compare the best result of the model for linear, polynomial and Gaussian kernel function using standard deviation; then the best values of kernel parameter are utilized as kernel-penalized SVM (kp-SVM) input. kp-SVM was compared with SVM-RFE, FSV in addition to applying a Fisher score filter approach and the results show that kp-SVM is the best. The authors suggest applying kp-SVM to multi-class SVM as well as on regression problems. Additionally, the authors also suggest the use of other kernel functions or weighted SVM.

2.3 Ant Colony Optimization

ACO is a group of algorithms to deal with optimization problems that is motivated by means of the pheromone trail put down and following the mannerisms of some ant kinds. While exploring, ants set down on the land a chemical material, named a pheromone that interests fellow nest colleagues. The placing of a pheromone trail and following manner of the ants creates a positive feedback procedure whereby trails of high densities of pheromones become increasingly attractive the further ants follow them. As an outcome, at any time two routes to the exact source of food are detected, the colony is further able to choose the shortest route because ants will cross it quickly and thus it will have a higher pheromone density than a longer trail (Dorigo et al., 2011; Twomey et al., 2010; and Birattari, Pellegrini & Dorigo, 2007).

ACO algorithms use a procedure similar to the one that permits colonies of real ants to obtain the shortest pathways. In ACO, (artificial) ants build nominee solutions to the problem under consideration. Their solution building is stochastically influenced through (artificial) pheromone trails, which are given in the shape of numerical information that is assigned by means of suitable defined solution components, and through heuristic information established on the input data of the problem being solved. A fundamental feature of an ACO algorithm is the utilization of a positive feedback loop applied via iterative updates of the artificial pheromone trails that are a function of the ants' seek knowledge; this feedback loop aims to influence the colony in the direction of, on the whole, hopeful solutions.

The ACO metaheuristic is a high-level algorithmic framework for implementing the previous concepts to the approximate solution of optimization problems. When implemented to a particular optimization problem, this ACO framework requires to be concretized through taking into account the details of the problem under consideration and probably through appending extra techniques, for example, as a problem-specific solution enhancement process (Dorigo et al., 2011).

The basic idea, broadly outlined by the behavior of actual ants, is that of a parallel seek through the foundation of many useful, calculating possibilities on local problem data and on memory construction involving information on the quality of past results obtained. The cooperative manner of the interaction of the alternative seek possibilities proves successful in solving problems. A group of calculations simultaneous and asynchronous agents (a colony of ants) walks the problem to solve. The ants walk by implementing a stochastic local decision policy built on two parameters, named trail and attractiveness. During the walking, each ant increasingly builds a solution to the problem. When an ant finishes a solution, or whilst building the process, the ant judges the solution and updates the trail value on the pieces used in its solution. This pheromone information will guide the seeking of the future ants.

Two important questions will appear in the foraging manner of ants: How does an ant know where to move? Who directs the ant? No-one directs the ant in its path to food origin. The grouping behavior of the ants is self-organized which means that the stigmergic interactions (build on local information) between ants are responsible for the global manner of the system. In other words, self-organization and stigmergic groups are responsible for the complex manner of an ant colony.

The first variant of ACO is Ant System (AS) (Colorni, Dorigo & Maniezzo, 1991) where the pheromone trail is updated only after all ants have constructed their solutions and the pheromone quantity deposit by each ant is calculated based on the solution quality. The first improvement on the ant system is called the Elitist strategy for Ant System (EAS) (Dorigo & Stützle, 2004). The improvement was done by providing strong additional reinforcement to the arcs belonging to the best tour found since the start of the algorithm. Rank-Based Ant System (AS_{rank}) is another improvement over ant system introduced by Bullnheimer, Hartl & Strauss (1999). In $AS_{rank,}$ each ant deposits an amount of pheromone that decreases with its rank. This is similar to EAS, where the best-so-far ant always deposits the largest amount of pheromone.

Max-Min Ant System (MMAS) proposed by Stützle & Hoos (1997) has four improvements over AS algorithm. First, the best-so-far solution during the execution is exploited by allowing only one ant to update the pheromone trail after iteration. The second improvement is the implementation of the limit range of pheromone trail values to the interval [τ_{min}, τ_{max}]. Third, the pheromone trails is initialized to τ_{max} in order to achieve higher exploration of solution at the start of the algorithm. Finally, in case of stagnation or no improved solution is generated for a specific number of iteration, MMAS will reinitialize to τ_{max}. MMAS algorithm achieves better performance than AS algorithm because of the modification over AS structure. Ant Colony System (ACS) is then introduced by Dorigo & Gambardella (1997) to improve the performance of AS in solving travelling salesman problem. In ACS algorithm, ants apply exploitation and exploration mechanisms when they select the next node to move to. ACS also applies local pheromone updates and global pheromone updates to direct the search for the next iteration. The global update is calculated based on the quality of the best solution so far while the local update applies the evaporation concept. All the above variants of ACO implement single colony of ants. However, multiple ant colonies can also be used to implement the variants of ACO (Pellegrini et al. 2007; Sim & Sun, 2003; and Al-Janabi, 2010).

The two main elements that make ACO successful when solving problems are that it utilizes of a group of priori information (heuristics) regarding the quality of nomine solutions (greedy strategy) and posteriori information (pheromone) about the reliability of the past solutions (positive or autocatalytic process). In other words, ACO does not simply utilize heuristics to select a solution from a group of nomine solutions but additionally utilizes the accumulated experiences concerning obtaining good solutions in the past iterations and directs the search in the neighborhood of these good solutions which are more likely to guide to enhancements and well solutions. The previous route looks logical since much research studies the properties of some good defined optimization problems, where it appears there is a relation between the solution quality and the distance from optimal solution. In other words, the neighborhood of the optimal solution usually involves good solutions and the quality of these solutions minimizes when they search far from the optimal solution (Al-Janabi, 2010).

2.3.1 Ant Colony Optimization for Feature Subset Selection

Feature subset selection is considered as a problem which needs to be optimized. In order to use ACO to solve this problem, there is a need to re-express this into a proper ACO problem. ACO needs a problem to be expressed as a graph. Vertices (nodes) represent features, with the edges between them representing the selection of the next feature. This is done by each ant moving through the graph with as small a number of vertices are visited that meet the termination condition.

Vertices are totally connected to permit any feature to be chosen next. All ants will have an alternative number of features to be chosen; therefore, it is not necessary for ant to visit all features in the seek domain as they move from one feature to another.

An ant will halt visiting the feature domain if it reaches a pre-defined chosen feature subset size or any other condition.

On the foundation of this re-expression of the graph representation, the moving laws and pheromone modification law of established ACO algorithms can be performed with minimum changes. In this situation, pheromone and heuristic value are not linked, but each feature has its own heuristic value and pheromone.

In order to build a solution, the procedure starts by creating an arbitrary number of ants which are then laid arbitrarily on the graph, that is, each ant begins with one arbitrary feature. From these initial places, the ants move vertices probabilistically until a moving condition is met. The resulting subsets are grouped and then evaluated. If the optimal subset has been obtained or the algorithm has run a specific number of paths, then the procedure stops and outputs the optimal feature subset. If none of these criteria is met, the procedure iterates once more (Al-Ani, 2005 and Rasmy et al., 2012). In the literature, much research deals with using ACO to select feature subset, some of these are as follows.

A recent research proposed by Li et al. (2013) named “An Ant Colony Optimization Based Dimension Reduction Method for High-Dimensional Datasets” aims to select feature subset for a SVM classifier with linear kernel function through proposing a method named Ant Colony Optimization-Selection (ACO-S) consisting of two phases: first, a modified ant system is utilized to filter features while an improved ant colony system is utilized to select features in the next phase. Additionally, the authors use two fuzzy logic controllers: one to dynamically adjust the number of ants in the ant system and the other to set the value for the q_0 parameter in the ant colony system. Also, F-score is used as heuristic information for ACO and the average classification accuracy is computed through utilizing 10-fold cross validation. Many values for SVM parameters were tested and one that gives the best result was selected. In order to evaluate the proposed approach, five microarray datasets are used and the experiment carried out in two directions: first the proposed approach compares GA based, ACO based, PSO based and SA based methods. The experiment results show that the proposed approach is better than other approaches while the second direction of experiment is to evaluate the performance of ACO-S with different classifiers additional to SVM. The experimental results show that ACO-S based SVM classifier outperforms other classifiers.

Another recent study introduced by Chen et al. (2013) entitled “Efficient ant colony optimization for image feature selection” aims to select feature subset using ACO. The main difference point between this study and other exiting studies that use ACO to select feature subset is that the ants in this study traverse with only O(2n) arc on a direct graph and not $O(n^2)$ arc like existing studies that use ACO to select feature subset where n is the number of features. Classification accuracy and size of feature subset are used as heuristic information guidance. After producing feature subset, SVM classifier is used for classification purposes and a cross validation method with ten-fold and five-fold is used to evaluate the average classification accuracy of SVM;

however, the authors do not mentioned how they tune SVM parameters. In order to evaluate the proposed approach, two large image databases and 15 non-image UCI datasets are used and the experimental results show that the proposed approach obtains high classification accuracy with small size of feature subset compared with other existing approaches.

Huang et al., (2012) undertook a study to improve classification performance of a BPNN. They use ACO to a solve features selection problem utilizing heuristic information calculated through a mRMR criterion. A Back Propagation (BP) model had twenty neurons in hidden layers with a tan-sigmoid activation function and output layer with linear activation function. The authors evaluate their approach in a classified hand motion surface of EMG signals on ten subjects' datasets with eight upper limb motions. Two features groups which are time domain features gathered with auto regression prototype coefficient and Wavelet Transform (WT) features are extracted from EMG signals. Their approach is compared with PCA and the results show that their approach produces higher classification accuracy. The authors suggest applying their approach in other physiological signals, alternative variant of ACO or other swarm intelligence as well as applying ACO to concurrently optimized BPNN structures.

Another hybrid study named "A new hybrid ant colony optimization algorithm for feature selection" was conducted by Kabir, Shahjahan and Murase (2012) that hybridized ACO and a BPNN for pattern classification. In their study Ant Colony

Optimization for Feature Selection (ACOFS), the authors utilize a filter and wrapper approach with bounded schema feature subset size to assist ACO to choose feature subset with minimum size and, in regard to this, ACOFS update classical pheromone update and heuristic information measurement rules. A constructive technique was used to build BPNN models automatically. The authors tested their approach on nine benchmark UCI datasets and compared against seven famous features selection approaches. The results show that ACOFS was the best in terms of classification accuracy with a small size of feature subset selection.

Another hybrid study, conducted by Rasmy et al., (2012) titled "A hybridized approach for feature selection using ant colony optimization and ant-miner for classification", attempts to improve Ant-Miner performance by hybridizing with ACO to choose suitable feature subset through a wrapper approach. In this study, five variants of ACO: AS, EAS, Rank-based Ant System (AS_{rank}), Max-Min Ant System (MMAS), and ACS are used. The pheromone is updated based on the classification accuracy and heuristic information and the transition probability to choose a solution track for each ant is computed based on the pheromone table and size of feature subset. The authors implement their approach with C programming language and use a myra-3.5.0 package for the Ant-Miner part. Cross validation is utilized to evaluate their work and applied on two groups of datasets taken from statistical and five UCI datasets. The results show that the proposed approach is better than others in terms of classification accuracy and size of feature subset. The authors suggest making more investigation on the parameters values of their work.

A study, presented by Chen, Liu and Xiong (2011) called “A novel feature selection method for affective recognition based on pulse signal”, aims to classify face emotion into six emotions which are happiness, surprise, disgust, grief, anger, and fear with high classification and few numbers of features through combined Max-Min ant colony algorithm and correlation analysis. Initially, Sequential Backward Selection (SBS) is utilized to select features, and then linear correlation coefficient is used to calculate the correlation degrees between features and those with high scores are eliminated. Finally, the features are chosen through a MMAS. The experiment results show that their approach is an effective and affective recognition technique using fixed and effective feature subset selection from original features.

A study based on ACO and BPNN, presented by Karnan and Akila (2010), is applied in a security domain to classify a keystroke dynamics typing pattern. 100 samples of digraph, duration, latency and their combination of each user keystroke for 27 users were considered as a feature and extracted in the extraction stage within the period of one week as well as the mean, median, and standard deviation being computed for each feature. The execution of their approach is evaluated by the overall classification mistake and accuracy. The authors compared their approach with PSO and GA as an optimization approach used to select feature subset; and the results show that when ACO is used to select the feature, the performance of ACO is the best in terms of classification accuracy and number of features selected.

Research conducted on selecting feature subset using ACO called “Edge and characteristic subset selection in image using ACO” by Venkatesan and Karnan (2010) attempts to maximize classification accuracy of ANNs through using ACO with two local search algorithms, local significance of features and overall execution of feature subset, to select the least numbers of feature subset to classify 56, 456 segments into six categories including vowel, fricative, glide, nasal, silence, and stop. MSE of trained ANNs with 2,000 arbitrarily selected segments is utilized as an objective function. Three groups of features are extracted from each frame of speech which are 16 log Mel-Filter Bank (MFB), 12 Linear Predictive Reflection (LPR) coefficient, and 10 wavelet energy bands. The authors compare their approach with SFS and GA. The result shows that MFB achieves the best execution compared with LPR and wavelet energy bands; however, it utilizes more features. Also, the results show that both ACO and GA achieve comparable execution to MFB utilizing like numbers of features with GA being slightly better than ACO, while SFS achieves a good execution when choosing fewer numbers of features; however, its execution is worse as the preferred number of features increases. The authors suggest applying their work on other classification problems.

A study introduced by Deriche (2009) aims to classify 16 log MFB, 12 linear LPR, and 10 wavelet energy bands speech segments into six categories including vowel, fricative, glide, nasal, silence, and stop. The author uses filter approaches including fisher criterion, mutual information based feature selection and mutual information evaluation function to compute the significance of features and then utilizes these scores as heuristic information for ACO to guide ant to select best feature subset and to predict the overall execution of subsets and local significant features. Beside this filter approach, a randomly chosen approach is also utilized to select feature subset. The author, unfortunately, did not mention anything about the classifier that used. The results show that the filter approaches are better than those randomly chosen when the number of selected features was less than 50, but in case of greater than 50 features, randomly chosen performance became close to other approaches. The author suggests investigating more on the effect of this filter approach when choosing large numbers of features.

Wei and Yuan (2009) propose a study to improve classification performance of k-NN classifier by selecting suitable feature subset through utilizing ACO. In their approach, they use wavelet transform to extract features from ECG signals and then utilized AS and ACS to select feature subset to present to the k-NN classifier to classify emotions of joy and sadness in 150 Southwest University undergraduate students aged from 19 to 25. The data was acquisitioned by high definition camera for about 20 minutes and the original ECG signals were gathered through a multi-channel physical acquisition device using SuperLab software to record the physiological data. The results show that ACS is better than AS in terms of feature subset size.

Another hybrid study named “A hybrid ant colony optimization and random forest feature selection method for microarray data”, conducted by Xiong and Wang,

(2009) hybridized enhancement self-adaptive ACO and Random Forest (RF) for leukaemia and colon tumour datasets' classification. ACO was used to choose suitable feature subset to present to the RF for classification. Initially, statistical t-test integrated with RF is utilized as a pre-selection for best features and the score of this integration is used as heuristic information combined with classification accuracy produced by RF and utilized as positive feedback for ACO to filter the pre-selected features. Subsequently, SFS is applied to compute LOOCV accuracy of feature subset. The authors compare their results with other approaches and the results shown that their approach obtains higher classification accuracy with fewer numbers of selected features.

2.3.2 Ant Colony Optimization for Optimizing Continuous Variable

While ACO was firstly presented to solve discrete optimization problems, it is modified to solve continuous-variables and mixed-variables optimization problems, because it cannot be applied to continuous-variable or mixed-variables optimization problems directly; consequently this direction is more interesting and one of the modern research fields in ACO (Socha, 2004; Blum, 2005; Dorigo & Blum, 2005; Socha & Blum, 2006; Dorigo, Birattari & Stutzle, 2006b***;*** Castro, 2007; Socha & Dorigo, 2008; Socha, 2008; Dorigo & Stutzle, 2010; Liao, 2011; and Liao et al., 2011).

Kovarik (2006) categorizes the literature of ACO to the continuous domain based on how ACO extended to handle continuous domain in three categories: direct simulation, discretization, and probabilistic sampling.

Direct simulation approaches did not directly follow ACO, but attempt to directly simulate real ants' movement and communication. Below are some studies that are included in this category.

Bilchev and Parmee (1995) present Continuous Ant Colony Optimization (CACO) integrated with GA as a global seek procedure. CACO is considered as the first study related to using ACO to a continuous domain, while Monmarch, Venturini and Slimane (2000) present pachycondyla apicalis (API). Dreo and Siarry (2002 and 2004) propose Continuous Interacting Ant Colony (CIAC) then CIAC was modified in 2006 by the same authors and presents a new algorithm called Hybrid CIAC (HCIAC). HCIAC is CIAC but hybridized with Nelder-Mead local search algorithm. HCIAC solved the drawbacks of CIAC. The drawbacks include the high amount of testing needed, the objective function and bad productivity. In 2007, Dreo and Siarry enhanced HCIAC and present Dynamic HCIAC (DHCIAC). DHCIAC includes two algorithms: $\text{DHCIAC}_{\text{find}}$ which hybridized with Nelder Mead local search algorithm; and $\text{DHCIAC}_{\text{track}}$ hybridized with adaptive simplex local search algorithm. The experiment shows that $\text{DHCIAC}_{\text{track}}$ performs better than $\text{DHCIAC}_{\text{find}}$ when working on fast problems, while $\text{DHCIAC}_{\text{find}}$ performs better results for value-based problems than for location-based problems as in $\text{DHCIAC}_{\text{track}}$. $\text{DHCIAC}_{\text{track}}$

implements the intensification strategy and enhances better performance for location-based problems while DHCIAC$_{find}$ implements the diversification strategy and offers more productivity on value-based problems.

According to Socha (2004), Socha and Blum (2006), Socha (2008), and Socha and Dorigo (2008); CACO and CIAC try to extract stimulation from the ACO, but they do not follow it directly, because they append some extra mechanisms, for example, direct communication in CIAC and API or nest term as in CACO which do not appear in classical ACO. Also they remove some other mechanisms such as properties of ACO like stigmergy was not in all API or incremental building solutions. CACO and CIAC deal with continuous optimization, while API may be utilized for discrete and continuous problems also.

ACO with discretization discretizes the search domain of continuous variables and labels pheromone quantity to a finite number of places expressed by these ranges. The following studies are ACO with discretization.

Yang et al., (2010) propose enhancing Ant Colony Algorithm (ACA) by hybridized elite strategy and distribution function to optimize continuous domain. Wei, Tuo and Jing (2010) present adaptive and commutative Binary Ant Colony Optimization (ACBACO) which is based on Binary Ant Colony Optimization (BACO). Xue, Sun and Peng (2010) propose to use the mutation process to enhance ACA. Hu, Zhang and Li (2008) present Continuous Orthogonal Ant Colony (COAC). COAC uses an orthogonal concept and ACO to optimize continuous variables. A hybridization approach between Orthogonal Scheme (OS) and ACO (OSACO) is presented by Zhang et al. (2009). Huang and Hao (2006) present Continuous ACO-Direct Encoding (CACO-DE). CACO-DE is based on direct encoding and uses strings of integer digits (0-9) to encode limit-accuracy real value parameters as well as adding extra digits at the start of the string to indicate the sign of the real value. Kong and Tian (2005) propose a Binary Ant System (BAS) which is considered as a variant of hyper cube frame for ACO.

ACO with probabilistic sampling would not discretize a seek domain, but an ant's position can be expressed probabilistically. These approaches simulate continuous quantities of pheromone in the seek domain. ACO with probabilistic sampling studies are listed below.

Seçkiner et al. (2013) propose a new method based on continuous ACO ($\mathrm{ACO_R}$) with novel pheromone updating. The pheromone is modifying based on ratios which compute the number of ants to track the best solution. Pourtakdoust and Nobahari (2004) modify ACS to optimize continuous variables and propose a new algorithm named Continuous Ant Colony System (CACS). Kong and Tian (2006a and 2006b) propose Direct Ant Colony Optimization (DACO). Hong and Shibo (2008) present Dynamic ACA (DACA). Franca et al., (2008) present a Multivariate Ant Colony Algorithm for Continuous Optimization (MACACO). Chen, Sun and Wang (2009) propose a new method based on replacing the discrete frequency in the ant choosing probability by a continuous probability distribution. They use the same mechanism used in ACO_R in storing the solution and updating it and name it solution array.

Socha (2004) and Socha and Dorigo (2008) propose ACO for continuous and mixed-variable optimization. Their work considers being the first approach that extends ACO framework to deal with continuous problems. They follow the same concept of established ACO but shift from using discrete probability distribution to utilizing a continuous probability distribution which is the Gaussian Probability Density Function (PDF) in the solution building step which is a very simple manner for data sampling. The disadvantage of this PDF is the inability to express circumstances where two disjointed regions of the seek domain are hopeful, as it only has one maximum. To overcome the problem, a PDF based on enhanced Gaussian functions (Gaussian kernel PDF) is utilized. A Gaussian kernel is a weighted sum of several one-dimensional Gaussian functions. For this, many Gaussian kernel PDFs are used as the number of dimensions of the problem increases instead of utilizing only a single PDF. The authors compare their work with three ant methods: CACO, API, and CIAC. The comparison shows that ACO_R is the best. They suggest executing ACO_R to mixed-variable (continuous + discrete) optimization problems. The original performance of ACO_R is in R that is a language and environment for statistical computing and graphics. For higher execution productivity, it must be implemented with compiled programming language, like C or C++ for example.

Socha (2008) proposes Ant colony optimization for continuous and mixed-variables domains. He follows same concept in his previous works (Socha, 2004 and Socha & Dorigo, 2008) except he utilizes a typical benchmark examining function. He also applies ACO_R to train an ANN. Medical pattern classification is used as a test problem; these problems hybridize types of ACO_R, called ACO_R-BP and ACO_R-Levenberg Marquardt (ACO_R-LM). ANN is utilized as a classifier and the training approach for the ANN was ACO_R. His experiment shows that the execution of simply ACO_R does not achieve the execution of derivative build approaches and the hybridized ACO_R execution is not bad and the results can be enhanced with ACO_R-BP or ACO_R-LM rather than simply ACO_R. He compares his results with the results obtained from using GA applied on the same problem and his results indicate that ACO_R supplies high execution for the examiner. His initial result was hopeful and this encourages examining ACO_R on more complex problems.

Liao (2011) and Liao et al. (2011) propose two new algorithms for solving continuous parameter optimization problems. These two new algorithms, based on ACO_R, are Incremental Ant Colony Optimization ($IACO_R$) and Incremental Ant Colony Optimization with Local Search ($IACO_R$-LS). Both of these algorithms enhance the original ACO_R. These enhancements are as follows. The size of solution archive in $IACO_R$ and $IACO_R$-LS is increased over time until it reaches a specific number, the mechanism for choosing the solution that directs the creation of new solutions and, finally, in solving stagnation. The algorithm involves an algorithm-level diversification strategy to avoid and solve 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, with a relative solution improvement lower than a certain threshold. $IACO_R$-LS algorithm is a hybridization of $IACO_R$ with a local search method. The local seek procedures halt if it reaches a maximum number of iterations or when the tolerance that is the related alternation between solutions obtained in two succeeding iterations is lower than a defined variable. They utilize an adaptive step size for the local seek methods. In order to avoid stagnation at the level of the local seek, the local seek method will be called from many alternative solutions from time to time. A variable *MaxFailures* computes the maximum number of duplicated calls to the local seek procedure from the same initial solution that does not result in a solution enhancement. The inventors of $IACO_R$-LS use three procedures for local seek, which are, Powell's conjugate directions set, Powell's BOBYQA and Lin-Yu Tseng's Mtsls1 procedures. From their experiments on nineteen functions, $IACO_R$-Mtsls1 appears to be more successful compared with other two local search types. Also, $IACO_R$-Mtsls1 compared with sixteen algorithms characterized in a recent special issue of the Soft Computing Journal show that $IACO_R$-Mtsls1 notably executes better than the original ACO_R and it can be competitive with the state of the art. $IACO_R$-Mtsls1 also tested twenty one extra functions taken from a special session on continuous optimization of the IEEE 2005 Congress on Evolutionary Computation and the results show that $IACO_R$-Mtsls1 remains very competitive. The authors suggest modify $IACO_R$ and $IACO_R$-LS to deal with mixed-variable optimization problems.

2.3.3 Ant Colony Optimization for Optimizing Mixed-Variable

Many real world problems are, lately, handled utilizing mixed kinds of decision parameters. A popular example is a mixture of discrete parameters and continuous parameters. Because of the empirical applicable of such problems, many mixed-parameters optimization algorithms have been presented built on GA, Differential Evolution, PSO, and Pattern Search. One of the most interesting ACO's for mixed-variables is Socha's work.

Socha (2008) proposes ant colony optimization for continuous and mixed-variables domains (ACO_{MV}). ACO_{MV} is an ACO_R expanding for mixed-variable optimization problems. ACO_{MV} combines an element of a continuous relaxation method $ACO_{MV\text{-}O}$ and an element of a native mixed-variable optimization method $ACO_{MV\text{-}C}$, as well as ACO_R. ACO_{MV} permits the definition of each parameter of the related problem as continuous, ordered discrete or categorically discrete. The continuous parameters of mixed-variable problems will be handled exactly as in ACO_R. If ordered discrete parameters are presented, an element of the continuous relaxation method, $ACO_{MV\text{-}O}$, is utilized and managed as in the original ACO_R. $ACO_{MV\text{-}C}$ is used for categorical parameters and the values of these parameters are created with an alternative approach, one that is near to the standard combinatorial ACO. To avoid stagnation, an easy restart mechanism involves restarting the algorithm but preserving the best-so-far solution in the archive. The restart condition is the number of iterations of ants modifying the archive with related solution enhancement lower than a constant threshold. In order to evaluate ACO_{MV}, Socha proposes new benchmark problems constructed on a rotated ellipsoid function as well as applying ACO_R and ACO_{MV} on three mixed-variables mechanical engineering examination problems which are, Pressure Vessel Design (PDS), Coil Spring Design (CSD), and Designing Thermal Insulation System (DTIS). The results show that ACO_R executes better on the PVD and CSD problems because these two problems do not involve any categorical parameters, ACO_{MV} is better on the DTIS problem which involves categorical parameters. Also the results taken from ACO_R and ACO_{MV} are good compared with the best results taken from the literature for the same three problems. Socha suggests undertaking more research on the execution of ACO_R and ACO_{MV} on different mixed-variable optimization problems; this is required to obtain better understanding of the advantages and disadvantages. Also, he suggests re-implementing his approach in *C* or other compiled language which is properly optimized, because he implements his approach with R-based implementation which is not efficient, nor is it properly optimized.

There are also another two approaches applied to mixed-variables, as discussed below.

A paper written by Madadger and Afshar (2011) proposes Modifies Improved ACO_R (M-$IACO_R$) which is a modification to ACO_R in utilized explorer ants and adaptation operator to guarantee the trustworthiness of the final solution, and the ants must vastly discover the decision domain at initial seek steps and gradually shrink surround the best solutions while explorer ants are utilized to jump from failing into local optimum. A discretization technique is used to deal with continuous variables. In order to evaluate M-IACO$_R$, the authors apply it on forced cross-country water main design and shared pumping sites' problems as a non-convex and highly non-linear problem. The authors suggest applying M-IACO$_R$ on different mixed parameters engineering design problems and on multi-objective optimization problems in order to explain the execution of their approach.

Sampling Ant Colony Optimization (SamACO), presented by Hu et al., (2010) alternates from ACO$_R$ in which it is sampling a nominee value for each parameter and choosing the values to form solutions. The contribution of SamACO is that the solution of a continuous optimization problem is, in effect, a group of feasible parameter values, which can be considered as a solution "track" moved by an ant. SamACO utilizes the benefits of classical ACO by means of sampling hopeful parameter values from the continuous space and utilizing pheromones on the nominee parameter values to direct the ants' solution building. SamACO examines sixteen benchmark numerical functions (unimodal and multimodal) and compares them with three methods. The first involves the ant-based methods such as CACO, COAC, and ACO$_R$. The second includes other well-known SI methods like ACO and PSO which is an enhancement version of PSO. The third and final kind includes representative methods that utilize an explicit probability learning notion. The results show that the execution of SamACO is the best compared with other techniques in terms of accuracy and computation.

2.4 Summary

Many approaches have been applied to classify patterns with high classification accuracy and SVM is considered as an excellent approach to classify patterns. It suffers from dealing with huge numbers of features and poor settings to its parameters which influence its performance. ACO is considered as an excellent optimization tool that can be used to solve SVM problems. However, the classical ACO cannot deal with continuous value. Recent ACO algorithms extend the established ACO to deal with continuous and mixed-variables optimization problems which can be used to solve SVM problems. Integrating ACO and SVM is an approach that can be studied to optimize feature subset and values for SVM parameters.

CHAPTER THREE
RESEARCH METHODOLOGY

This chapter presents the framework and methodology for this thesis to develop algorithms to classify patterns using hybrid Support Vector Machine (SVM) and Ant Colony Optimization (ACO) algorithm variants. Section 3.1 discusses the research framework. Section 3.2 gives the datasets development. While Section 3.3 presents the proposed methodology that used to tune SVM parameters; Section 3.4 presents the methodology used to simultaneously tune SVM parameters and select feature subset. Finally, section 3.5 summarizes this chapter.

3.1 Research Framework

The research framework is the roadmap of the research that aims to provide guidance to industry, researchers, and funding agencies (Forrester, 2006).

The research framework starts with the datasets' development. After that, new enhanced pattern classification algorithms are proposed for continuous SVM parameters. The third phase deals with proposing new enhanced pattern classification algorithms that have the ability to simultaneously optimize SVM parameters and feature subset selection. Figure 3.1 depicts the phases of the research framework.

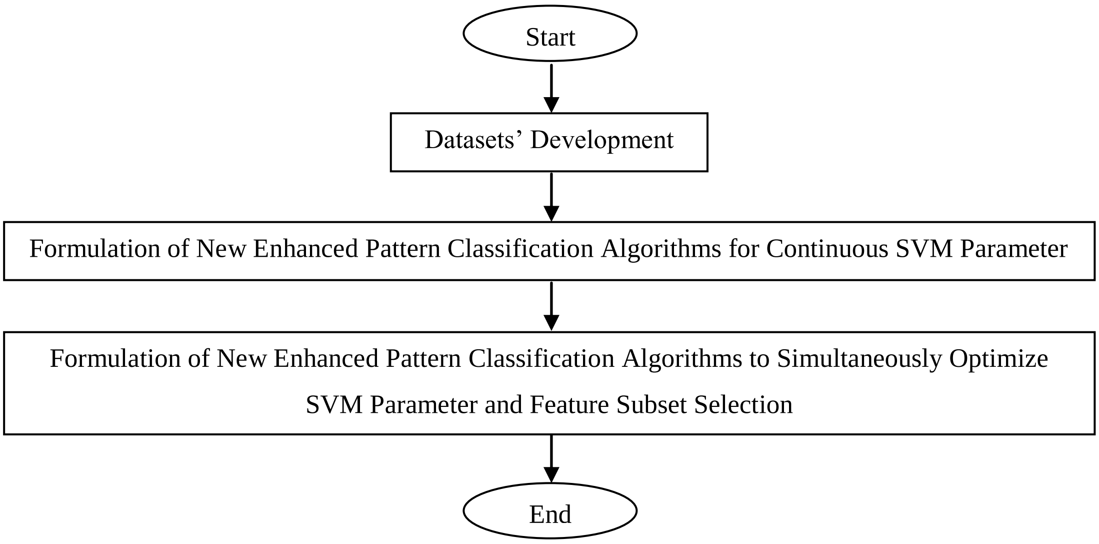

Figure 3.1 Research framework phases

Each phase of the framework has its own methodology. Research methodology is the route used to solve the research problem. It may be understood as a science of studying how research is carried out scientifically. The research methodology relates to the logic behind the methods used in the context of the research study and explains the used of one particular method or technique rather than another order to evaluate the research results. The research methodology is divided into retrospective, prospective, experimental, and non-experimental studies (Kumar, 2011).

This thesis proposes an enhanced pattern classification of algorithms based on hybridized ACO and SVM techniques. The proposed algorithms require to be evaluated with other approaches. In order to do this evaluation, conducting experiments are needed using datasets. Therefore, an experimental methodology approach is adopted in this thesis.

3.2 Datasets' Development

3.2.1 Datasets' Description

A collection of ten datasets from a University of California, Irvine (UCI) repository have been used in this study. The datasets comprise: Australian, Pima-Indian Diabetes, Heart (Statlog), Ionosphere, German, Sonar, Splice, Image Segmentation, Iris, and Vehicle. Table 3.1 summarizes the main characteristics for these datasets.

Table 3.1: Summary of UCI Datasets' Repository

Datasets	**Number of Instances**	**Number of Features**	**Number of Classes**	**Features' Type**
Australian	690	14	2	Categorical, Integer, Real
German	1000	20	2	Categorical, Integer
Heart (Statlog)	270	13	2	Categorical, Real
Image Segmentation	2310	19	7	Real
Ionosphere	351	34	2	Integer, Real
Iris	150	4	3	Real
Pima-Indian Diabetes	768	8	2	Integer, Real
Sonar	208	60	2	Real
Splice	3190	61	3	Categorical
Vehicle	846	18	4	Integer

The Australian Credit Approval dataset relates to credit card applications. This dataset contains 690 instances with 14 features, divided into two classes: 307 positive instances and 383 negative instances as well as containing missing values.

The German Credit dataset provided by Prof. Hofmann contains 1000 instances with 20 features, divided into two classes: 700 good instances and 300 bad instances; no information regarding missing data was available for this dataset.

The Heart dataset relates to the heart disease database and involves 270 instances with 13 features divided into two classes: 150 absent instances and 120 present instances with no missing data.

The Image Segmentation dataset was generated by Vision Group, University of Massachusetts. This dataset contains 2310 instances with 19 features, divided into seven classes: brickface, sky, foliage, cement window, path, and grass and with no missing data in this dataset.

The Ionosphere dataset consists of radar information returning from the ionosphere collected by a system in Goose Bay, Labrador. This dataset includes 351 instances with 34 features, divided into two classes: 225 good instances of radar returns are those showing evidence of some kind of structure in the ionosphere and 126 bad instances returns are those that do not, their signals pass through the ionosphere. This dataset contains no missing data.

The Iris dataset, developed by R. A. Fisher, includes 150 instances with four features, divided into three classes: Iris Setosa, Iris Versicolour, and Iris Virginica. There is no missing data with this dataset.

The Pima Indians Diabetes dataset, collected from female patients at least 21 years old, includes 768 instances with eight features, divided into two classes: 500 tested positive for diabetes and 268 tested negative for instances of diabetes. There are no missing values in this dataset.

The Sonar dataset, utilized by Gorman and Sejnowski, includes 208 instances utilized to distinguish between sonar signals bounced off a metal cylinder (Mines) and those bounced off a roughly cylindrical shape (Rock). This dataset includes 60 features divided into two classes: 97 Rock instances and 111 Mines instances; missing data information is not available for this dataset.

Splice junctions are positions on Deoxyribonucleic acid (DNA) series at which extra DNA is eliminated during the task of protein generation in higher organisms. The problem presented in this dataset is to recognize, given a series of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out). This problem includes two subtasks: recognizing exon/introns boundaries (referred to as EI sites referred to as acceptors), and recognizing introns/exon boundaries (IE sites referred to as donors). This dataset includes 3190 instances with 61 features, divided into three classes: 767 EI, 768 IE, and 1655. There is no missing data in this dataset.

The last dataset utilized in this thesis is the vehicle dataset, developed by Dr. Pete Mowforth and Barry Shepherd at the Turing Institute, Glasgow. The purpose is to classify a present silhouette as one of four kinds of vehicle, utilizing a group of features extracted from the silhouette. This dataset contained 946 instances with 18 features, divided into four classes: 212 Opel instances, 217 Saab instances, 218 bus instances, and 199 Van instances with no missing data.

3.2.2 Datasets' Cleaning

Data cleaning is needed to enhance the quality of the raw data for classification which will maximize the classification efficiency. Data from real-world sources are usually incorrect, imperfect, and conflicting, perhaps due to execution mistakes or system application flaws. Such low-quality data require to be cleaned prior to classifying the data (Colak, Sagiroglu & Yesilbudak, 2012; Gurbuz, Ozbakir & Yapici, 2011; Saltan, Terzi & Kucuksille, 2011; Farvaresh & Sepehri, 2011; and Christen, 2009). Table 3.2 summarizes the main characteristics for the used datasets after cleaning and this is again depicted in Figures 3.2-3.4.

Table 3.2: Summary of Cleaning Datasets in the UCI Repository

Datasets	Number of Instances	Number of Features	Number of Classes
Australian	397	11	2
German	1000	20	2
Heart (Statlog)	270	13	2
Image Segmentation	2310	18	7
Ionosphere	351	34	2
Iris	150	4	3
Pima-Indian Diabetes	395	8	2
Sonar	208	60	2
Splice	1000	61	3
Vehicle	846	18	4

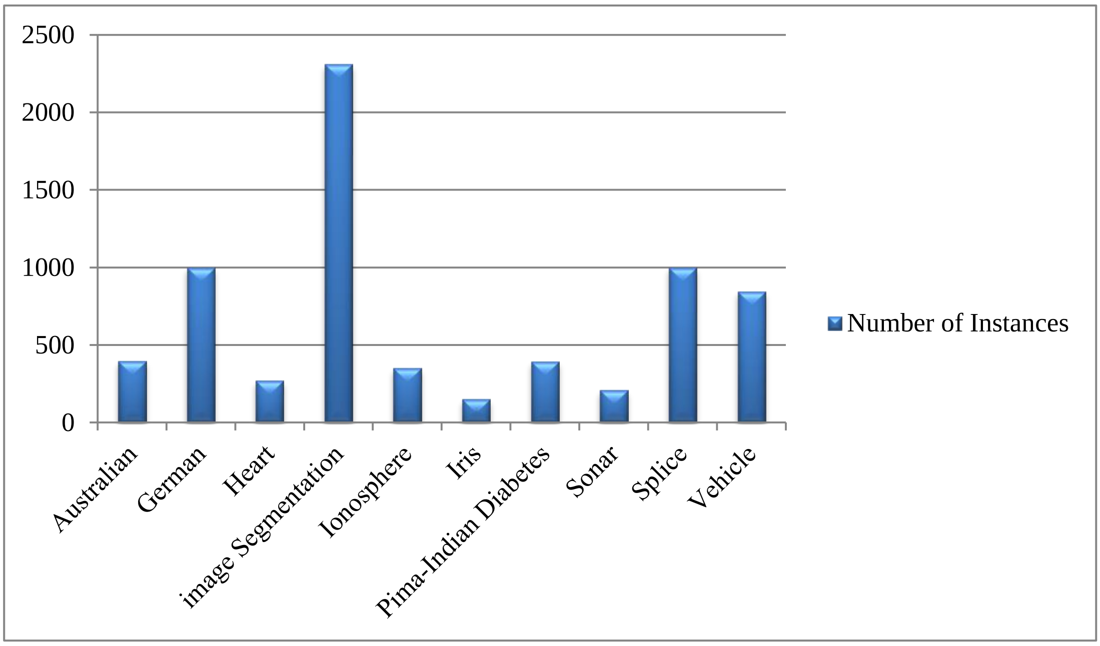

Figure 3.2 Number of instance

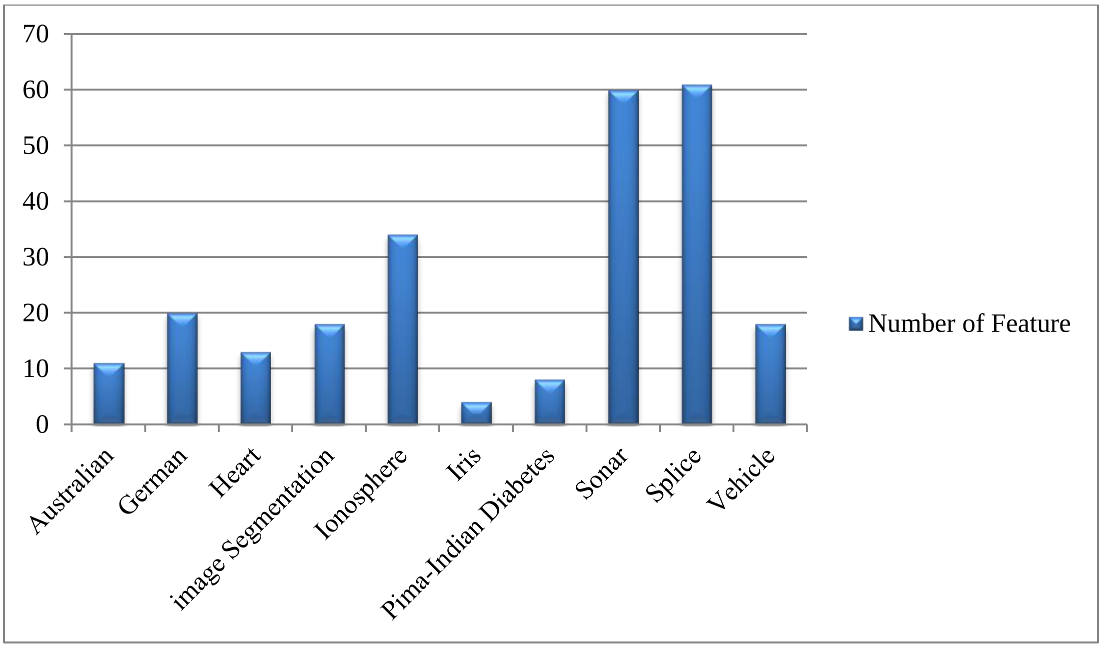

Figure 3.3 Number of feature

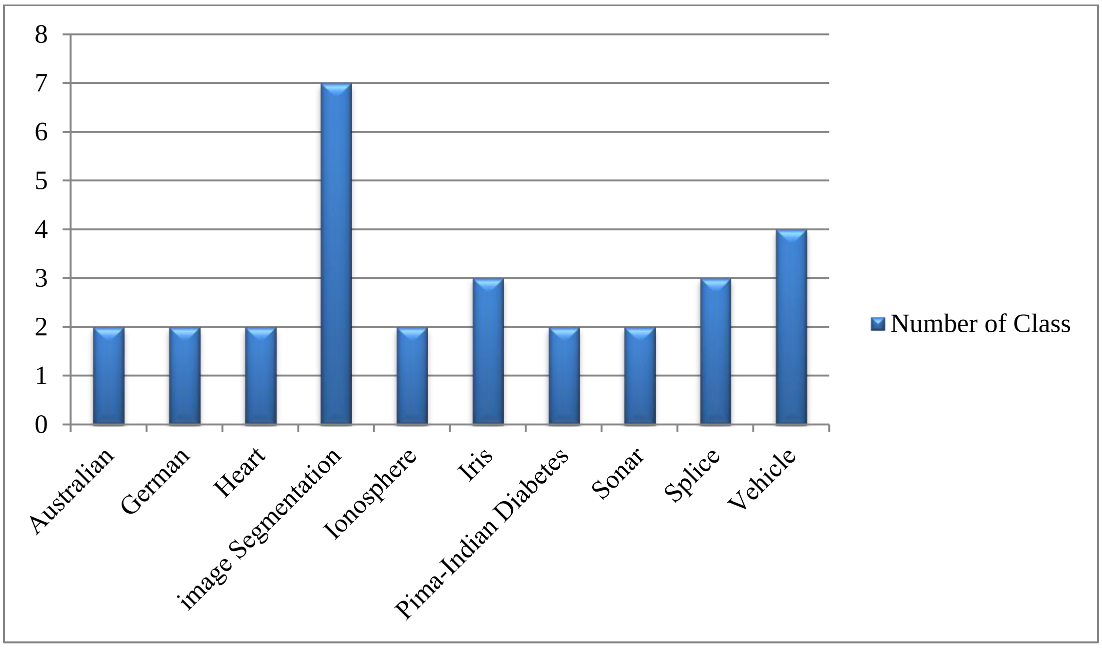

Figure 3.4 Number of class

The Australian Credit Approval dataset incorporated incorrect data so, in the cleaning phase, all incorrect data were deleted. Features 10, 13, and 14 were deleted, because the values for these features are continuous, albeit given in a discrete form. Beside these features, any instances given in discrete form which were hypothetically of continuous value were deleted also.

In the case of the Image Segmentation dataset, all values of third features were equal to nine, so this feature was deleted.

In the Pima Indians Diabetes dataset, some instances contain a value of zero which is biologically impossible, therefore, every instance containing a zero value was deleted.

The Splice dataset did not have any missing or incorrect data, but to simplify calculation only 1000 instances were selected randomly with a class label of only EI or IE, because it is required only to distinguish between these two classes.

3.2.3 Datasets’ Transformation

The Australian Credit Approval, Heart, German Credit, and Splice datasets contain categorical values. In order to deal with these datasets any categorical value were converted to a numerical value to manipulate these datasets. Also, the class labels of all binary classes’ datasets were converted from 1 and 0 to +1 and -1. In the case of multi class datasets the same strategy was used, but after grouping each two classes to become binary class datasets.

3.2.4 Datasets’ Scaling

All the datasets were scaled through the dataset development step to avoid features with high numerical ranges from dominating those in lower numerical ranges and to minimize the computation efforts. All attributes were linearly scaled to [0, 1] rang using this formula (Huang & Wang, 2006; Pham et al., 2007; Lin et al., 2008; Huang & Dun, 2008; Ding & Li, 2009; Lin & Chien, 2009; Huang, 2009; Chang & Lin, 2011; Zhao et al., 2011; and Sarafrazi & Pour, 2013):

$$\bar{x} = \frac{x - min_i}{max_i - min_i} \tag{3.1}$$

Where x is the original attribute value, $\bar{x}$ is the scaled attribute value, and max_i and min_i are the maximum and minimum attribute values of attribute i respectively.

3.3 Formulation of New Enhanced Pattern Classification Algorithm for Continuous SVM Parameter

The first SVM’s problem, relating to selecting its model, will be dealt with using only continuous ACO variants which are continuous ACO (ACO_R) and Incremental continuous ACO ($IACO_R$) for solving this problem without using ACO to select suitable feature subset, but an Fisher (F)-score will be used to select suitable feature subset. Here, a filter based approach is used to select feature subset after which the SVM classifier will be trained. In this way, it will not be possible to tightly gather the overall classification accuracy produced by SVM classifier and necessary features. The drawback for this schema is that the optimal feature subset selection will not be dependent of the inductive and representational biases of the learning algorithm that is utilized to build SVM classifier. An ant solution was used to represent a combination of the SVM classifier parameters, penalty parameters *C* and Radial Basis Function (RBF) kernel function parameter γ. Based on the solution archives, the transition probability is computed to choose a solution path for an ant.

The proposed algorithms are tested on ten benchmark datasets from the UCI repository and compared with other approaches such as Genetic Algorithm (GA)-SVM (Huang & Wang, 2006), $\mathrm{GA}_{\text{with feature chromosome}}$-SVM and Grid search (Zhao et al., 2011), ACO-SVM (Huang, 2009), Particle Swarm Optimization (PSO)-SVM (Lin et al., 2008) and Simulated Annealing (SA)-SVM (Lin et al., 2008) in terms of classification accuracy and feature subset size. The proposed algorithms have been implemented in C programming language. To compare the classification accuracy of the proposed algorithms with other approaches, the non-parametric Wilcoxon signed rank test for all of the datasets is used as it entails fewer and simpler calculations (Weiss, 2008).

3.4 Formulation of New Enhanced Pattern Classification Algorithm to Simultaneously Optimize SVM Parameter and Feature Subset Selection

The second SVM's problem, which is related to selecting suitable feature subset with a small number of features besides the SVM model selection problem, will be dealt with using mixed-variables ACO variants which are mixed-variable ACO ($\mathrm{ACO}_{\mathrm{MV}}$-

$_R$) and Incremental mixed-variable ACO (IACO$_{MV\text{-}R}$). Features are represented as discrete graph nodes while C and γ SVM parameters are continuous values, so there will be a need to use ACO$_{MV}$ that can deal with discrete, continuous, or both values' types. The number of chosen features varies from ant to ant. Hence, it is not necessary for an ant to visit all the features. The termination criterion for ant to stop its visits to the feature is when the ant arrives at a predefined selection of features which will be generated randomly. The ant's solution will represent a mix of SVM parameters which are penalty parameters C and γ for RBF kernel function and feature subset. Based on the solution archive, pheromone table and suitable features, the probability of transition is computed to choose a solution track for the ant. The pheromone table and solution archives will be modified based on the classification accuracy and feature quality. Wrapper approach strategy, to select feature subset, will be used to hybridize ACO variant with SVM. This will use the overall classification accuracy produced through the SVM classifier and the necessary features to hybridize together into ACO variants' algorithms. In this case, the optimal selected features will be dependent on the inductive and representational biases the learning algorithms that are utilized to build the SVM classifier.

The proposed algorithms are tested on ten benchmark datasets from the UCI repository and compared with other approaches such as Gravitational Search Algorithm (GSA)-SVM (Sarafrazi & Pour, 2013), GA$_{\text{with feature chromosome}}$-SVM and Grid search (Zhao et al., 2011), ACO-SVM (Huang, 2009), Cat Swarm Optimization (CSO)-SVM (Lin & Chien, 2009), Clonal Selection Algorithm (CSA)-SVM (Ding &

Li, 2009), PSO-SVM (Lin et al., 2008), SA-SVM (Lin et al., 2008), and GA-SVM (Huang & Wang, 2006) in terms of classification accuracy and feature subset size. The proposed algorithms have been implemented in C programming language. To compare the classification accuracy of the proposed algorithms with other approaches, the non-parametric Wilcoxon signed rank test was used for all of the datasets as it entails fewer and simpler calculations (Weiss, 2008).

3.5 Summary

In constructing hybrid ACO-SVM, it is necessary to work with the continuous value of SVM parameters and prevent discretization which, if done, will affect the classification accuracy. Additionally, proposed algorithms that have the ability to simultaneously optimize model and feature subset selection by each ant can produce classifier models that could classify data with acceptance classification accuracy and help humans to understand data better, especially for large amounts of data.

CHAPTER FOUR

ENHANCED ACO-SVM PATTERN CLASSIFICATION ALGORITHM FOR CONTINUOUS SVM PARAMETER

This chapter presents the proposed algorithms for tuning Support Vector Machine (SVM) parameters. Section 4.1 presents the proposed two algorithms; while Section 4.2 presents the experiments made on ten datasets from the University of California, Irvine (UCI) repository to evaluate the performance of the proposed algorithms. Finally, the summary of the chapter is presented in Section 4.3.

4.1 Enhanced Pattern Classification Algorithm for Continuous SVM Parameter

A classification algorithm is a methodical approach to construct classification models from an input data set. Each algorithm utilizes a learning algorithm to identify a model that best matches the correlation between the attribute group and class label of the input data. The model produced through a learning algorithm should both perfectly match the input data and exactly forecast the class labels of records it has never seen before. Therefore, the main goal of the learning algorithm is to construct models with good generalization capability; i.e., models that precisely forecast the class labels of previously unknown records (Tan, Steinbach & Kumar, 2006). Figure 4.1 depicts the generic flowchart for constructing a classification model. The result, which is the classification accuracy, will depend on the learning algorithm that is used. The learning algorithm in Figure 4.1 will be represented in detail in Figure 4.2 and Figure 4.3.

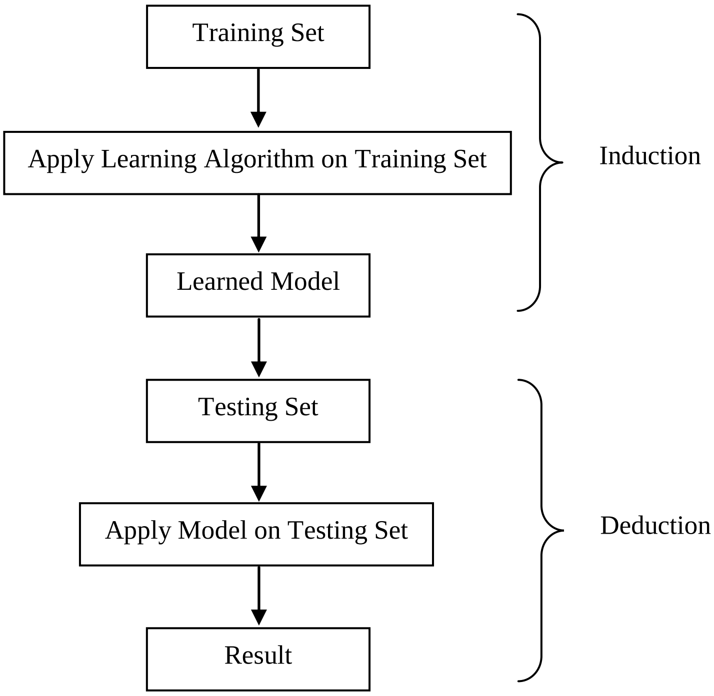

Figure 4.1 Generic flowchart for constructing a classification model

Figure 4.2 depicts the general flowchart of SVM pattern classification approaches that will convert the continuous value of SVM parameters to discrete values. As mentioned in Chapter One, this conversion process has produced classification errors and, thus, affects the classification accuracy (Huang, 2009), because it will limit the details at which optimistic regions of the seek domain can be explored (Blondin & Saad, 2010). Studies that discretize the continuous values have been reported in Zhao et al. (2011), Zhang Chen & He (2010), Huang (2009), Lin et al. (2008), and Huang & Wang (2006).

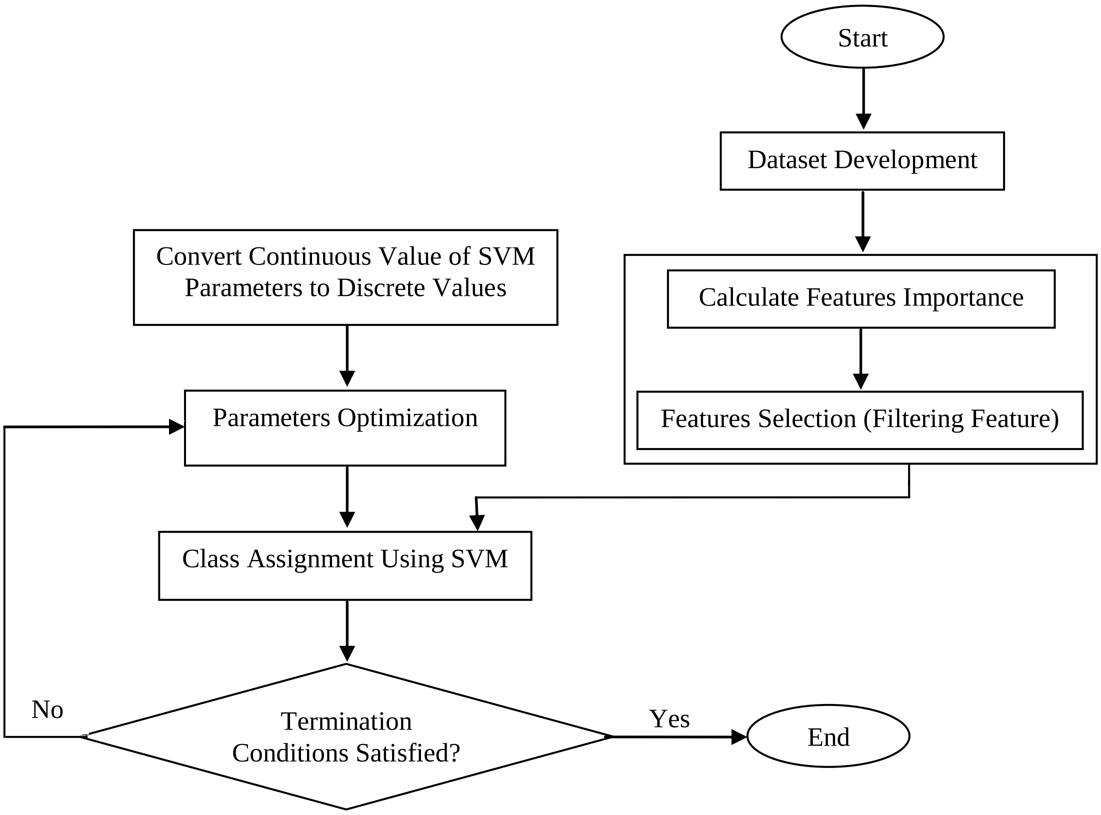

Figure 4.2 Flowchart for SVM pattern classification approach

In this thesis, two algorithms are proposed based on Ant Colony Optimization (ACO) variants, continuous ACO (ACO_R) and Incremental continuous ACO ($IACO_R$), in order to obtain the optimal continuous value for SVM parameters to classify patterns with acceptable classification accuracy (refer to Figure 4.3). These two proposed algorithms can directly handle the continuous nature of SVM parameters values in order to optimize it and then reduce the classification error. This was not possible in other approaches as mentioned earlier. One of the new ACO research directions is to optimize the continuous problem; so, ACO_R is considered the first algorithm that can handle the continuous variable and follow the same

ACO's framework, while IACO$_R$ is used to improve ACO$_R$ performance in solving stagnation.

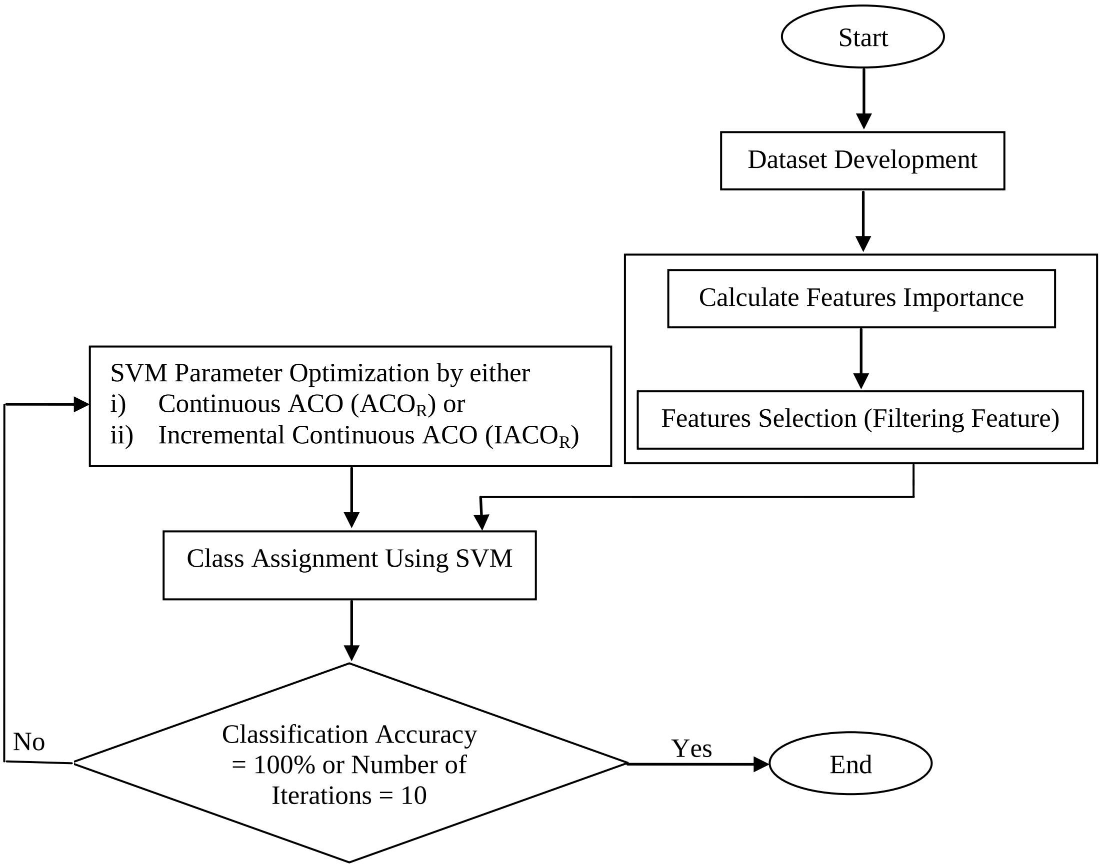

Figure 4.3 Flowchart for enhanced ACO-SVM pattern classification algorithm

Figure 4.4 presents the generic architecture for the proposed pattern classification algorithms. As shown in this figure, the proposed algorithm starts with dataset development followed by selection of the feature subset through filtering features by using the Fisher (F)-score and passing it to the SVM. Subsequently, and regarding user specification, ACO$_R$ or IACO$_R$ will be executed to optimize the value of SVM parameters and will be sent to the SVM to compute classification accuracy which will be fed back to ACO$_R$ or IACO$_R$ to update the solution archive in case the produced classification accuracy is not satisfied, otherwise the proposed algorithms will be halted.

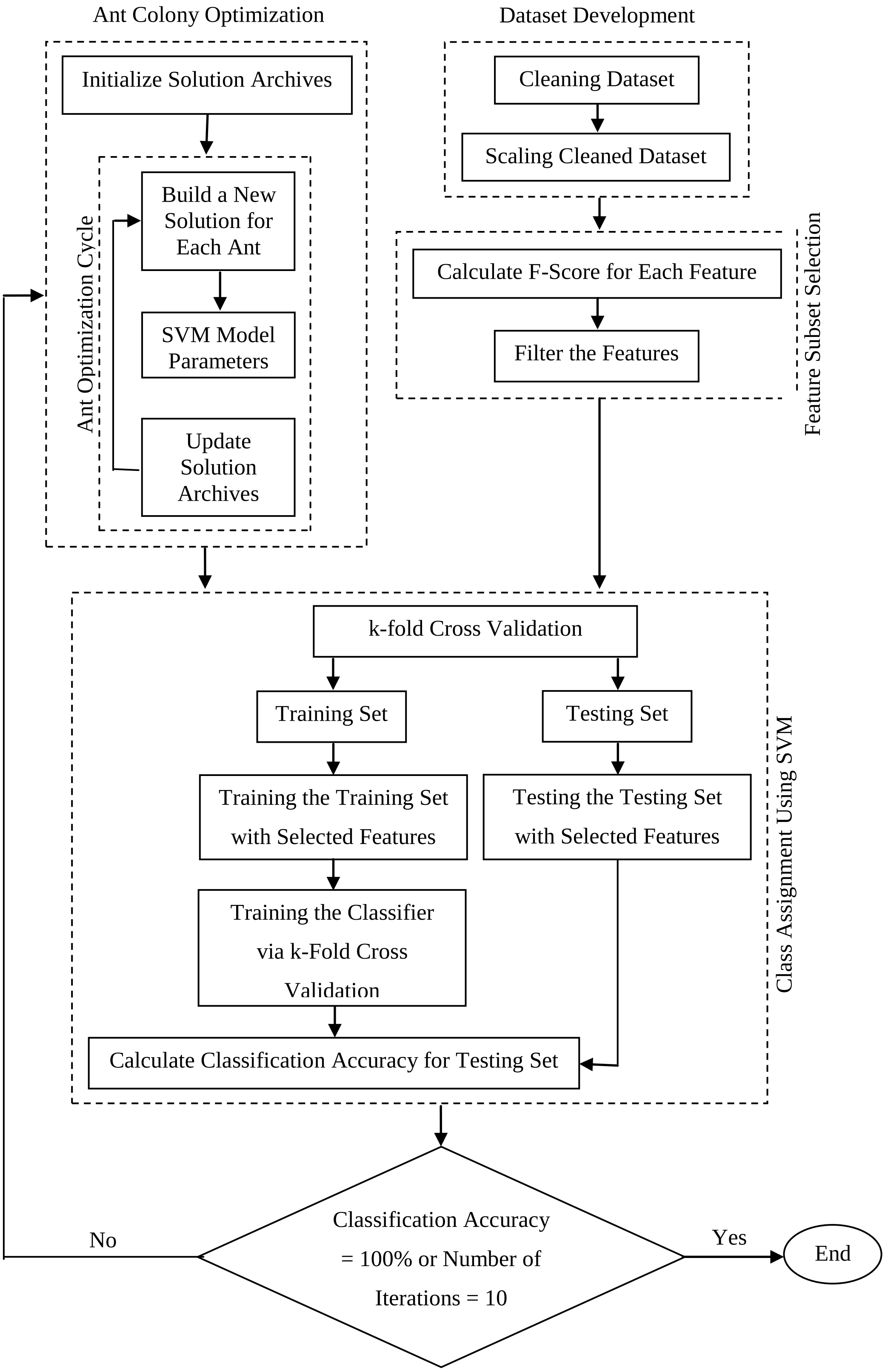

Figure 4.4 Enhanced generic ACO-SVM pattern classification algorithm architecture

Finally, Figure 4.5 presents the generic pseudo code for the proposed algorithms.

```
The Proposed Algorithm
Input: Datasets
Output: Optimal value for SVM parameters and classification accuracy
Begin
    clean dataset
    scale cleaned dataset using Eq. (3.1)
    //filtered features
    for each dataset do
        calculate F-score for each feature according to Eq. (4.1)
        calculate threshold = average F-score
        for $i$ = 1 to no. of features do
            if F-score for $feature_i \geq threshold$ then
                append $feature_i$ to features subset
            end
        end
        for $i$ = 1 to no. of runs do
            for $j$ = 1 to no. of iterations do
                //tune SVM parameters
                call $ACO_R$-SVM or $IACO_R$-SVM
            end
        end
    end
End
```

Figure 4.5 Generic pseudo code for proposed enhanced ACO-SVM algorithm

4.1.1 Datasets' Development

The proposed algorithms begin with the develop of the datasets as explained in Chapter Three which includes cleaning missing data from the dataset, data transformation and scaling the dataset using Eq. (3.1) to avoid features with high numerical ranges from dominating those in lower numerical ranges and to minimize the computation efforts. All attributes are linearly scaled to [0, 1] range.

4.1.2 Select Feature Subset

After the datasets are developed, selecting feature subset through filtering features using a F-score technique will be made. F-score is used as a measurement to determine the importance of features. The reason for using this measurement is because it is an easy measurement that judges the favoritism capability of a feature. A high value F-score means that it is most likely that this feature is favorable. The disadvantage of this measurement is that it takes each feature in isolation so it cannot reveal mutual information between features (Chen & Lin, 2006; and Chang & Lin, 2011). The equation for the F-score is as follows (Chen & Lin, 2006; Chang & Lin, 2011; and Huang, 2009):

$$F - Score_i = \frac{\sum_{c-1}^{v}\left(\bar{x}_i^{(c)} - \bar{x}_i\right)^2}{\sum_{c-1}^{v}\left(\frac{1}{N_i^{(c)}-1}\sum_{j=1}^{N_i^{(c)}}\left(x_{i,j}^{(c)} - \bar{x}_i^{(c)}\right)^2\right)}, i = 1, 2, \ldots, N_f \qquad (4.1)$$

where v is the number of categories of target variables. N_f is the number of features. $N_i^{(c)}$ is the number of samples of the i^{th} feature with categorical value c, $c \in \{1, 2, \ldots, v\}$. $\bar{x}_{i,j}^{(c)}$ is the j^{th} training sample for the i_{th} feature with categorical value c. $j \in \{1, 2, \ldots, N_i^{(c)}\}$, $\bar{x}_i$ is the ith feature, and $\bar{x}_i^{(c)}$ is the i^{th} feature with categorical value c. The numerator considers the inter-class variance and the denominator considers the sum of the variance within each class.

Subsequently, the average F-score is computed and it 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 placed in the feature subset and this subset is presented to the SVM to assign a class label for data instances. Table 4.1 shows the difference in selecting feature subset between the proposed algorithms used in this thesis and in other research.

Table 4.1: Filtering Method

Approaches	**Filtering methods**
Proposed ACO_R-SVM Algorithm	F-Score + Threshold
Proposed $IACO_R$-SVM Algorithm	F-Score + Threshold
ACO-SVM	F-Score + keep the first features according to the sorted F-Score
Particle Swarm Optimization (PSO)-SVM	All features
Genetic Algorithm (GA)-SVM	Did not mentioned
Simulated Annealing (SA)-SVM	All features
Grid-SVM	All features

After filtering features and produced feature subset, one of the two new continuous ACO variants will be used to optimize and tune SVM parameters. The following subsection presents details of the proposed ACO_R-SVM and $IACO_R$-SVM algorithms.

4.1.3 Proposed ACO_R-SVM Algorithm

The first proposed ACO_R-SVM algorithm is based on using ACO_R to tune SVM parameters. ACO_R is considered as the first ACO algorithm used to optimize continuous variables (Socha, 2004; Blum, 2005; Socha & Blum, 2006; Socha, 2008; Socha & Dorigo, 2008; Liao et al., 2011; and Liao, 2011). Figure 4.6 depicts the details of the steps of the proposed ACO_R-SVM algorithm.

```
ACO_R-SVM Algorithm
Input: Size of solution archive ($k$), no. of ants ($m$), $q$, range of $C$, range of $\gamma$, and termination
       criterion
Output: Optimal value for SVM parameters ($C$ and $\gamma$) and classification accuracy
Begin
        Initialize $k$ solutions
        call SVM algorithm to evaluate $k$ solutions
        //sort solutions and store them in the archives
        $T$ = Sort ($S_1, \ldots, S_k$)
        while classification accuracy $\neq$ 100% or number of iteration $\neq$ 10 do
                //generate m new solutions
                for $i$ = 1 to $m$ do
                        //construct solution
                        select $S$ according to its weight
                        sample selected $S$
                        store newly generated solutions
                        call SVM algorithm to evaluate newly generated solutions
                end
                //sort solutions and select the best k
                $T$ = Best  (Sort $S_1, \ldots S_k + m$), $k$)
        end

End
```

Figure 4.6 Pseudo code of proposed ACO_R-SVM algorithm

ACO_R-SVM starts with randomly initializing two solution archives for SVM parameters; one for regularization or soft margin parameter C value and the other for Radial Basis Function (RBF) kernel function parameter γ value. These solution archives will be filled with solutions initially generated randomly by dividing C and γ value over parameter k which represents the size of solution archives. This is then used by SVM to compute its objective function.

After initializing solution archives are achieved, the solution will be established by each ant. For each ant establishing a solution path for parameter C and parameter γ, these two solution archives are needed to design the transition probabilities for C and for γ. The weight vector (*w)* is then computed for each sample for *C* and γ stored in solution archives as follows:

$$w_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \tag{4.2}$$

where k is the size of solution archive, q is the algorithm's parameter to control diversification of the search process, and *l* is the index of $solution_l$ in the solution archive which is depend on its objective function. The weights values are also stored in solution archives. 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:

$$p_l = \frac{w_l}{\sum_{r=1}^{k} w_r} \tag{4.3}$$

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. In this thesis, the size of solution achieved (*k*) is used as a random number to sample the selecting *w*. This initializing constructs the transition probabilities for ants. The solution will be constructed by each ant; each ant builds its own solution which will be a combination of *C* and γ. This solution will be sent to the SVM to classify data instances and based on the outcome of the SVM the solution archives will be updated by appending the newly generated group solution that gives the best classification accuracy to the solution archive and then delete the exact number of worst solutions. This ensures that the size of solution archive does not change. This procedure guarantees that only good solutions are stored in the archives, and it will efficiently influence the ants in the seek process.

4.1.4 Proposed IACO$_R$-SVM Algorithm

The second proposed algorithm is IACO$_R$-SVM which incorporates IACO$_R$ to tune SVM parameters. The IACO$_R$ algorithm is an enhanced ACO$_R$ and attempts to solve the main drawbacks of ACO$_R$ which are related to stagnation problems and the application gap with the state-of-art continuous solvers (Liao, 2011 and Liao et al., 2011).

IACO$_R$-SVM as compared to ACO$_R$-SVM has more seek diversification mechanisms. The optimization procedure in IACO$_R$-SVM starts with a small archive size. A parameter *InitArhiveSize* defines its size which will be equal to the *k* parameter as in ACO$_R$-SVM. Like ACO$_R$-SVM, IACO$_R$-SVM starts with randomly initializing two solution archives for SVM parameters; one for regularization or soft margin parameter *C* value and the other for RBF kernel function parameter, γ value. These solution archives will be filled with solutions initially generated randomly as in ACO$_R$-SVM by dividing *C* and γ value over parameter *k* and then these samples will be sent to the SVM to evaluate it and compute objective function for each sample.

IACO$_R$-SVM also characterizes a strategy alternate from the one utilized in ACO$_R$-SVM for choosing the solution that directs the creation of new solutions. The new procedure is built on parameter $p \in [0, 1]$, which monitors the probability of utilizing only 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-SVM, they are compared according to their objective function which is computed according to the SVM output. If the newly created solution is better than the directing solution, it replaces it in the archive. This replacement mechanism is alternate from the one utilized in ACO_R-SVM 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 archives size, defined by *MaxArchiveSize*, is reached. A parameter *Growth* monitors the percentage at which the archives grow. Fast growth percentage support seeks diversification while slow growth supports intensification. Each time a new solution is appended, it is initialized utilizing 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 archive S_{best} utilizing the following formula:

$$\bar{S}_{new} = S_{new} + rand(0,1)(S_{best} - S_{new}) \qquad (4.4)$$

where $rand(0, 1)$ is an arbitrary number in range [0, 1).

$IACO_R$-SVM involves an algorithm-level diversification strategy for fighting 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 which is the classification accuracy. Figure 4.7 depicts the pseudo code of proposed IACO$_R$-SVM algorithm.

```
IACO_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 classification accuracy ≠ 100% or number of iteration ≠ 10 do
    // Generate new solutions
    if rand (0,1) < p then
        for i = 1 to no. of ants do
            Select best solution using Eq. (4.3)
            Sample best selected solution
            Call SVM algorithm to evaluate the new generated solutions
            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 using Eq. (4.3)
            Sample selected S
            Store newly generated solutions
            Call SVM algorithm to evaluate the new generated 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 using Eq. (4.4)
        Add new solution to the archive
        k + +
    end
    // Restart Mechanism
    if # (number) of iterations without improving classification accuracy of S_best =
    MaxStagIter then
        Re-initialize T (solution archive) but keeping S_best
    end
End
```

Figure 4.7 Pseudo code of proposed IACO$_R$-SVM algorithm

4.1.5 Class Assignment Using SVM and Evaluation

A classifier model is constructed after each ant builds its solution to assign a class label for each dataset instance. A 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. The benefits of using CV are (1) each of the test groups is independent and (2) the dependent outcomes can be enhanced (Huang, 2009). k-fold CV will randomly rearrange the whole dataset and divide it into approximately equal sized *k* fold subsets, and the holdout approach is repeated *k* times. One of the *k* sub-groups is utilized as the test set and the remaining *k-1* sub-groups are combined to construct the training group (Ding & Li, 2009 and Huang, 2009). The following procedure shows a k-fold CV technique (Sartakhti, Zangooei & Mozafari, 2011):

1. Re-order data and divide: randomly re-arrange the whole data and then divide it into *k* non-overlapping, approximately equally sized subgroups.
2. Train classifiers over folds: one subgroup is tested using the classifier trained on the remaining *k-1* subgroups. For each $i = 1, 2, \ldots, k$ creates a classifier. Thus, each instance of the whole training group is forecast.
3. Evaluate the percentage of instances that are classified in the correct way to determine k-folds through computing test accuracy using the following formula:

$$Test\ Accuracy = \frac{no.of\ correctly\ predicted\ data}{total\ testing\ data} * \ 100\% \qquad (4.5)$$

4. The above steps will be repeated k times through taking the next subgroup as a test set and the remaining subgroups as training sets, each time utilizing an alternative subgroup as a test set.
5. k different test results available for each training-test arrangement: the average of these results produces the test accuracy of the algorithm. CV accuracy is calculated as follows:

$$CV_{accuracy} = \frac{\sum_i test\_accuracy}{k}, i = 1, 2, \dots, k \quad (4.6)$$

For binary class classification problems, given M training examples where each example is represented through a pair of (x_i, y_i) where $i = 1, \dots, M$, $x_i \in R^M$ corresponds to the feature group for the i^{th} example, and $y_i \in \{+1, -1\}$ denotes the class label, SVM needs 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 (Vapnik, 1995; Vapnik, 1999; Chapelle et al., 2002; Cortes & Vapnik, 1995; Vapnik & Chapelle, 2000; Tan, Steinbach & Kumar, 2006; and Pham et al., 2007):

$$\langle w.x_i \rangle + b \geq +1 \; for \; y_i = +1 \quad (4.7)$$

$$\langle w.x_i \rangle + b \leq -1 \; for \; y_i = -1 \quad (4.8)$$

Gathering inequalities (4.7) and (4.8) gives:

$$y_i(\langle w.x_i \rangle + b) - 1 \geq 0 \; \forall i = 1, \dots, M \quad (4.9)$$

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

$$min_{w,b} \frac{1}{2} w^T w \tag{4.10}$$

Subject to $y_i \langle w.x_i \rangle + b) - 1 \geq 0$ (4.11)

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

$$L_P(w, b, \alpha) = \frac{1}{2} w^T.w - \sum_{i=1}^{M} (\alpha_i y_i (\langle w.x_i \rangle + b) - 1) \tag{4.12}$$

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. (4.12) with respect to *w* and *b*:

$$\frac{\partial}{\partial w} L_P = 0, w = \sum_{i=1}^{M} \alpha_i x_i y_i \tag{4.13}$$

$$\frac{\partial}{\partial b} L_P = 0, \; \sum_{i=1}^{M} \alpha_i y_i \tag{4.14}$$

Substituting Eqs. (4.13) and (4.14) into Eq. (4.12) yields the dual Lagrangian L_D to be maximized:

$$Max_{\alpha_i} L_D(\alpha_i) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j \langle x_i.x_j \rangle \tag{4.15}$$

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:

$$f(x, \alpha_i^*, b^*) = \sum_{i=1}^{M} y_i \alpha_i^* \langle x_i . x \rangle + b^* \tag{4.16}$$

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

$$y_i(x_i w^* + b^*) - 1 = 0 \; for \; i = 1, \dots, M \tag{4.17}$$

Note that vectors x_i for which Eq. (4.17) holds are called support vectors and w is the feature's weight, utilizing Eq. (4.13):

$$w^* = \sum_{i=1}^{M} \alpha_i^* y_i x_i \tag{4.18}$$

In non-separable cases, the goal is to build a hyperplane that will generate the smallest number of classification mistakes. Slack variables $\xi_i \geq 0, i = 1, \dots, M$ are introduced in Inequalities Eq. (4.7) and Eq. (4.8) such that:

$$\langle w . x_i \rangle + b \geq +1 - \xi_i \; for \; y_i = +1 \tag{4.19}$$

$$\langle w . x_i \rangle + b \leq -1 + \xi_i \; for \; y_i = -1 \tag{4.20}$$

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

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{M} \xi_i \tag{4.21}$$

Subject to $\qquad y_i(\langle w . x_i \rangle + b) + \xi_i - 1, \xi_i \geq 0, i = 1, \dots, M$

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.

$$Max_{\alpha_i} L_D(\alpha_i) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2}\sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j \langle x_i . x_j \rangle \tag{4.22}$$

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. (4.22) is the same as Eq. (4.15) 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 (Lin et al., 2008). 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. (4.15) (Pham et al., 2007). The Lagrangian maximization problem becomes:

$$Max_{\alpha_i} L_D(\alpha_i) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2}\sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j k \langle x_i . x_j \rangle \tag{4.23}$$

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

Here, the kernel function used is RBF which is shown below:

$$K(x_i, x) = exp^{(-\gamma\|x_i - x\|^2)} \tag{4.24}$$

The classification decision function formula for testing data becomes:

$$sgn(w^T \phi(x) + b) = sgn(\sum_{i=1}^{M} \alpha_i \, y_i K(x_i, x) + b) \tag{4.25}$$

where $\{\alpha_i\}_{i=1}^{i}$ is Lagrange Multipliers which is computed according to pseudo code express in Appendix A and the bias (b) is compute using the following formula:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (w. x_i - y_i) \tag{4.26}$$

where N_{SV} is the number of support vectors x_i that has α_i greater than zero.

After obtaining the sign for the testing set instance, the accuracy test is computed using Eq. (4.5) which is computed for only one fold testing set and after finishing computing the all *k*-fold testing sets; the classification accuracy is computed using Eq. (4.6). Figure 4.8 depicts the algorithm's steps for binary class SVM.

```
Binary Class SVM's Algorithm
Input: Optimal Value for C, Optimal Value for γ, and Features Subset
Output: Classification Accuracy
Begin
        Compute RBF kernel matrix for training sets using Eq. (4.24)
        Call $\alpha_i$ routine in Appendix A
        Compute weight w for training sets' features using Eq. (4.18)
        Compute b for training sets using Eq. (4.26)
        Classify testing set using Eq. (4.25)
        Compute test accuracy using Eq. (4.5)
        Compute cross validation accuracy using Eq. (4.6)
End
```

Figure 4.8 Binary class SVM's algorithm

For multi class SVM, One-Against-One (OAO) strategy is used in this thesis, because it is more accurate compared with One-Against-All (OAA). OAO constructs a number of binary SVM classifiers where each one is trained on data from two classes. The number of binary SVM classifiers is computed according to the following formula:

$$C_2^v = \frac{v(v-1)}{2} \tag{4.27}$$

In classification, a voting strategy is used, each binary classification is related to be a vote where votes can be direct for all data points; ultimately, a point is considered to be in the class with the maximum number of votes. In cases where two classes have identical votes, the class appearing first in the array of storing class names is chosen (Hsu & Lin, 2002a; Hsu & Lin, 2002b; and Chang & Lin, 2011).

Subsequently, if the termination conditions are satisfied the algorithm will be stopped, otherwise it will be returned to tune SVM parameters by using ACO variants. Figure 4.9 depicts the algorithm steps for a multi-class SVM.

```
Multi Class SVM's (One-Against-One) Algorithm
Input: Optimal Value for $C$, Optimal Value for $\gamma$, and Features Subset
Output: Classification Accuracy
Begin
        Compute number of binary classes SVM using Eq. (4.27)
        Call Figure 4.8 binary class SVM to train number of binary classes
        Predicting all binary SVMs for a testing data
        Select the one with the largest vote
End
```

Figure 4.9 Multi class SVM's OAO algorithm

4.2 Experiment and Result

The proposed algorithms were tested on ten datasets from the UCI repository using a ten-fold Cross Validation (CV) technique. The experiment repeats CV technique ten times and the program runs ten times to enable each fold of data to take a turn as the testing dataset, producing ten individual sets of performance statistics, such as classification accuracy. Finally, the experiment averages these performance statistics and calculates the standard deviations for each of the performance statistics.

C programming language was used to implement the proposed algorithms. Experiments were performed on an Intel(R) Core (TM) 2 Duo CPU T5750, running at 2.00 GH$_{\text{Z}}$ with 4.00 GB RAM and 32-bit operating system.

The search range for C was $[2^{-1}, 2^{12}]$ and γ [2-12, 22] (Huang, 2009). The number of tested ants was (2, 4, 6, 8, and 10). The results show that the performance of the proposed algorithms did not depend on number of ants, therefore, only two ants were selected to execute the proposed algorithms. The value of tested q variable which is the algorithm’s parameter to control diversification of search process was (0.1, 0.3, 0.5, 0.7, and 0.9) and the results show that when the q value is small it gives a better result, so the q value was selected to be equal to 0.1.

The performance of the proposed algorithms is evaluated by comparing it with GA$_{\text{with feature chromosome}}$-SVM and Grid search (Zhao et al., 2011), ACO-SVM (Huang, 2009), PSO-SVM and GA-SVM (Lin et al., 2008), and SA-SVM (Lin et al., 2008).

Table 4.2-4.5 summarizes the performance statistics for ACO_R-SVM, $IACO_R$-SVM, ACO-SVM, $GA_{with\ feature\ chromosome}$-SVM and Grid search (Zhao et al., 2011), PSO-SVM and GA-SVM (Lin et al., 2008), and SA-SVM (Lin et al., 2008).

Table 4.2 and Figure 4.10 compare the classification accuracy between the proposed ACO_R-SVM and $IACO_R$-SVM algorithms and $GA_{with\ feature\ chromosome}$-SVM, Grid search, PSO-SVM, GA-SVM, and SA-SVM. The comparison result shows that the proposed $IACO_R$-SVM algorithm achieves highest classification accuracy in all datasets, while ACO_R-SVM achieves highest classification accuracy in nine datasets and only in one dataset, the Ionosphere dataset, $GA_{with\ feature\ chromosome}$-SVM was better than ACO_R-SVM in 0.81% while the performance of the proposed ACO_R-SVM algorithm was similar to PSO-SVM and SA-SVM for the same dataset. The reason the proposed algorithms performed better than other approaches is because that the proposed algorithms tune SVM parameters without the need to discretize the continuous value. It shows that the proposed algorithms can handle categorical, real and integer values. Also, the experiment results show that the proposed $IACO_R$-SVM algorithm was better than ACO_R-SVM in the Pima-India Diabetes, Australian, German, and Heart datasets, while in other datasets (Image Segmentation, Splice, Ionosphere, Iris, Sonar, and Vehicle datasets) the performance of the proposed ACO_R-SVM algorithms and $IACO_R$-SVM algorithms was similar.

Table 4.2: Comparison of Classification Accuracy

Datasets	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Australian	91.99 ± 0.29	94.62 ± 0.52	86.81 ± 3.64	88.09	88.34	88.09	84.74 ± 4.52
German	88.72 ± 0.04	91.56 ± 0.1	80.80 ± 2.10	79.00	-	84.24	78.9 ± 1.73
Heart	94.87 ± 0.4	96.28 ± 0.21	91.11 ± 2.58	-	-	-	88.15 ± 5.18
Ionosphere	97.76 ± 0.55	98.90 ± 0.52	98.57 ± 2.02	97.50	97.50	96.61	94.29 ± 3.56
Iris	99.87 ± 0.29	99.86 ± 0.29	96.00 ± 3.44	98.00	-	97.56	94.09 ± 4.77
Pima-India Diabetes	88.00 ± 0	91.25 ± 0.13	81.97 ± 5.34	80.19	80.19	82.98	76.58 ± 5.14
Sonar	98.36 ± 0.63	98.92 ± 0.08	95.00 ± 2.36	88.32	91.85	95.22	90.50 ± 8.32
Vehicle	93.00 ± 0	93.20 ± 0.18	84.74 ± 2.32	88.71	88.76	85.87	83.94 ± 2.74

1 Proposed ACO_R-SVM Algorithm
2 Proposed $IACO_R$-SVM Algorithm
3 $GA_{\text{with feature chromosome}}$-SVM
4 PSO-SVM
5 SA-SVM
6 GA-SVM
7 Grid

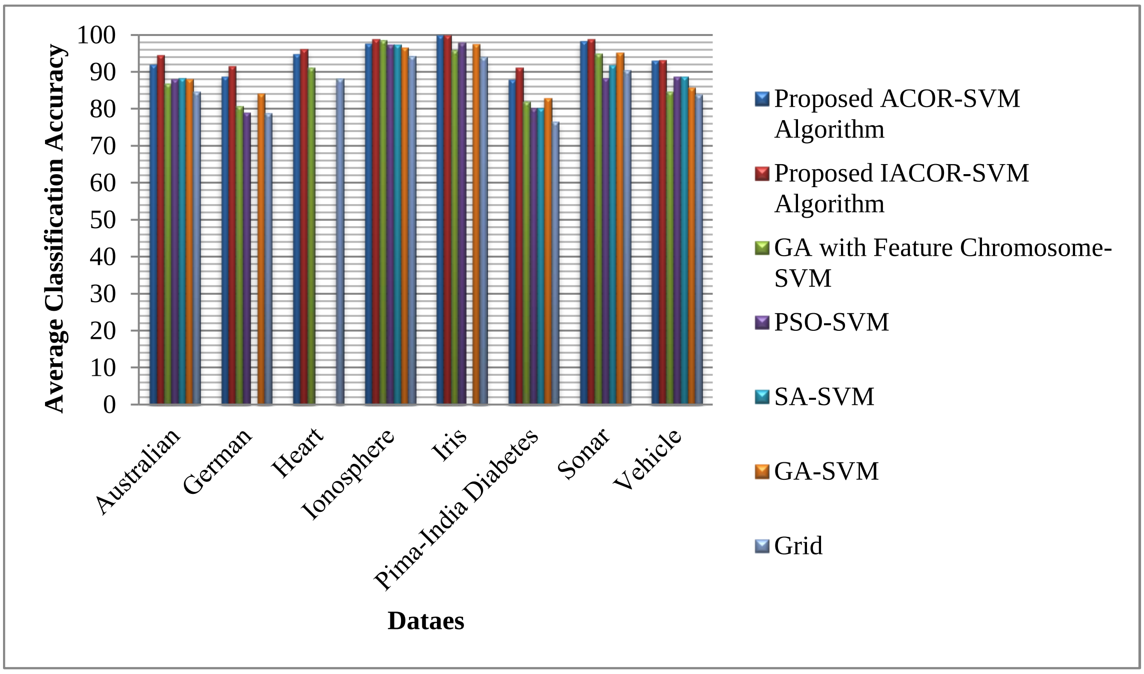

Figure 4.10 Comparison of classification accuracy

Table 4.3 and Figure 4.11 compare the classification accuracy between the proposed ACO_R-SVM and $IACO_R$-SVM algorithms and ACO-SVM. The comparison results show that the proposed algorithms are better than ACO-SVM and achieve higher classification accuracy in all datasets as the proposed algorithms handle the continuous value of SVM parameters directly without the need to discretize them. It also shows that the proposed algorithms can handle categorical, real and integer values.

Table 4.3: Percentage Accuracy of Average Classification

Datasets	**Proposed ACO_R-SVM algorithm**	**Proposed $IACO_R$-SVM algorithm**	**ACO-SVM**
Image Segmentation	98.00 ± 0	97.29 ± 0.08	54.64
Pima-India Diabetes	88.00 ± 0	91.25 ± 0.13	63.57
Splice	96.72 ± 0.18	96.76 ± 0.48	64.73

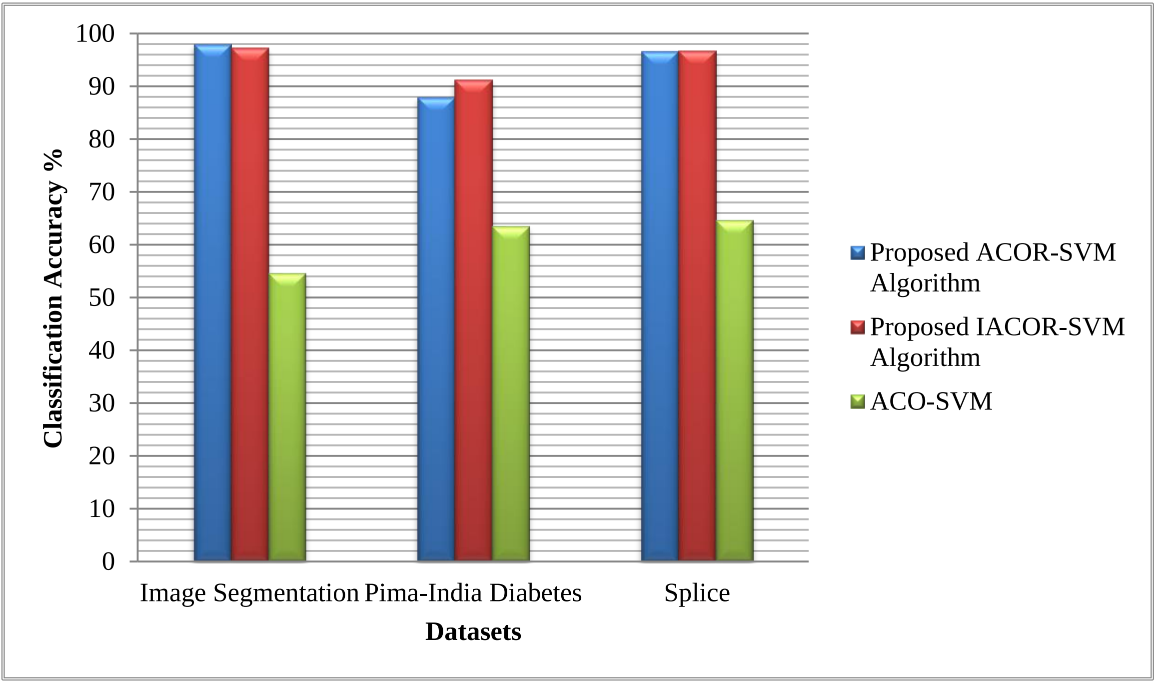

Figure 4.11 Percentage accuracy of average classification

Table 4.4 and Figure 4.12 compare the size of feature subset between the proposed ACO_R-SVM and $IACO_R$-SVM algorithms and $GA_{with\ feature\ chromosome}$-SVM, Grid search, PSO-SVM, GA-SVM, and SA-SVM. The comparison results show that the proposed ACO_R-SVM algorithms and $IACO_R$-SVM algorithms achieve the smallest feature subset size for nine datasets, while in only one dataset, the Iris dataset, $GA_{with\ feature\ chromosome}$-SVM approach achieves the smallest feature subset size. Both of the proposed algorithms produce similar feature subset size; this is because both the proposed algorithms use the same technique to select features. The smallest reduction in feature size was 48.33% for the Vehicle dataset, while the highest reduction in feature size was 80.82% for the Splice dataset.

Table 4.4: Comparison of Feature Subset Size

Datasets	**Original number of features**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Australian	11	3.2 ± 0.4	3.3 ± 0.46	6.7 ± 3.16	15	15	14	14
German	20	6.2 ± 0.77	6.4 ± 0.63	11.8 ± 3.33	30	-	24	24
Heart	13	6.2 ± 0.6	6.0 ± 0.45	7.0 ± 1.05	-	-	-	13
Ionosphere	34	11.0 ± 1	10.6 ± 1.74	15.4 ± 3.32	34	34	34	34
Iris	4	2 ± 0	2 ± 0	1.8 ± 0.38	4	-	4	4
Pima-India Diabetes	8	2.6 ± 0.49	2.6 ± 0.49	5.1 ± 1.63	8	8	8	8
Sonar	60	21.5 ± 1.12	21.7 ± 1.42	28.7 ± 4.00	60	60	60	60
Vehicle	18	8.9 ± 1.04	9.3 ± 1.0	10.3 ± 2.72	18	18	18	18

1 Proposed ACO_R-SVM Algorithm
2 Proposed $IACO_R$-SVM Algorithm
3 $GA_{\text{with feature chromosome}}$-SVM
4 PSO-SVM
5 SA-SVM
6 GA-SVM
7 Grid

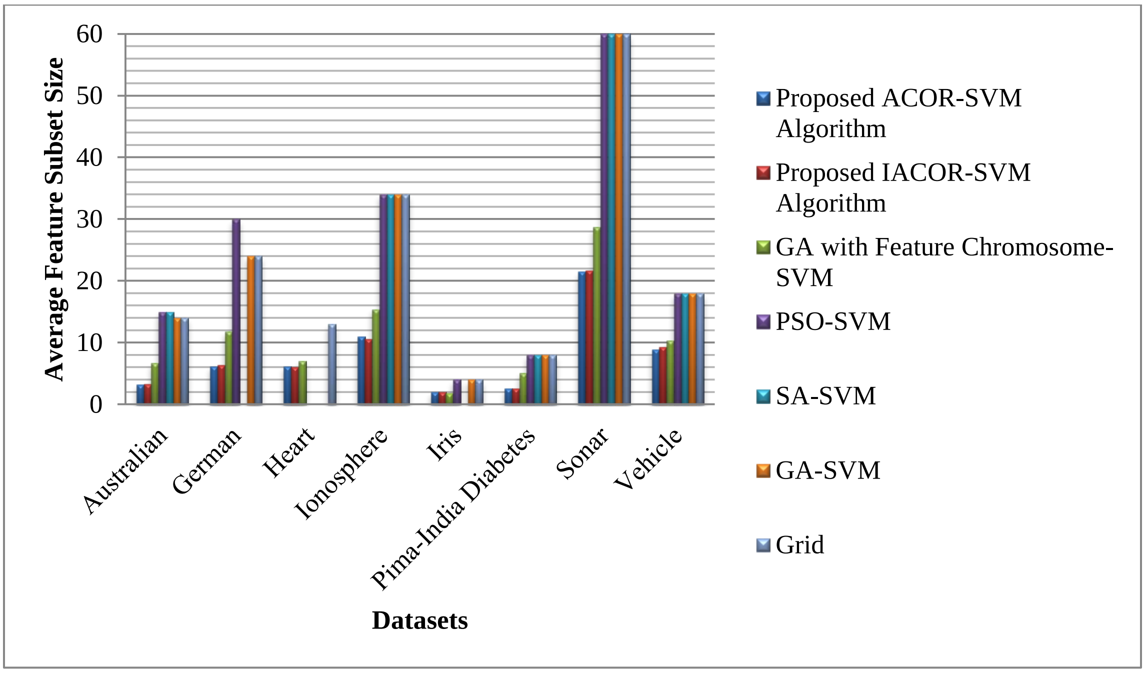

Figure 4.12 Comparison of feature subset size

Table 4.5 and Figure 4.13 compare the size of feature subset between the proposed ACO$_R$-SVM and IACO$_R$-SVM algorithms and ACO-SVM. The comparison results show that the proposed ACO$_R$-SVM algorithms and IACO$_R$-SVM algorithms achieve the smallest feature subset size for all datasets.

Table 4.5: Average Feature Subset Size

Datasets	**Original number of features**	**Proposed ACO$_R$-SVM algorithm**	**Proposed IACO$_R$-SVM algorithm**	**ACO-SVM**
Image Segmentation	18	8.3 ± 0.46	8.3 ± 0.46	16.7
Pima-India Diabetes	8	2.6 ± 0.49	2.6 ± 0.49	6.1
Splice	61	11.7 ± 1.27	11.7 ± 1	12.3

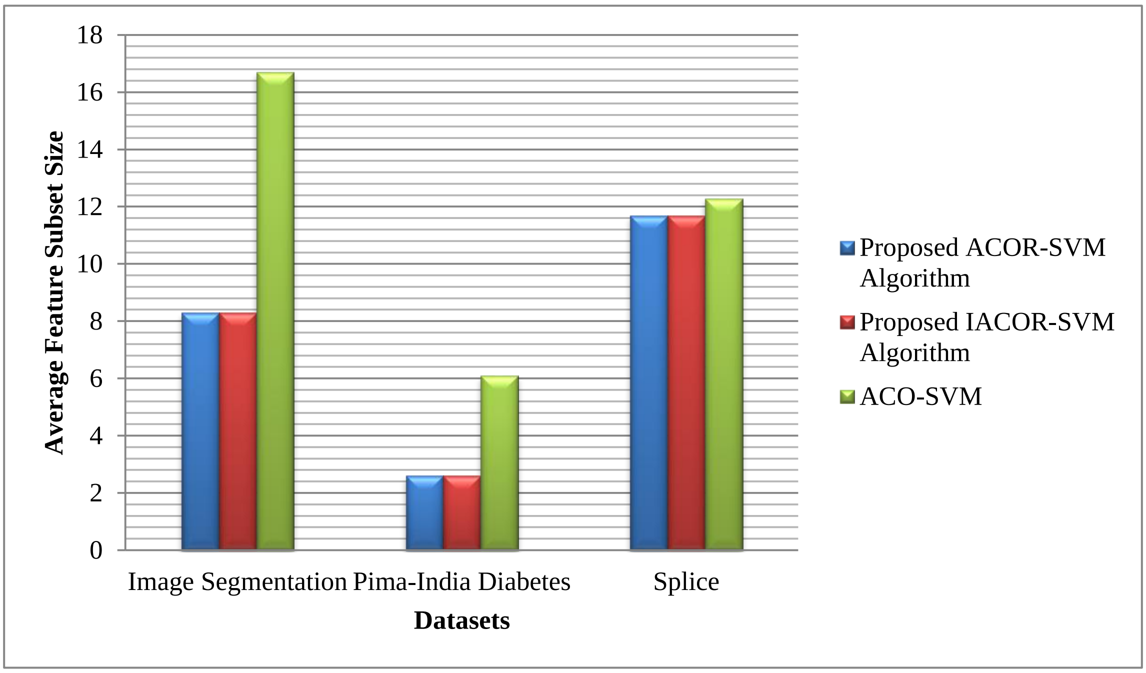

Figure 4.13 Average feature subset size

Table 4.6 shows the *p* value of nonparametric Wilconxon signed-rank test in comparison with the proposed IACO$_R$-SVM algorithm with GA$_{\text{with feature chromosome}}$-SVM, Grid search, PSO-SVM, GA-SVM, and SA-SVM approaches. As shown in Table 4.6 the *p*-value for the Ionosphere dataset of GA$_{\text{with feature chromosome}}$-SVM is greater than the preselected statistical significance level of 0.05, but other *p*-values are lower than the significance level of 0.05. Generally, the proposed IACO$_R$-SVM algorithm has higher classification accuracy.

Table 4.6: p-value for Wilcoxon test between Proposed $IACO_R$*-SVM*[1] *Algorithm,* $GA_{with\ feature\ chromosome}$*-SVM*[2]*, PSO-SVM*[3]*, SA-SVM*[4]*, GA-SVM*[5] *and Grid*[6]

Datasets	**1 & 2**	**1 & 3**	**1 & 4**	**1 & 5**	**1 & 6**
Australian	0.0025	0.0025	0.0025	0.0025	0.0025
German	0.0025	0.0025	-	0.0025	0.0025
Heart	0.0025	-	-	-	0.0025
Ionosphere	0.1013	0.0025	0.0025	0.0025	0.0025
Iris	0.0025	0.0025	-	0.0025	0.0025
Pima-India Diabetes	0.0216	0.0216	0.0216	0.0216	0.0216
Sonar	0.0025	0.0025	0.0025	0.0025	0.0025
Vehicle	0.0025	0.0025	0.0025	0.0025	0.0025

Table 4.7 shows the *p* value of nonparametric Wilconxon signed-rank test with the comparison of the proposed ACO_R-SVM algorithm with $GA_{with\ feature\ chromosome}$-SVM, Grid search, PSO-SVM, GA-SVM, and SA-SVM approaches. As shown in Table 4.7 the *p*-value for the Ionosphere dataset of $GA_{with\ feature\ chromosome}$-SVM, PSO-SVM, and SA-SVM is greater than the preselected statistical significance level of 0.05, but other *p*-values are lower than the significance level of 0.05. Generally, the proposed ACO_R-SVM algorithm has higher classification accuracy.

Table 4.7: p-value for Wilcoxon test between Proposed ACO_R-SVM^1 Algorithm, $GA_{with\ feature\ chromosome}$-$SVM^2$, PSO-$SVM^3$, SA-$SVM^4$, GA-$SVM^5$ and $Grid^6$

Datasets	**1 & 2**	**1 & 3**	**1 & 4**	**1 & 5**	**1 & 6**
Australian	0.0025	0.0025	0.0025	0.0025	0.0025
German	0.0025	0.0025	-	0.0025	0.0025
Heart	0.0025	-	-	-	0.0025
Ionosphere	0.9975	0.1931	0.1931	0.0025	0.0025
Iris	0.0025	0.0025	-	0.0025	0.0025
Pima-India Diabetes	0.0216	0.0216	0.0216	0.0216	0.0216
Sonar	0.0025	0.0025	0.0025	0.0025	0.0025
Vehicle	0.0025	0.0025	0.0025	0.0025	0.0025

Table 4.8 shows the *p* value of nonparametric Wilconxon signed-rank test of the comparison of the proposed $IACO_R$-SVM and ACO_R-SVM algorithms with ACO-SVM approach. As shown in Table 4.8 the *p*-values for all datasets are smaller than the preselected statistical significance level of 0.05 meaning that the proposed $IACO_R$-SVM and ACO_R-SVM algorithms are better than the ACO-SVM approach in terms of producing high classification accuracy.

Table 4.8: p-value for Wilcoxon test between Proposed $IACO_R$-SVM^1 Algorithm, Proposed ACO_R-SVM^2 Algorithm and ACO-SVM^3

Datasets	**1 & 3**	**2 & 3**
Image Segmentation	0.0025	0.0025
Pima-India Diabetes	0.0216	0.0216
Splice	0.0025	0.0025

Finally, Table 4.9 shows the *p* value of nonparametric Wilconxon signed-rank test with the comparison of the proposed IACO$_R$-SVM and ACO$_R$-SVM algorithms. The *p*-value for the Image Segmentation, Splice, and Iris datasets was greater than the preselected statistical significance level of 0.05, but other *p*-values are lower than the significance level of 0.05. Generally, the proposed IACO$_R$-SVM algorithm has higher classification accuracy compared with ACO$_R$-SVM.

Table 4.9: p-value for Wilcoxon test between Proposed IACO$_R$-SVM^1 Algorithm and Proposed ACO$_R$-SVM^2 Algorithm

Datasets	**1 & 2**
Australian	0.0025
German	0.0025
Heart	0.0025
Image Segmentation	0.9975
Ionosphere	0.0025
Iris	0.0697
Pima-India Diabetes	0.0216
Sonar	0.0025
Splice	0.3994
Vehicle	0.0063

4.3 Summary

This chapter 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. This chapter compared the performance of the proposed algorithms to ACO-SVM, GA$_{\text{with feature chromosome}}$-SVM, Grid search, PSO-SVM, GA-SVM, and SA-SVM concerning classification accuracy and found the proposed algorithms to be the best. Also, the experiments' results show that IACO$_R$-

SVM is better than ACO_R-SVM in four datasets. Moreover, the experiments in this chapter found that the size of features selected by the proposed algorithms were smaller than the compared approaches in nine datasets while in only one dataset, $GA_{\text{with feature chromosome}}$-SVM produced the smaller size for the Iris dataset. The size of feature subset selected by ACO_R-SVM and $IACO_R$-SVM were almost similar because both the proposed algorithms used same technique to select feature subset. However, ACO_R and $IACO_R$ are able to help SVMs to enhance performance in addition to the proposed technique being used to select feature subset.

CHAPTER FIVE

ENHANCED ACO-SVM PATTERN CLASSIFICATION ALGORITHM FOR SIMULTANEOUS SVM PARAMETER OPTIMIZATION AND FEATURE SUBSET SELECTION

This chapter presents the proposed algorithms for simultaneously optimizing Support Vector Machine (SVM) parameters and feature subset selection. Section 5.1 presents the proposed two algorithms while Section 5.2 presents the experiments carried out on ten datasets from the University of California, Irvine (UCI) repository to evaluate the performance of the proposed algorithms. Finally, the summary of the chapter is present in Section 5.3.

5.1 Enhanced Pattern Classification Algorithm to Simultaneously Optimize SVM Parameter and Feature Subset Selection

In this thesis, two algorithms are proposed in order to simultaneously optimize feature subset selection and SVM parameters. Other approaches that simultaneously optimize feature subset selection and SVM parameters differ from these two proposed algorithms in the section related to optimizing SVM parameters. Other approaches discretize the continuous value of SVM parameters which will be affected on classification accuracy through maximizing the classification error. Figure 5.1 depicts the general flowchart of the other SVM pattern classification approaches that simultaneously optimize SVM parameters and feature subset selection.

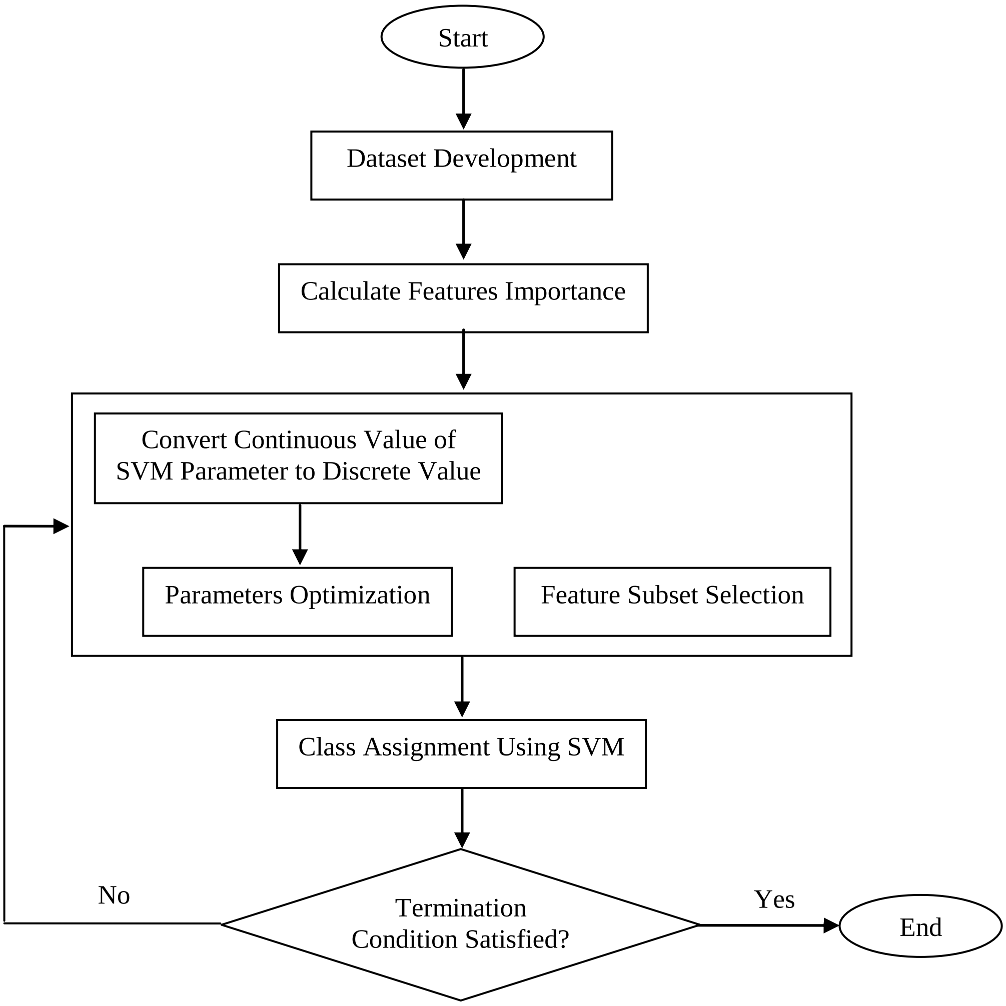

Figure 5.1 Flowchart for SVM pattern classification approach for simultaneous optimization

In this thesis, two algorithms are proposed based on new mixed-variable Ant Colony Optimization variants: mixed-variable Ant Colony Optimization (ACO_{MV}) (Socha, 2008) and Liao et al.'s (2011) suggestion of Incremental mixed-variable Ant Colony Optimization ($IACO_{MV}$) in order to simultaneously obtain the optimal continuous value for SVM parameters and feature subset to classify patterns with acceptable classification accuracy.

These two proposed algorithms work directly with the continuous value of SVM parameters without the need to discretize it and convert it from continuous values to discrete values and then optimize the discrete values. Also, the two proposed algorithms differ from the two proposed algorithms presented in Chapter Four in that they can simultaneously optimize SVM parameters and feature subset selection. One of the new ACO research directions is to optimize mixed-variable (continuous and discrete) problems. ACO_{MV} is considered the first algorithm that can handle the mixed-variable and follow the same ACO framework, while $IACO_{MV}$ is suggested by Liao et al. (2011) to improve ACO_{MV} performance in solving stagnation. Figure 5.2 depicts the proposed enhanced ($ACO_{MV\text{-}R}$-SVM and $IACO_{MV\text{-}R}$-SVM) algorithms.

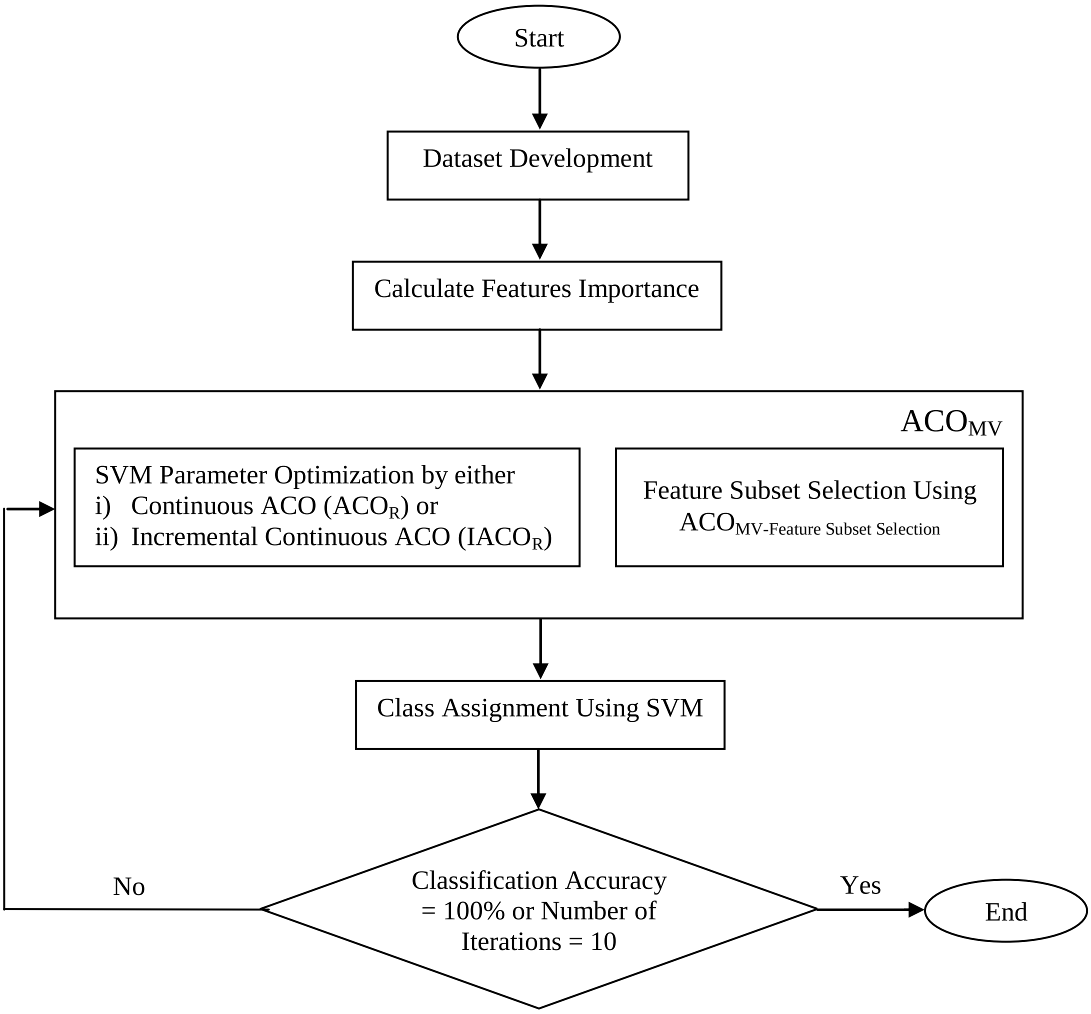

Figure 5.2 Flowchart for enhanced ACO-SVM pattern classification algorithm for simultaneous optimization

Figure 5.3 presents the generic architecture for the proposed enhanced pattern classification algorithm. As shown in this figure, the proposed algorithms start with dataset development and compute the importance of each feature through using Fisher (F)-scores.

The first proposed algorithm will simultaneously select feature subset and tune SVM parameters using continuous ACO (ACO$_R$), while the second proposed algorithm will also simultaneously select feature subset and tune SVM parameters using Incremental ACO_R ($IACO_R$). The output of both proposed algorithms will assign the class by using SVM and compute classification accuracy which will be fed back to ACO algorithms to update solution archives and pheromone tables in cases where the produced classification accuracy is not satisfied. Otherwise, if the classification accuracy is acceptable, the proposed algorithms will terminate.

Details of the ACO_R and $IACO_R$ are explained in Chapter Four. The feature subset selection phase is explained in Section 5.1.2.

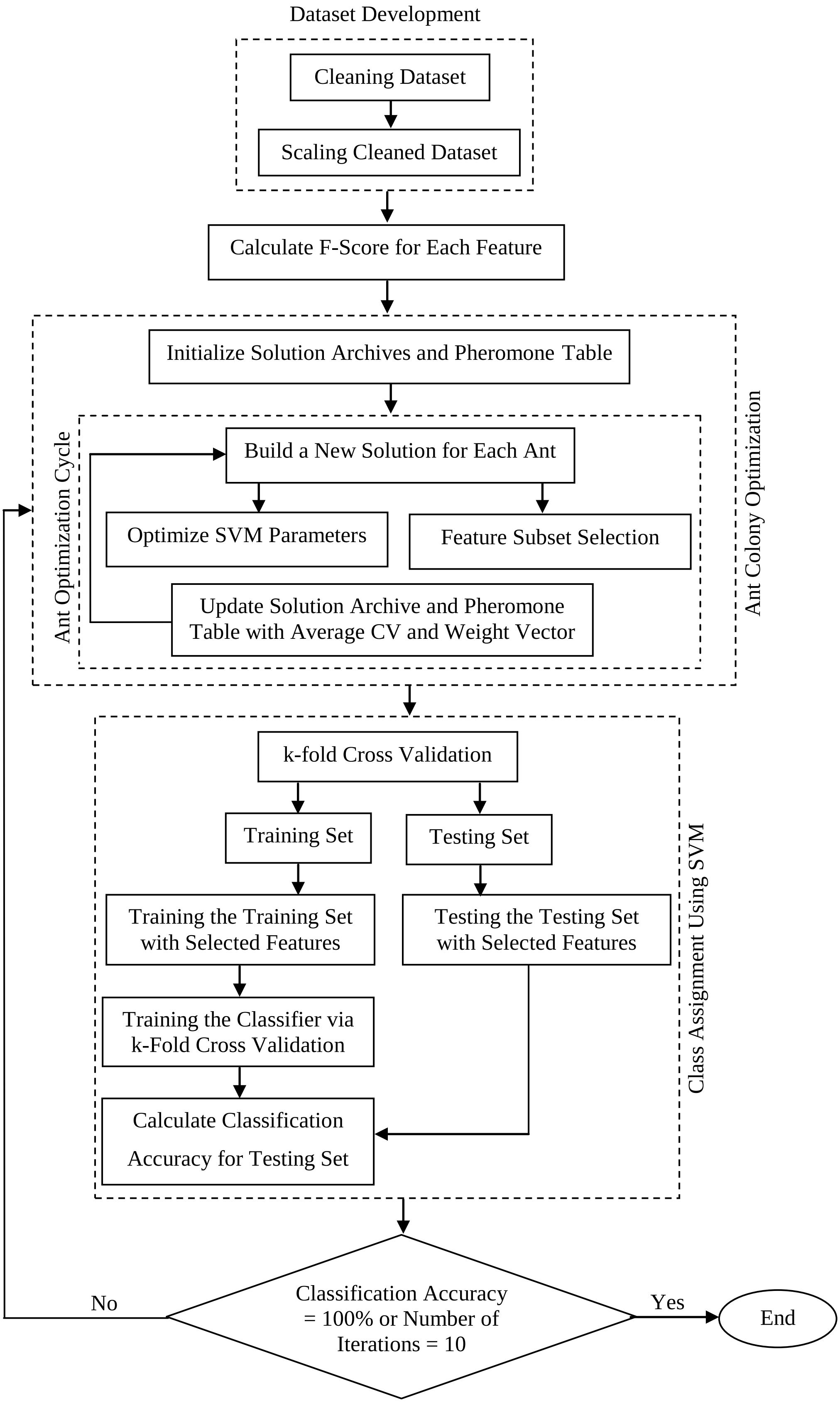

Figure 5.3 Enhanced generic ACO-SVM pattern classification algorithm architecture for simultaneous optimization

Figure 5.4 presents the generic pseudo code for proposed algorithms.

```
Input: Datasets
Output: Optimal value for SVM parameters, Optimal feature subset, and classification accuracy
Begin
        //dataset development
        clean dataset
        scale cleaned dataset
        calculate F-score for each feature using Eq. (4.1)
        for $i$ = 1 to no. of runs do
                for $j$ = 1 to no. of iterations do
                        //simultaneously optimize feature subset selection and tune SVM
                        parameters
                        call $ACO_{MV}$
                end
        end
End
```

Figure 5.4 Generic pseudo code for proposed enhanced ACO-SVM algorithm for simultaneous optimization

5.1.1 Datasets' Development

As with the proposed algorithms in Chapter Four, the proposed algorithms in this chapter also begin with developed datasets as explained in Chapter Three which includes cleaning missing data from datasets and scaling the cleaned dataset using Eq. (3.1) to avoid features with high numerical ranges from dominating those in lower numerical ranges and to minimize the computation efforts. All attributes are linearly scaled to the [0, 1] range.

5.1.2 Proposed Algorithm for Simultaneously Optimizing Support Vector Machine Parameter and Optimizing Feature Subset Selection

The ACO_{MV} algorithm has the ability to optimize mixed-variable problems which are the continuous variables for the SVM parameters and feature subset represented as discrete graph nodes.

In general ACO_{MV} is like ACO_R and $IACO_R$ in keeping generated solutions' values and its fitness function values in solution archives rather than pheromone tables. ACO_{MV} begins by initializing this solution's archives with arbitrary solutions and then orders them according to its fitness function to influence the seek procedure in the direction of the best solutions obtained through seeking. The solutions in the solution archives are often stored on the basis of their fitness function, so that the best solution is always at the top.

ACO_{MV}-SVM starts by initializing three solution archives, one for *C* values, one for γ values, and the third for feature subset. The initialization for *C* and γ values will be randomly and made as in Chapter Four by dividing the range of *C* and γ over parameter *k* which represents the size of the solution archives. The solution archive for a feature subset will be initialized by store and sort features related to their F-score as computed in Eq. (4.1). After initializing these three solution archives, each solution which will include three parts; the first for *C*, the second for γ, and the third for features will be sent to the SVM to compute its objective function and, according to this objective function, the solutions will be sorted in the solution archives.

After that, each ant will start to construct its solution by calling $ACO_{MV\text{-tune SVM parameter}}$ and $ACO_{MV\text{-feature subset selection}}$, because in this thesis ACO_{MV}-SVM consists of two parts. Figure 5.5 depicts the pseudo code of proposed ACO_{MV}-SVM algorithm.

```
Input: features, termination condition, no. of ant ($m$), size of solution archive ($k$), $q$, range of $C$,
       range of $\gamma$, $\alpha$, and $\beta$
Output: classification accuracy, optimal features subset, and optimal value for SVM parameters
Begin
        //initialize solution archive
        $C \rightarrow$ solution archive$_C$
        $\gamma \rightarrow$ solution archive$_\gamma$
        features $\rightarrow$ solution archive$_{\text{feature}}$
        call SVM algorithm to evaluate the initialize solution in solution archive
        while classification accuracy $\neq$ 100% or number of iteration $\neq$ 10 do
                //construct ant solution
                for $n = 1$ to $N_{ants}$ do
                        //construct solution ($S_1, \ldots, S_{Nant}$)
                        //probabilistic solution construction for tuning SVM parameters
                        call ACO$_{\text{MV-tune SVM parameter}}$
                        //probabilistic solution construction for feature subset
                        call ACO$_{\text{MV-feature subset Selection}}$
                        call SVM algorithm to evaluate the newly built solution
                end
                solution archive = first $\leftarrow$ Rank ($S_{old}$ U $S_1, \ldots S_{Nants}$)
                update solution archives
        end
End
```

Figure 5.5 Pseudo code for proposed enhanced ACO$_{MV}$-SVM algorithm for simultaneous optimization

The first part is ACO$_{\text{MV-tune SVM parameter}}$ which relates to the optimization of the continuous value of SVM parameters through using ACO$_R$ or IACO$_R$ as explained in Chapter Four, but it differs from them where an ant is establishing its solution which will included three parts; the first for C value, the second for γ Radial Basis Function (RBF) kernel parameter, and the third for features. C and γ values will be generated according to Eqs. (4.2) and (4.3) and then sampling a selected weight over k, this constructed solution will not be sent to the SVM together with the feature subset generated through the second part ACO$_{\text{MV-feature subset selection}}$ of the ACO$_{MV}$-SVM. Based on the outcome of SVM, the solution archives will update through using the same strategies used in ACO$_R$ or IACO$_R$ as explained in Chapter Four.

The second part of ACO_{MV}-SVM is $ACO_{MV\text{-feature subset selection}}$ which is related to optimization of feature subset selection. The $ACO_{MV\text{-feature subset selection}}$ algorithm that has been used to construct feature subset is shown in Figure 5.6.

```
Input: Features
Output: Optimal feature subset
Begin
        calculate feature subset size randomly
        initialize pheromone table
        for $i$ = 1 to no. of features do
                //select first feature in feature subset
                compute weight for each feature using Eq. (5.2)
                compute probability for each feature using Eq. (5.1)
                select feature with highest probability
                append feature with highest probability to feature subset
                remove appended feature from original features set
        end
        for $j$ = 1 to feature subset size - 1 do
                //select other features in feature subset
                compute probability for remaining features using Eq. (5.4)
                select feature with highest probability
                append feature with highest probability to feature subset
                remove appended feature from original features set
        end
        update pheromone table
End
```

Figure 5.6 Pseudo code of proposed enhanced algorithm for feature subset selection

The $ACO_{MV\text{-feature subset selection}}$ algorithm starts by computing the size of the feature subset for each ant randomly and then initializing a pheromone table using the following formula:

$$phero_{feature_i feature_j} = \frac{1}{\sum_{i=1}^{no.of\ feature} F\text{-}score_{feature}} \tag{5.1}$$

After that, the ant will start to construct its feature subset. The first feature in the feature subset will be selected according to its probability as follows:

$$P = \frac{w_{feature_i} * F\text{-}Score_{feature_i}}{\sum_{i=1}^{no.of\ features} w_{feature_i}} \tag{5.2}$$

Where the $F - Score_{feature_i}$ is computed using Eq. (4.1) and $w_{feature_i}$ is the weight of *feature* $feature_i$ and it is computed using the following formula:

$$w_{feature_i} = \frac{w}{u} + \frac{q}{\eta} \tag{5.3}$$

where w is computed using Eq. (5.4), u as a counter that counts how many times $feature_i$ is selected, q as in Chapter Four, is the algorithm's parameter to control diversification of search process, and η is a number of unselected features.

$$w = \frac{1}{qk\sqrt{2\pi}} e^{\frac{-(l-1)^2}{2q^2k^2}} \tag{5.4}$$

where l is as in Chapter Four, it is the index of $solution_l$ in the solutions archive which is dependent on its fitness function, and k is the number of features.

The reason for using the probability function as shown in Eq. (5.2) instead of using the standard established discrete probability is because there is a need to traverse from continuous variables, (SVM parameters), to discrete, (feature subset), variables.

After selecting the first feature in the feature subset as explained above, the ant continues to build its feature subset by selecting other features and appending them to the feature subset. The selection of other features is completed through computing the probability for each of the features as follows:

$$Prob_{ij}^{k}(t) = \begin{cases} \frac{(phero_{feature_i feature_j})^{\alpha} (F\text{-}Score_{feature_i})^{\beta}}{\sum_{j \in I_i^k} (phero_{feature_i feature_j})^{\alpha} (F\text{-}Score_{feature_i})^{\beta}} & if\ j \in I_i^k \\ 0 & otherwise \end{cases} \tag{5.5}$$

where $phero_{feature_i feature_j}$ is the pheromone value on the edge that connects *feature*$_i$ and *feature*$_j$ and the $F - Score_{feature_i}$ is computed using Eq. (4.1).

After all ants finish building the feature subset, the pheromone will be updated using the following formula:

$$phero_{feature_i feature_j}(t+1) = p * phero_{feature_i feature_j} + \sum_{k=1}^{no.of\ ants} \Delta phero_{feature_i feature_j}{}^{k}(t) \qquad (5.6)$$

where p is a random number generated in the range of (0, 1), $phero_{feature_i feature_j}$ is the current pheromone on the edge that connects *feature*$_i$ and *feature*$_j$, and $\Delta phero_{feature_i feature_j}{}^{k}$ is computed as following:

$$\Delta phero_{feature_i feature_j}{}^{k} = \begin{cases} CVACC^{K} * weight^{k}_{feature_i} * weight^{k}_{feature_j} & if\ ant\ k\ use\ edge\ (feature_i feature_j) \\ 0 & otherwise \end{cases} \qquad (5.7)$$

where $CVACC^k$ is the cross validation classification accuracy generated by SVM from *ant*$_k$ solution, $weight^{k}_{feature_i}$ and $weight^{k}_{feature_j}$ is the weight of *feature*$_i$ and *feature*$_j$ respectively generated from SVM on the training set from *ant*$_k$ solution.

Ultimately, the ant solution will include three parts; the first two parts will be related to SVM parameters which include the *C* and *γ* values, and the third part is related to feature subset.

5.1.3 Class Assignment Using SVM and Evaluation

The class assignment using SVM and to evaluate its performance, here in this chapter, is as explained in Chapter Four which includes computing the RBF kernel matrix, Q matrix, α_i , weight w for training sets' features, b for training sets, and then classifying the testing set using and computing test accuracy and cross validation accuracy.

5.2 Experiment and Result

The proposed algorithms were tested on ten datasets from the UCI repository using a ten-fold Cross Validation (CV) technique. The experiment repeats the CV technique ten times and the program runs ten times to enable each fold of data to take a turn as the testing dataset, producing ten individual sets of performance statistics, such as classification accuracy. Finally, the experiment averages these performance statistics and calculates the standard deviations for each of the performance statistics.

C programming language was used to implement the proposed algorithms. 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.

In these experiments, the search range for C was $[2^{-1}, 2^{12}]$ and γ [2-12, 22]. The number of ants and q value used was the same as in Chapter Four, where number of ants = 2 and $q = 0$. $\alpha = 1$ and $\beta = 2$ are the recommended values according to Dorigo & Stützle (2004). Finally the *Growth* test value was (2, 4, 6, 8, and 10). The results show that the best value was 5. While for *Stag* test the value was (1, 2, 3, 4, and 5) and the results show that the best value is 2. The Initial solution archive size test value was (2, 4, 6, 8, 10, 12, and 14). The result shows that the best value was 10, while for Maximum solution archive size test the value was (3, 5, 7, 9, 11, 13, and 15) and the results show that the best value was 15.

The performance of the proposed algorithms is evaluated by comparing it with Gravitational Search Algorithm (GSA)-SVM (Sarafrazi & Pour, 2013), Genetic Algorithm $(GA)_{\text{with feature chromosome}}$-SVM (Zhao et al., 2011), Ant Colony Optimization (ACO)-SVM (Huang, 2009), Cat Swarm Optimization (CSO)-SVM (Lin & Chien, 2009), Clonal Selection Algorithm (CSA)-SVM (Ding & Li, 2009), Particle Swarm Optimization (PSO)-SVM (Lin et al., 2008), Simulated Annealing (SA)-SVM (Lin et al., 2008), and GA-SVM (Huang & Wang, 2006). Table 5.1-5.8 summarizes the performance statistics for $ACO_{MV\text{-}R}$-SVM, $IACO_{MV\text{-}R}$-SVM, GSA-SVM (Sarafrazi & Pour, 2013), $GA_{\text{with feature chromosome}}$-SVM (Zhao et al., 2011), ACO-SVM (Huang, 2009), CSO-SVM (Lin & Chien, 2009), CSA-SVM (Ding & Li, 2009), PSO-SVM (Lin et al., 2008), SA-SVM (Lin et al., 2008), and GA-SVM (Huang & Wang, 2006).

Table 5.1 and Figure 5.7 compares the classification accuracy between the proposed $ACO_{MV\text{-}R}$-SVM and $IACO_{MV\text{-}R}$-SVM algorithms with CSA-SVM, CSO-SVM, PSO-SVM, SA-SVM, and GA-SVM. The comparison results show that the proposed $IACO_{MV\text{-}R}$-SVM algorithm achieves highest classification accuracy in nine datasets, while in only one dataset, the Iris dataset, $\text{IACO}_{\text{MV-R}}$-SVM was comparative with $\text{GA}_{\text{with feature Chromosome}}$-SVM, CSA-SVM, and GA-SVM. On the other hand, the proposed $\text{ACO}_{\text{MV-R}}$-SVM algorithm achieves the highest classification accuracy in eight datasets, while in only two datasets, where the Image Segmentation dataset, $\text{GA}_{\text{with feature Chromosome}}$-SVM achieves higher classification accuracy; and the Iris dataset, $\text{ACO}_{\text{MV-R}}$-SVM is comparative with $\text{GA}_{\text{with feature Chromosome}}$-SVM, CSA-SVM, and GA-SVM. The proposed algorithms performed better than other approaches because the proposed algorithms simultaneously tune SVM parameters without the need to discretize the continuous value as well as selecting feature subset. It shows that the proposed algorithms can handle categorical, real and integer values. Also, the experiment results show that the proposed $\text{IACO}_{\text{MV-R}}$-SVM was better than $\text{ACO}_{\text{MV-R}}$-SVM algorithm in Image Segmentation, Pima-India Diabetes, and German datasets, while in other datasets the performance of the proposed $\text{ACO}_{\text{MV-R}}$-SVM algorithm and $\text{IACO}_{\text{MV-R}}$-SVM algorithm was similar.

Table 5.1: Comparison of Classification Accuracy

Datasets	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Australian	96.33 ± 0.91	96.96 ± 0.53	91.59 ± 2.14	90.82	93.77 ± 2.27	91.03	92.19 ± 3.23	88.10 ± 2.25
German	96.16 ± 0.57	97.23 ± 0.46	86.10 ± 1.97	86.40	82.20 ± 2.82	81.62	-	85.60 ± 1.96
Heart	97.70 ± 0.93	98.01 ± 0.35	95.56 ± 2.34	92.59	97.04 ± 2.34	-	-	94.80 ± 3.32
Ionosphere	99.86 ± 0.25	99.99 ± 0.02	99.43 ± 1.21	98.56	99.43 ± 1.2	99.01	99.07 ± 0.73	98.56 ± 2.03
Iris	99.95 ± 0.08	99.98 ± 0.03	100 ± 0	100	99.33 ± 2.1	99.20	-	100 ± 0
Pima-India Diabetes	95.07 ± 1.73	97.22 ± 0.81	83.84 ± 5.14	82.70	84.73 ± 5.37	82.68	82.22 ± 3.55	81.50 ± 7.13
Sonar	99.94 ± 0.97	99.99 ± 0.02	99.00 ± 2.11	98.80	98.10 ±3.33	96.26	95.99 ± 3.90	98.00 ± 3.5
Vehicle	93.32 ± 2.20	93.92 ± 0.29	88.24 ± 1.47	90.20	90.77 ± 2.7	89.83	90.14 ± 2.21	84.06 ± 3.54

1 Proposed ACO$_{MV\text{-}R}$-SVM Algorithm 2 Proposed IACO$_{MV\text{-}R}$-SVM Algorithm
3 GA$_{\text{with feature chromosome}}$-SVM 4 CSA-SVM 5 CSO-SVM
6 PSO-SVM 7 SA-SVM 8 GA-SVM

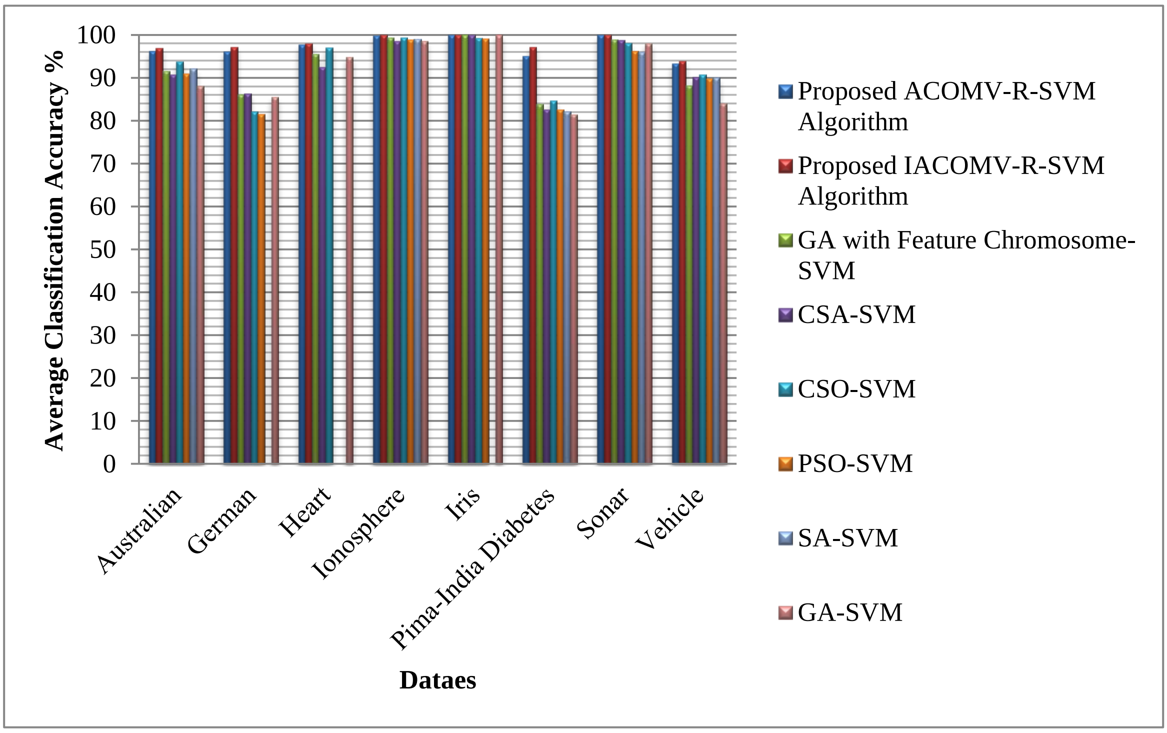

Figure 5.7 Comparison of classification accuracy

Table 5.2 and Figure 5.8 compare the classification accuracy between the proposed $\text{ACO}_{\text{MV-R}}$-SVM and $\text{IACO}_{\text{MV-R}}$-SVM algorithms with ACO-SVM and $\text{GA}_{\text{with feature chromosome}}$-SVM. The results show that the proposed $\text{ACO}_{\text{MV-R}}$-SVM and $\text{IACO}_{\text{MV-R}}$-SVM algorithms achieve best classification accuracy in all datasets compared with ACO-SVM, while when compared with $\text{GA}_{\text{with feature chromosome}}$-SVM, the proposed algorithms were the best in two datasets, while in one dataset, Image Segmentation, the performance of the proposed algorithms was slightly similar to the performance of $\text{GA}_{\text{with feature chromosome}}$-SVM. The proposed algorithms performed better than other approaches because the proposed algorithms simultaneously tune SVM parameters without the need to discretize the continuous value as well as selecting feature subset. It shows that the proposed algorithms can handle categorical, real and integer values. Also, the comparison results show that the proposed algorithm IACO$_{MV-R}$-SVM was better than the performance of the proposed algorithm ACO$_{R}$-SVM.

Table 5.2: Average Classification Accuracy %

Datasets	**Proposed ACO$_{MV-R}$-SVM algorithm**	**Proposed IACO$_{MV-R}$-SVM algorithm**	**GA$_{\text{with feature chromosome}}$-SVM**	**ACO-SVM**
Image Segmentation	97.5 ± 2.5	98.96 ±0.41	98.12	94.76
Pima-India Diabetes	95.07 ± 1.73	97.22 ±0.81	83.84	76.28
Splice	97.90 ± 0.68	98.65 ±0.55	90.53	94.65

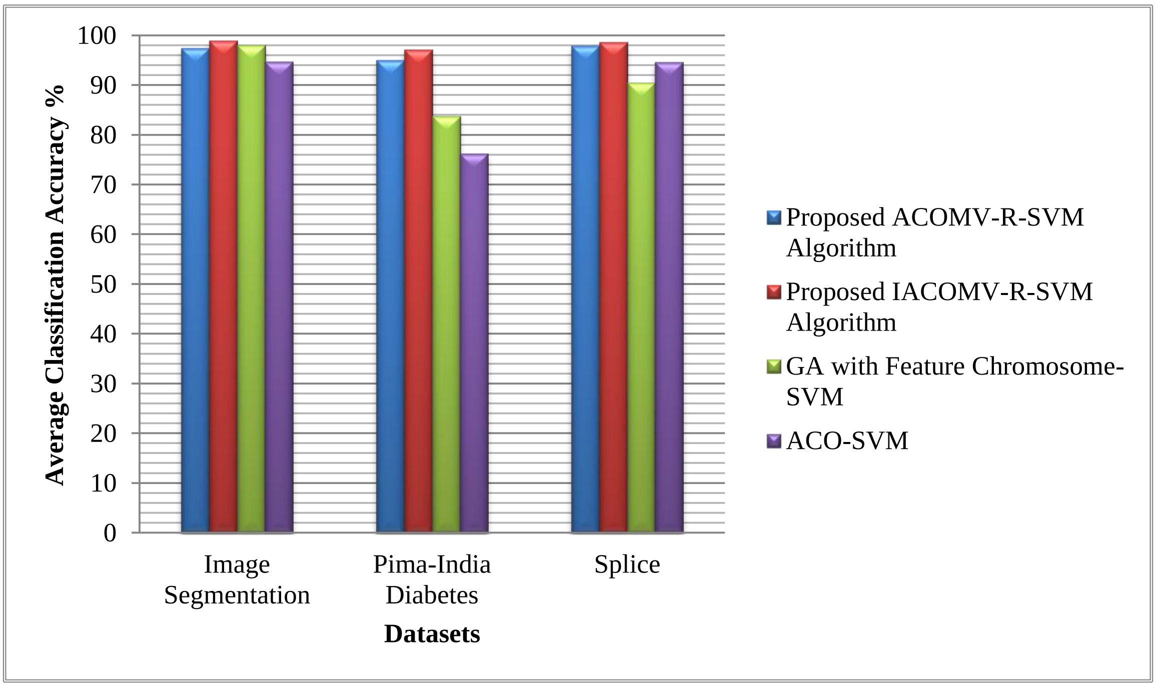

Figure 5.8 Average classification accuracy %

Table 5.3 and Figure 5.9 compare the classification accuracy between the proposed $ACO_{MV\text{-}R}$-SVM and $IACO_{MV\text{-}R}$-SVM algorithms with GSA-SVM. The results show that the performance of the proposed algorithms was slightly better than GSA-SVM for only Ionosphere dataset. The results also showed that the proposed algorithms performed well on datasets with big number of features.

Table 5.3: Comparison between Proposed Algorithm and GSA-SVM

Datasets	**Proposed $ACO_{MV\text{-}R}$-SVM algorithm**	**Proposed $IACO_{MV\text{-}R}$-SVM algorithm**	**GSA-SVM**
Australian	96.33 ± 0.91	96.96 ± 0.53	98.88 ± 0.62
German	96.16 ± 0.57	97.23 ± 0.46	98.25 ± 0.97
Ionosphere	99.86 ± 0.25	99.99 ± 0.02	99.45 ± 0.31
Pima-India Diabetes	95.07 ± 1.73	97.22 ± 0.81	98.90 ± 0.61

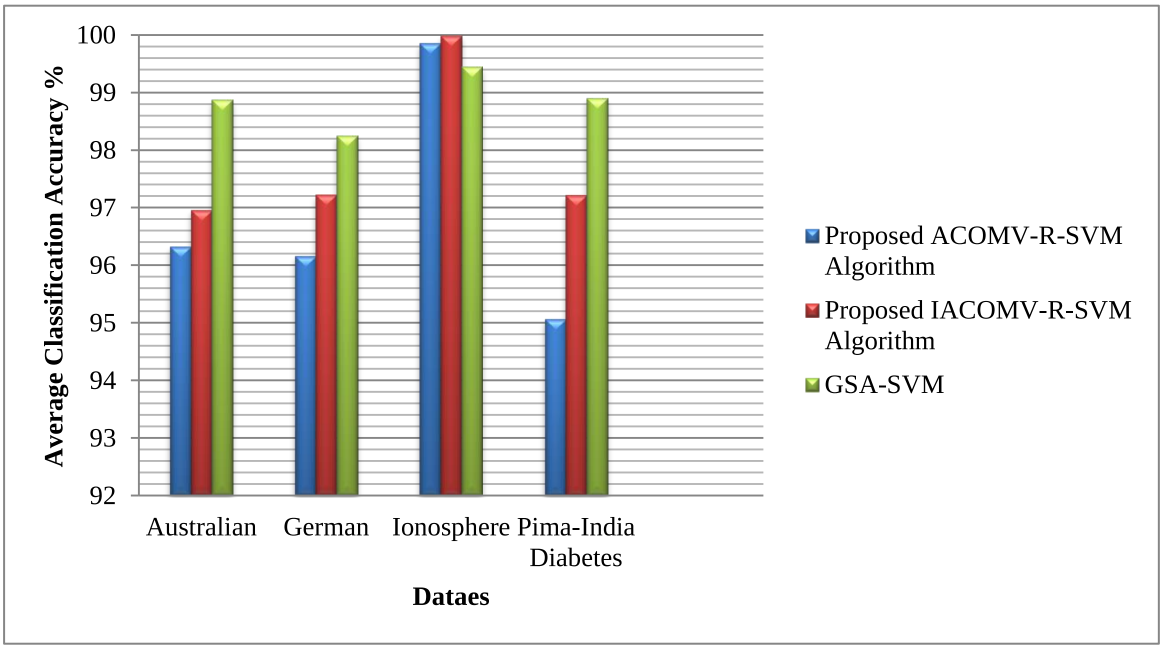

Figure 5.9 Comparison between proposed algorithm and GSA-SVM

Table 5.4 and Figure 5.10 compare the proposed ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM algorithms with the proposed ACO_R-SVM and $IACO_R$-SVM algorithms. The comparison results between the four proposed algorithms show that the results produced by the proposed $IACO_{MV-R}$-SVM algorithm were better than other proposed algorithms' results in all ten datasets and in some datasets, Iris and Vehicle datasets, the results slightly differ than other proposed algorithms because the proposed ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM algorithms can simultaneously tune SVM parameters and select feature subset.

Table 5.4: Comparison of Proposed Algorithm

Datasets	**1**	**2**	**3**	**4**
Australian	91.99 ± 0.29	94.62 ± 0.52	96.33 ± 0.91	96.96 ± 0.53
German	88.72 ± 0.04	91.56 ± 0.1	96.16 ± 0.57	97.23 ± 0.46
Heart	94.87 ± 0.4	96.28 ± 0.21	97.70 ± 0.93	98.01 ± 0.35
Image Segmentation	98.00 ± 0	97.29 ± 0.08	97.5 ± 2.5	98.96 ±0.41
Ionosphere	97.76 ± 0.55	98.90 ± 0.52	99.86 ± 0.25	99.99 ± 0.02
Iris	99.87 ± 0.29	99.86 ± 0.03	99.95 ± 0.08	99.98 ± 0.29
Pima-India Diabetes	88.00 ± 0	91.25 ± 0.13	95.07 ± 1.73	97.22 ± 0.81
Splice	96.72 ± 0.18	96.76 ± 0.48	97.90 ± 0.68	98.65 ±0.55
Sonar	98.36 ± 0.63	98.92 ± 0.08	99.94 ± 0.97	99.99 ± 0.02
Vehicle	93.00 ± 0	93.20 ± 0.18	93.32 ± 2.20	93.92 ± 0.29

1 Proposed ACO_R-SVM Algorithm
2 Proposed $IACO_R$-SVM Algorithm
3 Proposed ACO_{MV-R}-SVM Algorithm
4 Proposed $IACO_{MV-R}$-SVM Algorithm

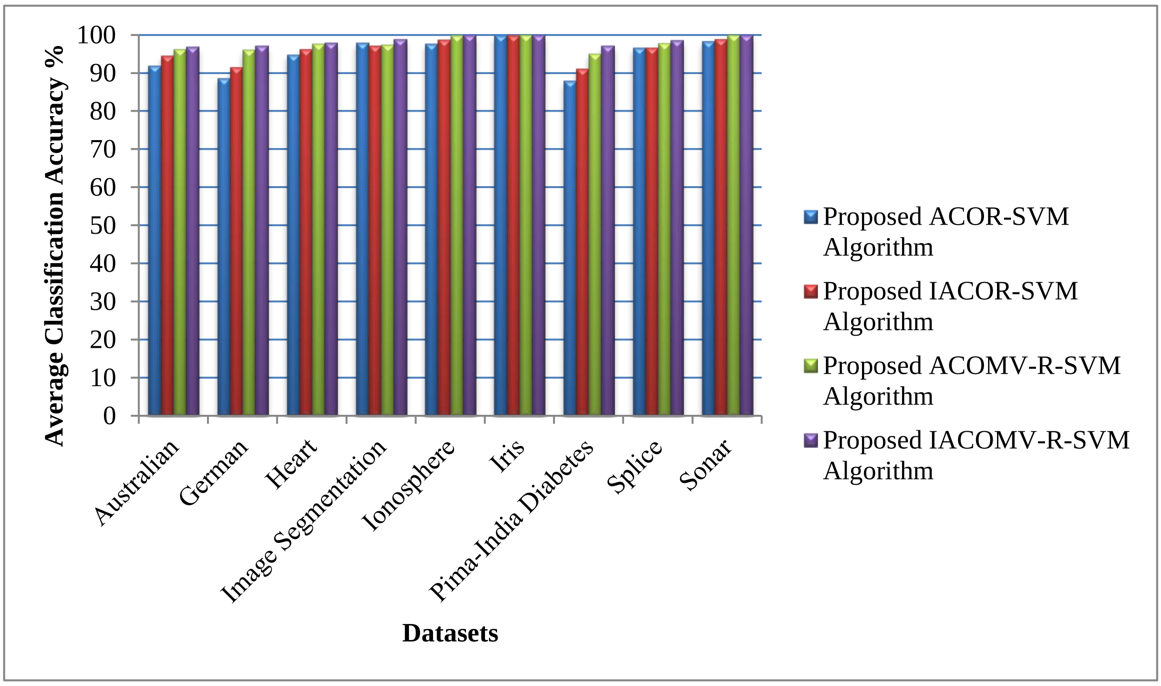

Figure 5.10 Comparison of proposed algorithm

Table 5.5 and Figure 5.11 compare the size of feature subset selection between the proposed ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM algorithms with $GA_{\text{with feature chromosome}}$-SVM, CSA-SVM, CSO-SVM, PSO-SVM, SA-SVM, and GA-SVM. The comparison results show that the proposed ACO_{MV-R}-SVM algorithm and $IACO_{MV-R}$-SVM algorithm achieve the smallest feature subset size for eight datasets, while in only two datasets, Ionosphere and Iris, the size of feature subset produced by the proposed algorithms was comparative with CSA-SVM. Both of the proposed algorithms produced similar feature subset size; this is because both the proposed algorithms count on the ants to select features. The smallest reduction in feature size was 75% for the Iris dataset, while the highest reduction in feature size was 87.22% for Splice and Vehicle datasets.

Table 5.5: Comparison of Feature Subset Size

Datasets	**Original number of features**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Australian	11	1.6 ± 0.49	1.8 ± 0.4	5.2 ± 2.15	6.7	7.2 ± 2.25	9 ± 2.01	10.60 ± 2.46	3 ± 2.45
German	20	3.3 ± 1.19	3.9 ± 0.3	10.3 ± 1.76	12	12.5 ± 3.31	18 ± 3.49	-	13 ± 1.83
Heart	13	1.9 ± 0.3	2 ± 0	6.2 ± 1.12	4.5	8 ± 2.49	-	-	5.4 ± 1.85
Ionosphere	34	5.4 ± 1.5	6 ± 0	13.9 ± 3.45	6	17 ± 6.16	21 ± 3.23	22.36 ± 5.02	6 ± 0
Iris	4	1 ± 0	1 ± 0	1.2 ± 0.28	1	2.6 ± 0.51	2 ± 0.64	-	1 ± 0
Pima-India Diabetes	8	1.6 ± 0.49	2 ± 0	3.7 ± 1.26	3.5	5.2 ± 0.92	5 ± 1.26	5.86 ± 1.17	3.7 ± 0.95
Sonar	60	10.8 ± 2.56	12 ± 0	26.4 ± 3.20	13.7	24.4 ± 8.11	37 ± 4.75	48.18 ± 45.53	15 ± 1.1
Vehicle	18	2.3 ± 0.9	2.8 ± 0.4	9.2 ± 1.71	10.3	13.8 ± 2.66	13 ± 1.85	15.56 ± 2.17	9.2 ± 1.4

1 Proposed ACO$_{MV-R}$-SVM Algorithm
2 Proposed IACO$_{MV-R}$-SVM Algorithm
3 GA$_{\text{with feature chromosome}}$-SVM
4 CSA-SVM
5 CSO-SVM
6 PSO-SVM
7 SA-SVM
8 GA-SVM

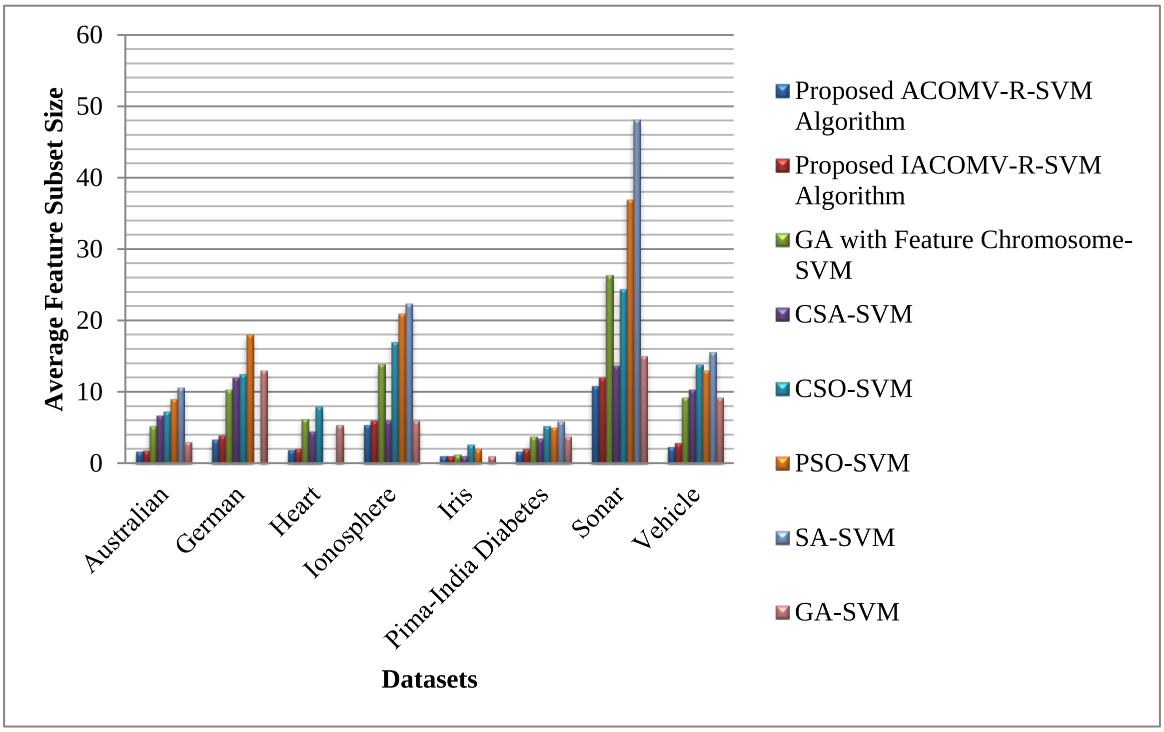

Figure 5.11 Comparison of feature subset size

Table 5.6 and Figure 5.12 compare the size of feature subset selection between the proposed ACO$_{\text{MV-R}}$-SVM and IACO$_{\text{MV-R}}$-SVM algorithms with ACO-SVM and GA$_{\text{with feature chromosome}}$-SVM. The results show that the proposed ACO$_{\text{MV-R}}$-SVM and IACO$_{\text{MV-R}}$-SVM algorithms achieve the smallest feature subset size.

Table 5.6: Average Feature Subset Size

Datasets	**Original number of features**	**1**	**2**	**3**	**4**
Image Segmentation	18	2.3 ± 0.9	3 ± 0	18	13.2
Pima-India Diabetes	8	1.6 ± 0.49	2 ± 0	3.7	5.4
Splice	61	10.5 ± 3.2	9.5 ± 4.18	61	7

1 Proposed ACO$_{\text{MV-R}}$-SVM Algorithm
2 Proposed IACO$_{\text{MV-R}}$-SVM Algorithm
3 GA$_{\text{with Feature Chromosome}}$-SVM
4 ACO-SVM

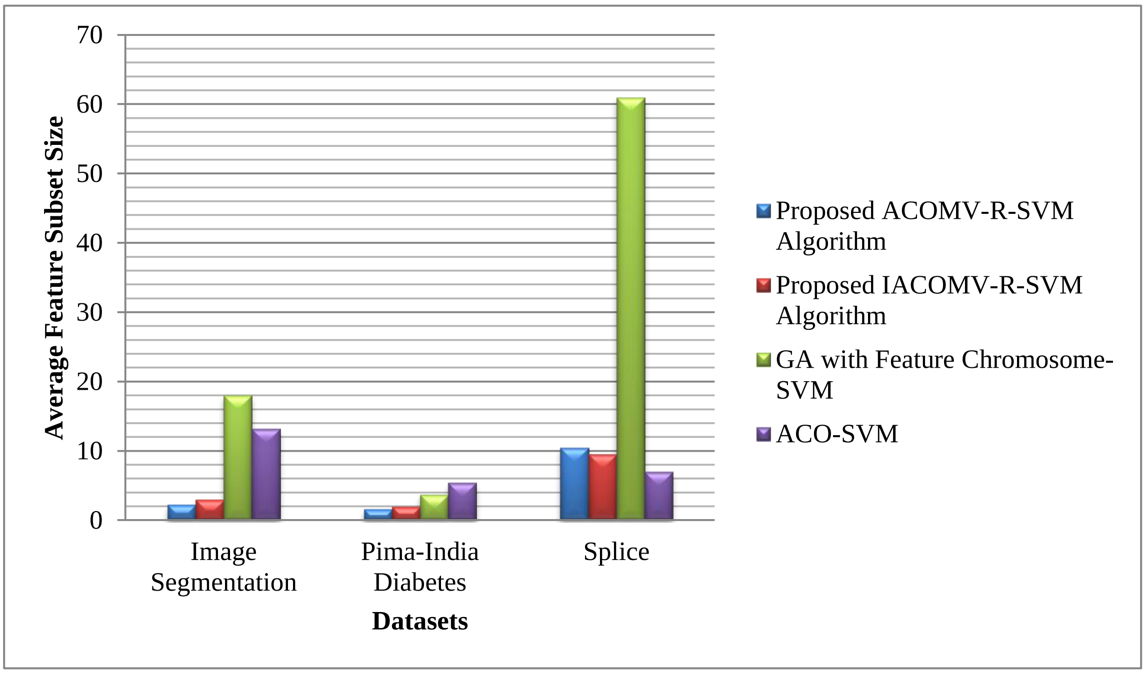

Figure 5.12 Average feature subset size

Table 5.7 and Figure 5.13 compare the size of feature subset selection between the proposed ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM algorithms with GSA-SVM. The results show that the size of feature subset selection of the proposed algorithms was similar to the size of feature subset selection produced by GSA-SVM.

Table 5.7: Comparison of Feature Subset Size between Proposed Algorithm and GSA-SVM

Datasets	**Original number of features**	**1**	**2**	**3**
Australian	11	1.6 ± 0.49	1.8 ± 0.4	2.2 ± 1.3
German	20	3.3 ± 1.19	3.9 ± 0.3	4.6 ± 2.1
Ionosphere	34	5.4 ± 1.5	6 ± 0	5.8 ± 2.49
Pima-India Diabetes	8	1.6 ± 0.49	2 ± 0	2 ± 1

1 Proposed ACO_{MV-R}-SVM Algorithm 2 Proposed $IACO_{MV-R}$-SVM Algorithm
3 GSA-SVM

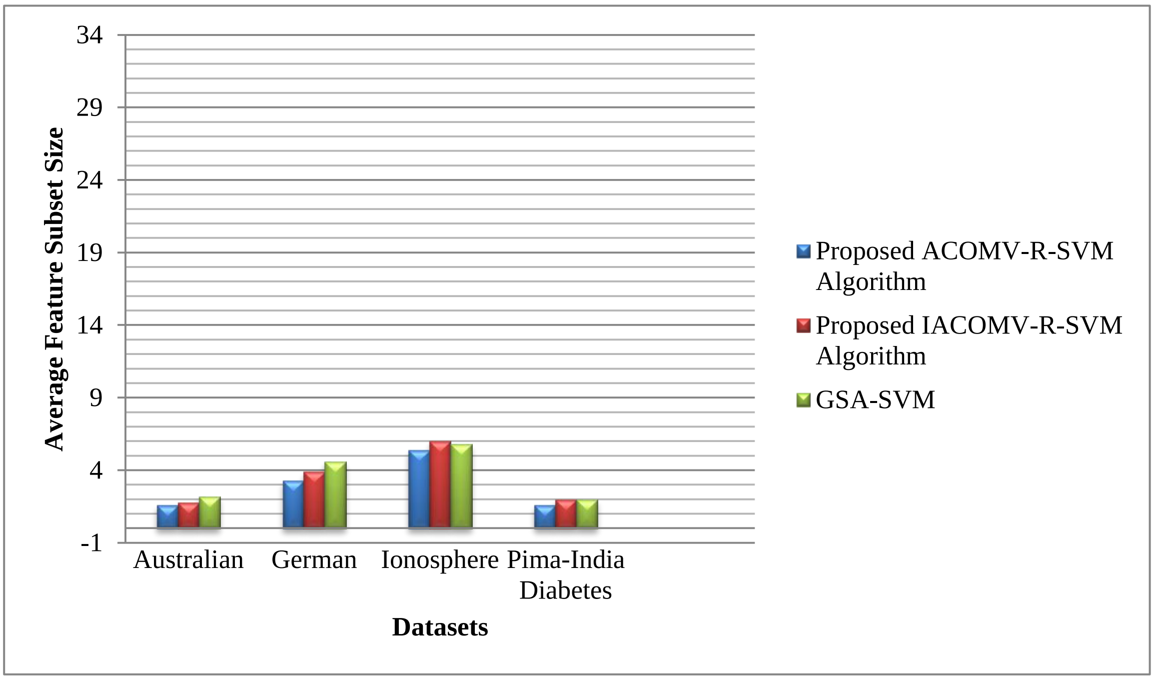

Figure 5.13 Comparison of feature subset size between proposed algorithm and GSA-SVM

Table 5.8 and Figure 5.14 compares the size of feature subset selection of the proposed $ACO_{MV\text{-}R}$-SVM and $IACO_{MV\text{-}R}$-SVM algorithms with the size of feature subset selection produced by the proposed ACO_R-SVM and $IACO_R$-SVM algorithms. The comparison results between the four proposed algorithms show that the proposed $ACO_{MV\text{-}R}$-SVM algorithm and $IACO_{MV\text{-}R}$-SVM algorithm produced the smallest feature subset size compared with the proposed ACO_R-SVM algorithm and $IACO_R$-SVM algorithm. This is because the proposed $ACO_{MV\text{-}R}$-SVM algorithm and $IACO_{MV\text{-}R}$-SVM algorithm depend on ants to select feature subset unlike that used in the proposed ACO_R-SVM algorithm and $IACO_R$-SVM algorithm which filtered the features to select the feature subset. As mentioned in Chapter Four, The smallest reduction in feature size was 48.33% for the Vehicle dataset, while the highest reduction in feature size was 80.82% for the Splice dataset.

Table 5.8: Comparison of Feature Subset Size between Proposed Algorithm

Datasets	**Original no. of features**	**1**	**2**	**3**	**4**
Australian	11	3.2 ± 0.4	3.3 ± 0.46	1.6 ± 0.49	1.8 ± 0.4
German	20	6.2 ± 0.77	6.4 ± 0.63	3.3 ± 1.19	3.9 ± 0.3
Heart	13	6.2 ± 0.6	6.0 ± 0.45	1.9 ± 0.3	2 ± 0
Image Segmentation	18	8.3 ± 0.46	8.3 ± 0.46	2.3 ± 0.9	3 ± 0
Ionosphere	34	11.0 ± 1	10.6 ± 1.74	5.4 ± 1.5	6 ± 0
Iris	4	2 ± 0	2 ± 0	1 ± 0	1 ± 0
Pima-India Diabetes	8	2.6 ± 0.49	2.6 ± 0.49	1.6 ± 0.49	2 ± 0
Splice	61	11.7 ± 1.27	11.7 ± 1	10.5 ± 3.2	9.5 ± 4.18
Sonar	60	21.5 ± 1.12	21.7 ± 1.42	10.8 ± 2.56	12 ± 0
Vehicle	18	8.9 ± 1.04	9.3 ± 1.0	2.3 ± 0.9	2.8 ± 0.4

1 Proposed ACO_R-SVM Algorithm
2 Proposed $IACO_R$-SVM Algorithm
3 Proposed ACO_{MV-R}-SVM Algorithm
4 Proposed $IACO_{MV-R}$-SVM Algorithm

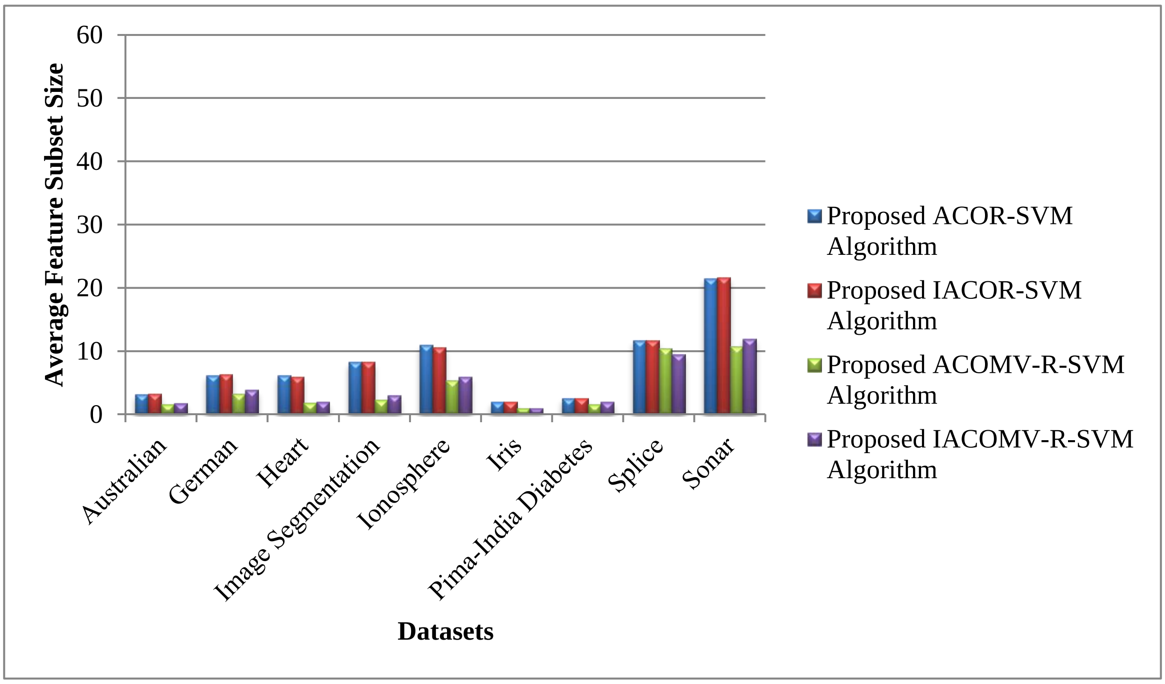

Figure 5.14 Comparison of feature subset size between proposed algorithm

Table 5.9 shows the *p* value of nonparametric Wilconxon signed-rank test of the comparison of the proposed IACO$_{MV-R}$-SVM algorithm with GA $_{\text{with feature chromosome}}$-SVM, CSA-SVM, CSO-SVM, PSO-SVM, SA-SVM, and GA-SVM. As shown in Table 5.9, the *p* value for the Iris dataset of GA $_{\text{with feature chromosome}}$-SVM, CSA-SVM, and GA-SVM is greater than the preselected statistical significance level of 0.05, but other *p* values are lower than the significance level of 0.05. Generally, the proposed IACO$_{MV-R}$-SVM algorithm has higher classification accuracy.

Table 5.9: p-value for Wilcoxon test between Proposed IACO$_{MV\text{-}R}$-SVM^{1}Algorithm, GA$_{with\ feature\ chromosome}$-SVM^2, CSA-SVM^3, CSO-SVM^4, PSO-SVM^5, SA-SVM^6 and GA-SVM^7

Datasets	**1 & 2**	**1 & 3**	**1 & 4**	**1 & 5**	**1 & 6**	**1 & 7**
Australian	0.001	0.001	0.001	0.001	0.001	0.001
German	0.001	0.001	0.001	0.001	-	0.001
Heart	0.0025	0.0025	0.0025	-	-	0.0025
Ionosphere	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
Iris	0.9661	0.9661	0.0025	0.0025	-	0.9661
Pima-India Diabetes	0.0216	0.0216	0.0216	0.0216	0.0216	0.0216
Sonar	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
Vehicle	0.001	0.001	0.001	0.001	0.001	0.001

Table 5.10 shows the *p* value of nonparametric Wilconxon signed-rank test of the comparison of the proposed ACO$_{MV\text{-}R}$-SVM algorithm with GA$_{with\ feature\ chromosome}$-SVM, CSA-SVM, CSO-SVM, PSO-SVM, SA-SVM, and GA-SVM. As shown in Table 5.10, the *p* value for the Iris dataset of GA$_{with\ feature\ chromosome}$-SVM, CSA-SVM, and GA-SVM is greater than the preselected statistical significance level of 0.05, but other *p* values are lower than the significance level of 0.05. Generally, the proposed ACO$_{MV\text{-}R}$-SVM algorithm has higher classification accuracy.

Table 5.10: p-value for Wilcoxon test between Proposed $ACO_{MV\text{-}R}$*-*SVM^1 *Algorithm,* $GA_{with\ feature\ chromosome}$*-*$SVM^2$*, CSA-*$SVM^3$*, CSO-*$SVM^4$*, PSO-*$SVM^5$*, SA-*$SVM^6$ *and GA-*SVM^7

Datasets	**1 & 2**	**1 & 3**	**1 & 4**	**1 & 5**	**1 & 6**	**1 & 7**
Australian	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
German	0.001	0.001	0.001	0.001	-	0.001
Heart	0.0035	0.0025	0.0297	-	-	0.0025
Ionosphere	0.0047	0.0025	0.0047	0.0025	0.0025	0.0025
Iris	0.9784	0.9784	0.0025	0.0025	-	0.9784
Pima-India Diabetes	0.0216	0.0216	0.0216	0.0216	0.0216	0.0216
Sonar	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025
Vehicle	0.0025	0.0142	0.0142	0.0025	0.0142	0.0025

Table 5.11 shows the *p* value of nonparametric Wilconxon signed-rank test of the comparison of the proposed $IACO_{MV\text{-}R}$-SVM algorithm with $GA_{with\ feature\ chromosome}$-SVM and ACO-SVM approaches as well as between the proposed $ACO_{MV\text{-}R}$-SVM algorithm with $GA_{with\ feature\ chromosome}$-SVM and ACO-SVM approaches. As shown in Table 5.11, the *p* values for Image Segmentation dataset of $GA_{with\ feature\ chromosome}$-SVM is greater than the preselected statistical significance level of 0.05 compared with the proposed $ACO_{MV\text{-}R}$-SVM algorithm, but other *p*-values are lower than the significance level of 0.05. Generally, the proposed $IACO_{MV\text{-}R}$-SVM and $ACO_{MV\text{-}R}$-SVM algorithms produced higher classification accuracy.

Table 5.11: p-value for Wilcoxon test between Proposed IACO$_{MV\text{-}R}$-SVM^1 Algorithm, Proposed ACO$_{MV\text{-}R}$-SVM2^{2}Algorithm, GA$_{with\ feature\ chromosome}$-SVM^3 and ACO-SVM^4

Datasets	**1 & 3**	**1 & 4**	**2 & 3**	**2 & 4**
Image Segmentation	0.001	0.001	0.8987	0.0025
Pima-India Diabetes	0.0216	0.0216	0.0216	0.0216
Splice	0.001	0.001	0.001	0.001

Table 5.12 shows the *p* value of nonparametric Wilconxon signed rank test of the comparison between the proposed IACO$_{MV\text{-}R}$-SVM algorithm and GSA approaches as well as between the proposed ACO$_{MV\text{-}R}$-SVM algorithm and GSA approaches. As shown in Table 5.12, the *p* values for the Ionosphere dataset is lower than the preselected statistical significance level of 0.05, but other *p* values are greater than the preselected nonparametric Wilconxon signed-rank test meaning that the proposed IACO$_{MV\text{-}R}$-SVM and ACO$_{MV\text{-}R}$-SVM algorithms produced lower classification accuracy compared with the GSA-SVM approach.

Table 5.12: p-value for Wilcoxon test between Proposed IACO$_{MV\text{-}R}$-SVM^1 Algorithm, Proposed ACO$_{MV\text{-}R}$-SVM^{2}Algorithm and GSA-SVM^3

Datasets	**1 & 3**	**2 & 3**
Australian	0.999	0.9975
German	0.999	0.999
Ionosphere	0.0025	0.0047
Pima-India Diabetes	0.9784	0.9784

Table 5.13 shows the *p*-value of the comparison results between IACO$_{MV\text{-}R}$-SVM and ACO$_{MV\text{-}R}$-SVM. The *p*-value for all datasets was lower than the preselected statistical significance level of 0.05 meaning that the proposed IACO$_{MV\text{-}R}$-SVM

algorithm has higher classification accuracy compared with the proposed ACO_{MV-R}-SVM algorithm.

Table 5.13: p-value for Wilcoxon test between Proposed $IACO_{MV-R}$-SVM^1 Algorithm and Proposed ACO_{MV-R}-SVM^{2}Algorithm

Datasets	**1 & 2**
Australian	0.002
German	0.001
Heart	0.0142
Image Segmentation	0.001
Ionosphere	0.0025
Iris	0.0178
Pima-India Diabetes	0.0216
Sonar	0.0025
Splice	0.002
Vehicle	0.001

Finally, Table 5.14 shows the *p*-value of the comparison results between $IACO_{MV-R}$-SVM and ACO_R-SVM, $IACO_{MV-R}$-SVM and $IACO_R$-SVM, ACO_{MV-R}-SVM and ACO_R-SVM, and ACO_{MV-R}-SVM and $IACO_R$-SVM. The *p*-value for Image Segmentation and Vehicle datasets of ACO_{MV-R}-SVM and ACO_R-SVM, and ACO_{MV-R}-SVM and $IACO_R$-SVM are greater than the preselected statistical significance level of 0.05. Generally, the proposed $IACO_{MV-R}$-SVM and ACO_{MV-R}-SVM algorithms have higher classification accuracy compared with the proposed ACO_R-SVM, and $IACO_R$-SVM algorithms.

Table 5.14: p-value for Wilcoxon test between Proposed $IACO_{MV\text{-}R}$*-SVM*^{1} *Algorithm, Proposed* $ACO_{MV\text{-}R}$*-SVM*^{2} *Algorithm, Proposed* ACO_R*-SVM*^{3} *Algorithm and Proposed* $IACO_R$*-SVM*^{4}*Algorithm*

Datasets	1 & 3	1 & 4	2 & 3	2 & 4
Australian	0.001	0.001	0.0025	0.0035
German	0.001	0.001	0.001	0.001
Heart	0.0025	0.0025	0.0025	0.0035
Image Segmentation	0.001	0.001	0.8987	0.1013
Ionosphere	0.0025	0.0025	0.0025	0.0025
Iris	0.0025	0.0025	0.0109	0.0083
Pima-India Diabetes	0.0216	0.0216	0.0216	0.0216
Sonar	0.001	0.001	0.001	0.001
Splice	0.0025	0.0025	0.0025	0.0025
Vehicle	0.001	0.001	0.4797	0.4797

5.3 Summary

This chapter proposed enhanced pattern classification algorithms based on $ACO_{MV\text{-}R}$ and $IACO_{MV\text{-}R}$ with SVM for pattern classification. $ACO_{MV\text{-}R}$ and $IACO_{MV\text{-}R}$ act as optimization algorithms in order to simultaneously optimize SVM parameters and feature subset selection. This chapter compared the performance of the proposed algorithms to GSA-SVM, ACO-SVM, $GA_{\text{with feature chromosome}}$-SVM, CSA-SVM, CSO-SVM, PSO-SVM, GA-SVM, and SA-SVM concerning the classification accuracy and found the proposed algorithms to be better than other approaches and comparative with GSA-SVM. Also, the experiments' results show that $IACO_{MV\text{-}R}$-SVM is better than $ACO_{MV\text{-}R}$-SVM. Moreover, the experiments in this chapter found that the size of feature selected by the proposed algorithms were smaller than the compared approaches in all datasets while the size of feature subset selected by ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM were almost similar, this is because both the proposed algorithms count on the ants to select feature subset. However, ACO_{MV-R} and $IACO_{MV-R}$ are able to aid the SVM in enhancing its performance due to increased classification accuracy.

CHAPTER SIX
CONCLUSION AND FUTURE WORK

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 while mixed-variable Ant Colony Optimization (ACO_{MV-R}) and Incremental mixed-variable Ant Colony Optimization ($IACO_{MV-R}$) as extensions of the ACO algorithm offer the opportunity to deal with mixed-variable (discrete and continuous) optimization problems (Socha, 2008; Socha & Dorigo, 2008; Socha & Blum, 2005; and Liao et al., 2011). Support Vector Machine (SVM) is considered as one of the most successful classifiers, but it has two problems. The problems comprise how to tune parameters and select the best feature subset for input. The idea to hybridize ACO and SVM in solving SVM problems was first introduced by Huang (2009) where ACO was used for feature selection and tuning SVM parameters.

Four research questions have been considered in this study and four research objectives have been stated to answer these questions. Four generic algorithms have been proposed and comparisons have been made with other well known approaches to test the credibility of the proposed algorithms. Results obtained show significant contributions towards tuning SVM parameters and selecting feature subset.

This chapter presents the research contributions and future work related to pattern classification. Section 6.1 presents the research contributions while Section 6.2 provides the possible extensions of this research.

6.1 Research Contribution

The main goal of this research is to design an enhanced ACO based SVM algorithm for pattern classification. In this study, the four proposed algorithms focus on tuning SVM parameters and selecting feature subset.

The first two proposed ACO_R-SVM and $IACO_R$-SVM algorithms have been used to tune SVM parameters. The continuous variants of ACO, which are ACO_R and $IACO_R$, have been used as an optimization algorithm to tune SVM parameters to improve the performance of SVM without the need to discretize the continuous value of SVM parameters. Selecting 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 algorithms are: (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) solving stagnation, a diversification technique was utilized in $IACO_R$-SVM. 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.

Both algorithms produced good classification accuracy because they can handle continuous values of SVM parameters. However, $IACO_R$ produced better results than ACO_R.

The next two proposed ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM algorithms have been used to simultaneously tune SVM parameters and select feature subset. The mixed-variable variants of ACO which are ACO_{MV-R} and $IACO_{MV-R}$ have been used as an optimization algorithm to simultaneously tune SVM parameters and select feature subset without the need to discretize the continuous value of SVM parameters. The benefit in solving these two problems simultaneously is accumulation of error from feature selection phase to tuning SVM parameters phase can be eliminated. In principle, the differences between the two proposed algorithms are the same as the differences between the proposed ACO_R-SVM and $IACO_R$-SVM algorithms.

The proposed ACO_{MV-R}-SVM and $IACO_{MV-R}$-SVM algorithms select feature subset through a wrapper approach and not through a filter approach as has been used with the first two proposed ACO_R-SVM and $IACO_R$-SVM algorithms. This is done because it gives the ability to integrate the classification accuracy and necessary features introduced to the classifier from an ACO's variants. Furthermore, feature subset selection via the wrapper approach follows the inductive learning approach where, feedback is obtained from the classifier and the process of features selection is repeated to obtain better features.

The contributions mentioned above are employed towards enhancing SVM classification accuracy which is the main objective of this research. Experiments in this thesis show that continuous variant ACO and mixed-variable ACO variants were able to reduce misclassification problems of the SVM classifier. In other words, continuous and mixed-variable ACO variants work successfully to optimize the SVM parameters as well as simultaneously optimizing SVM parameters and selecting suitable feature subset.

To the knowledge of the researcher, there is no study that considers using continuous and mixed-variable ACO based SVM for pattern classification. This study has also covered all the important constraints which make the SVM classifier more accurate compared to previous studies. Comprehensive testing was also performed using ten benchmark datasets.

6.2 Future Work

The first future direction is to apply the proposed algorithms on Support Vector Regression (SVR). According to Huang (2009) and Huang & Wang (2006), SVR has the same problems as SVM. This task requires minimal changes to the generic 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). SVR is extended from Support Vector Classification (SVC). Both SVR and SVC are often used with the kernel trick. The kernel trick maps data to a higher dimensional space and employs a kernel function.

Another future direction is to use the proposed algorithms in tackling 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 be used to tackle 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 (RBF) 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.

Experimental results taken from the University of California, Irvine (UCI) repository, other benchmark datasets and real world problems can be examined in the future to prove and enhance the proposed algorithms.

REFERENCES

Abd-Alsabour, N. & Randall, M. (2010). Feature selection for classification using an ant colony system. Proceedings of The 6th International Conference on e-Science Workshops, 7-10 December 2010, Brisbane, QLD, 86-91.

Abd-Alsabour, N. (2010). Feature selection for classification using an ant system approach. In M. Hinchey, B. Kleinjohann, L. Kleinjohann, P. Lindsay, F. Ramming, J. Timmis & M. Wolf (Eds.), *Distributed, Parallel and Biologically Inspired Systems*, (pp. 233-241). Berlin Heidelberg: Springer.

Abdi, M. & Giveki, D. (2013). Automatic detection of erythemato-squamous diseases using PSO–SVM based on association rules. *Engineering Applications of Artificial Intelligence, 26*(1), 603-608.

Acharya, J., Das, A., Orlitsky, A., Pan, S. & Santhanam, N. (2010). Classification using pattern probability estimators. Proceeding of The International Symposium on Information theory, 13-18 June 2010, Austin, TX, 1493 - 1497.

Acir, N., Ozdamar, O. & Guzelis, C. (2006). Automatic classification of auditory brainstem responses using SVM-based selection algorithm for threshold detection. *Engineering Applications of Artificial Intelligence*, *19*(2), pp. 209-218.

Ahsan, Md., Ibrahimy, M. & Khalifa, O. (2012). EMG motion pattern classification through design and optimization of neural network. Proceedings of The International Conference on Biomedical Engineering, 27-28 February 2012, Penang, Malaysia, 175-179.

Akay, M. (2009). Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications, 36*(2, Part 2), 3240-3247.

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

Al-Janabi, A. (2010). *Interacted multiple ant colonies for search stagnation problem.* (Doctoral dissertation, UUM, 2010). Retrieved from cache:z0fkJlvzCMoJ:www.idsia.ch/~luca/aco2004.pdf maniezzo v. gambardella l. & luigi f. ant colony optimization.pdf.

Al-Naami, B., Al-Nabulsi, J., Amasha, H. & Torry, J. (2010). Utilizing wavelet transform and support vector machine for detection of the paradoxical splitting in the second heart sound. *Medical and Biological Engineering and Computing, 48*(2), 177-184.

Anguita, D. & Ghio, A. (2011). In-sample model selection for support vector machines. Proceeding of The international Joint Conference on Neural Networks, 31 July-5 August 2011, San Jose, CA, 1154 - 1161.

Aribarg, T., Supratid, S. & Lursinsap, C. (2012). Optimizing the modified fuzzy ant-miner for efficient medical diagnosis. *Applied Intelligence, 37*(3), 357-376.

Atienza, F., Alvarez, J., Munoz, A., Vinagre, J., Alberola, A. & Valls, G. (2012). Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection. *Expert System with Applications, 39*(2), pp. 1956-1967.

Azadeh, A., Saberi, M., Kazem, A., Ebrahimipour, V., Nourmohammadzadeh, A. & Saberi, Z. (2013). A flexible algorithm for fault diagnosis in a centrifugal pump with corrupted data and noise based on ANN and support vector machine with hyper-parameters optimization. *Applied Soft Computing, 13*(3), 1478-1485.

Bajla, I., Rublik, F., Arendacka, B., Farkas, I., Hornisova, K. & Stole, S. (2009). Segmentation and supervised classification of image objects in EPO doping-control. M*achine Vision and Applications, 20*(4), 243-259.

Basiri, M., Aghaee, N. & Aghdam, M. (2008). Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins. In E. Marchiori & J. Moore (Eds.), *Evolutionary computation, machine learning and data mining in bioinformatics,* (pp. 12-23). Berlin Heidelberg: Springer.

Bhadra, T., Bandyopadhyay, S. & Maulik, U. (2012). Differential evolution based optimization of SVM parameters for meta classifier design. *Procedia Technology, 4,* 50-57.

Bilchev, G. & Parmee, I. (1995). The ant colony metaphor for searching continuous design spaces. In F. Terence (Ed.), *Evolutionary computing. (pp. 25-39).* Berlin Heidelberg: Springer.

Birattari, M., Pellegrini, P. & Dorigo, M. (2007). On the invariance of ant colony optimization. *Transaction on Evolutionary Computation, 11*(6), 732-742.

Blondin, J. & Saad, A. (2010). Metaheuristic techniques for support vector machine model selection. Proceedings of The 10th International Conference on Hybrid Intelligent Systems, 23-25 August 2010, Atlanta, GA, 197 – 200.

Blum, C. & Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. *Transaction on System, Man, and Cyberntics*, *34*(2), 1161-1172.

Blum, C. & Dorigo, M. (2005). Search bias in ant colony optimization: on the role of competition-balanced systems. *Transactions on Evolutionary Computation*, *9*(2), 159-174.

Blum, C. (2005). Ant colony optimization: introduction and recent trends. *Physics of Life Reviews*, *2*(4), 353-373.

Brogini, A. & Slanzi, D. (2010). Several computational studies about variable selection for probabilistic Bayesian classifiers. In F. Palumbo, C. Lauro & M. Greenacre (Eds.), *Data analysis and classification, studies in classification, data analysis, and knowledge organization*, (pp.1-9). Berlin Heidelberg: Springer.

Bullnheimer, B., Hartl, R. F. & Strauss, C. (1999). A new rank-based version of the ant system: a computational study. *Central European for Operations Research and Economics*, 7(1), 25 – 38.

Castro, L. (2007). Fundamentals of natural computing: an overview. *Physics of Life Reviews*, *4*(1), 1-36.

Chang, C. & Lin, C. (2011). LIBSVM: a library for support vector machines. Retrieved April 16, 2012, from http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.

Chang-Wei, M. & Guang-Yuan, L. (2009). Feature extraction, feature selection and classification from electrocardiography to emotions. Proceeding of the International Conference on Computational Intelligence and Natural Computing, 6-7 June 2009, Wuhan, 190 - 193.

Chapelle, O. & Vapnik, V. (2000). *Model selection for support vector machines*. Retrieved January 1, 2011, from http://olivier.chapelle.cc/pub/nips99.pdf.

Chapelle, O., Vapnik, V., Bousquet, O. & Mukherjee, A. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*. 46(1-3), 131-159.

Chen, B., Chen & Chen, Y. (2013). Efficient ant colony optimization for image feature selection. *Signal Processing 93*(6), 1566-1576.

Chen, H., Liu, G. & Xiong, X. (2011). A novel feature selection method for affective recognition based on pulse signal. Proceeding of The 4th International Symposium on Computational Intelligence and Design, 28-30 October 2011, Hangzhou, 110 - 113.

Chen, L., Sun, H. & Wang, S. (2009). Solving continuous optimization using ant colony algorithm. Proceedings of The 2nd International conference on Future Information Technology and Management Engineering, 13-14 December 2009, Sanya, 92-95.

Chen, Y.-W. & Lin, C.-J. (2006). Combining SVMs with various feature selection strategies. In I. Guyon, M. Nikravesh, S. Gunn & L. Zadeh (Eds.), *Feature Extraction*, (pp. 315-324). Berlin Heidelberg: Springer.

Cheng, X., Xu, J., Pei, J. & Liu, J. (2010). Hierarchical distributed data classification in wireless sensor networks. *Computer Communications*, *33*(12), 1404-1413.

Choi, Y. & Noh, J. (2004). Relevance feedback for content-based image retrieval using proximal support vector machine. In A. Lagana, M. Gavrilova, V. Kumar, Y. Mun, C. J. Tan & O. Gervasi (Eds.), *Computational science and its applications*, (pp. 942-951). Berlin Heidelberg: Springer.

Christen, P. (2009). Development and user experiences of an open source data cleaning, deduplication and record linkage system. *SIGKDD Explorations*, *11*(1), 39-48.

Colak, I., Sagiroglu, S. & Yesilbudak, M. (2012). Data mining and wind power prediction: a literature review. *Renewable Energy*, *46*, 241-247.

Colorni, A., Dorigo, M. & Maniezzo, V. (1991). Distributed optimization by ant colonies. Proceedings of The 1st European Conference on Artificial Life, Paris, 134 – 142.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

David, V. & A. S. (2003). Advanced support vector machines and kernel methods. *Neurocomputing, 55*(1-2), 5-20.

Deriche, M. (20009). Feature selection using ant colony optimization. Proceedings of The 6th International Multi-Conference on Systems, Signals and Devices, 23-26 March 2009, Djerba, 1-4.

Ding, S. & Li, S. (2009). Clonal selection algorithm for feature selection and parameters optimization of support vector machines. Proceedings of The 2nd International Symposium on Knowledge Acquisition and Modeling, 30 November-1 December 2009, Wuhan, 17-20.

Dioşan, L., Rogozan, A. & Pecuchet, J.-P. (2012). Improving classification performance of support vector machine by genetically optimizing kernel shape and hyper-parameters. *Applied Intelligence, 36*(2), 280-294.

Dong, Y., Xia, Z., Tu, M. & Xing, G. (2007). An optimization method for selecting parameters in support vector machines. Proceedings of The 6th International Conference on Machine Learning and Applications, 13-15 December 2007, Cincinnati, OH, 1-6.

Dorigo, M. & Blum, C. (2005). Ant colony optimization theory: a survey. *Theoretical Computer Science, 344*(2-3), 243-278.

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

Dorigo, M. & Stützle, T. (2003). The ant colony optimization metaheuristic: algorithms, applications, and advances. In F. Glover & G. Kochenberger (Eds.), *Handbook of Metaheuristics,* (pp. 250-285). Springer: US.

Dorigo, M. & Stützle, T. (2004). Ant colony optimization. Cambridge, England: MIT Press.

Dorigo, M. & Stützle, T. (2010). Ant colony optimization: overview and recent advances. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*, (pp. 227-263). Springer: US.

Dorigo, M., Birattari, M. & Stützle, T. (2006b). Ant colony optimization artificial ants as a computational intelligence technique. *Computational Intelligence Magazine*, 1(4), 28-39.

Dorigo, M., Oca, M., Oliveira, S. & Stützle, T. (2011). *Ant colony optimization*. Wiley Encyclopedia of Operations Research and Management Science.

Dreo, J. & Siarry, P. (2002). A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions. In M. Dorigo., G. Caro & M. Samples (Eds.), *Ant algorithms* (pp. 216-221). Berlin Heidelberg: Springer.

Dreo, J. & Siarry, P. (2004). Continuous interacting ant colony algorithm based on dense heterarchy. *Future Generation Computer Systems, 20*(5), 841-856.

Dreo, J. & Siarry, P. (2006). An ant colony algorithm aimed at dynamic continuous optimization. *Applied Mathematics and Computation, 181*(1), 457-467.

Dreo, J. & Siarry, P. (2007). Hybrid continuous interacting ant colony aimed at enhanced global optimization. *Algorithmic Operations Research. 2*(1), 52-64.

Dunbar, M., Murray, J., Cysique, L., Brew, B. & Jeyakumar, V. (2010). Simultaneous classification and feature selection via convex quadratic programming with application to HIV-associated neurocognitive disorder assessment. *European Journal of Operational Research, 206*(2), 470-478.

Fan, R.-E., Chen, P.-H. & Lin, C.-J. (2005). Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research, 6,* 1889-1918.

Fang, X. & Bai, T. (2009). Share price prediction using wavelet transform and ant colony algorithm for parameters optimization in SVM. Proceedings of The Global Congress on Intelligent Systems, 19-21 May 2009, Xiamen, 288-292.

Fang, Z. & Li, Q. (2009). An integrated space-time pattern classification approach for individuals' travel trajectories. Proceeding of The 6th International Conference on Fuzzy Systems and Knowledge Discovery, 14-16 August 2009, Tianjin,119-123.

Farvaresh, H. & Sepehri, M. (2011). A data mining framework for detecting subscription fraud in telecommunication. *Engineering Applications of Artificial Intelligence, 24*(1). 182-194.

Feki, A., Ishak, A. & Feki, S. (2012). Feature selection using Bayesian and multiclass support vector machines approaches: application to bank risk prediction. *Expert Systems with Applications, 39*(3), 3087-3099.

Foithong, S., Pinngern, O. & Attachoo, B. (2012). Feature subset selection wrapper based on mutual information and rough sets. *Expert Systems with Applications, 39*(1), 574-584.

Forrester, E. (2006). *A process research framework.* Pittsburgh, USA: Carnegie Mellon University.

Franca, F., Coelho, G., Zuben, F. & Attux R. (2008). Multivariate ant colony optimization in continuous search spaces. Proceedings of The 10th Annual Conference on Genetic and Evolutionary Computation, 12-16 July 2008, Atlanta, Georgia, USA, 9-16.

Frohlich, H., Chapelle, O. & Scholkopf, B. (2003). Feature selection for support vector machines by means of genetic algorithms. Proceedings of The 15th International Conference on Tools with Artificial Intelligence, 3-5 November 2003, 142-148.

Fu, Z., Lu, G., Ting, K. M. & Zhang, D. (2013). Efficient nonlinear classification via low-rank regularised least squares. *Neural Computing & Applications, 22*(7-8), 1279-1289.

Garbarine, E., DePasquale, J., Gadia, V., Polikar, R. & Rosen, G. (2011). Information-theoretic approaches to SVM feature selection for metagenome read classification. *Computational Biology and Chemistry, 35*(3), 199-209.

Garde, A., Voss, A., Caminal, P. & Benito, S. (2013). SVM-based feature selection to optimize sensitivity–specificity balance applied to weaning. *Computers in Biology and Medicine, 43*(5), 533-540.

Glasmachers, T. & Igel, C. (2010). Maximum likelihood model selection for 1-norm soft margin SVMs with multiple parameters. *Transactions on Pattern Analysis and Machine Intelligence, 32*(8), 1522-1528.

Gürbüz, F., Özbakir, L. & Yapici, H. (2011). Data mining and preprocessing application on component reports of an airline company in turkey. *Expert Systems with Applications, 38*(6). 6618-6626.

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1-3), 389-422.

Hall, M. (2000). *Feature selection for discrete and numeric class machine learning*. Retrieved December 20, 2010, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.6025.

Han, H., Hou-Jun, W. & Xiucheng, D. (2011). Transformer fault dignosis based on feature selection and parameter optimization. *Energy Procedia, 12*, 662-668.

Harvey, J. (2009). *GPI acceleration object classification algorithm using NVIDIA CUDA*. (Master Thesis, Rochester Institute of Technology, 2009). Retrieved from http://ritdml.rit.edu/bitstream/handle/1850/10894/35445_pdf_00B0B24A-DFD8-11DE-9A30-D21AD352ABB1.pdf?sequence=1.

Hidalgo-Muňoz, A.R., López, M.M., Santos, I.M., Pereira, A.T., Vázquez-Marrufo, M., Galvao-Carmona, A. & Tomé, A.M. (2013). Application of SVM-RFE on EEG signals for detecting the most relevant scalp regions linked to affective valence processing. *Expert systems with Applications*, *40*(6), 2101-2108.

Hong, L. & Shibo, X. (2008). On ant colony algorithm for solving continuous optimization problem. Proceedings of The International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 15-17 August 2008, Harbin, 1450 - 1453.

Hric, M., Chmulik, M. & Jarina, R. (2011). Model parameters selection for SVM classification using Particle Swarm Optimization. Proceedings of The 21st International Conference of the Radioelektronika, 19-20 April 2011, Brno, 1-4.

Hsu, C.-W. & Lin, C.-J. (2002a). A simple decomposition method for support vector machines. *Machine Learning, 46,* 291-314.

Hsu, C.-W. & Lin, C.-J. (2002b). A comparison of methods for multiclass support vector machines. *Neural Networks, 13*(2), 415-425.

Hu, Q., Pan, W., Song, Y. & Yu, D. (2012). Large-margin feature selection for monotonic classification. *Knowledge-Based Systems, 31*, 8-18.

Hu, X., Zhang, J. & Li, Y. (2008). Orthogonal methods based ant colony search for solving continuous optimization problems. *Computer Science and Technology, 23*(1), 2-18.

Hu, X., Zhang, J., Chung, H., Li, Y. & Liu, O. (2010). SamACO:variable sampling ant colony optimization algorithm for continuous optimization. *Transactions on System, Man, and Cybernetics, 40*(6), 1555-1566.

Huang, C. & Dun, J. (2008). A Distributed PSO-SVM hybrid system with feature selection and parameter optimization. *Applied Soft Computing, 8*(4), 1381-1391.

Huang, C. & Wang, C. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, 31(2), 231-240.

Huang, C. (2009). ACO-based hybrid classification system with feature subset selection and model parameters optimization. *Neurocomputing*, 73(1-3), 438-448.

Huang, H. & Hao, Z. (2006). ACO for continuous optimization based on discrete encoding. In M. Dorigo, L. Gambardella, M. Birattari, A. Martinoli, R. Poli & T. Stützle (Eds.), *Ant colony optimization and swarm intelligence* (pp. 504-505). Berlin Heidelberg: Springer.

Huang, H., Xie, H.-B., Guo, J.-Y. & Chen, H.-J. (2012). Ant colony optimization-based feature selection method for surface electromyography signals classification. *Computers in Biology and Medicine*, *42*(1), 30-38.

Ibrahim, R., Sethu, V. & Ambikairajah, E. (2010). Novel delta zero crossing regression features for gait pattern classification. Proceeding of The 32nd Annual International Conference of The on Engineering in Medicine and Biology Society, 31 August-4 September 2010, Buenos Aires, 2427 – 2430.

Imbault, F. & Lebart, K. (2004). A stochastic optimization approach for parameter tuning of support vector machines. Proceedings of The 17th International Conference on Pattern Recognition, 23-26 August 2004, 597-600.

Jensen, R. & Shen, Q. (2004). Fuzzy-rough data reduction with ant colony optimization. *Fuzzy Sets and Systems*, *149* (1), 5–20.

Jensen, R. (2006). Performing feature selection with ACO. In A. Abraham, C. Grosan & V. Ramos (Eds.), *Swarm intelligence in data mining*, (pp 45-73). Berlin Heidelberg: Springer.

Jin, Y., Hu, Y., Huang, J. & Zhang, J. (2011). Study on influences of model parameters on the performance of SVM. Proceedings of The International Conference on Electrical and Control Engineering, 16-18 September 2011, Yichang, 3667 - 3670.

Jin, Z., Chen, Y. & Ma, X. (2010). Model selection for SVM based on similarity margin of inner product. Proceedings of The Conference on Control and Decision Conference, 26-28 May 2010, Xuzhou, 1653 - 1657.

Ju, W., Shan, J., Yan, C. & Cheng, H. (2009). Discrimination of disease-related non-synonymous single nucleotide polymorphisms using multi-scale RBF kernel fuzzy support vector machine. *Pattern Recognition Letters*, *30*(4), 391-396.

Kabir, M., Shahjahan, M. & Murase, K. (2009). An efficient feature selection using ant colony optimization algorithm. In C. Leung & J. Chan (Eds.), *Neural information processing*, (pp. 242-252). Berlin Heidelberg: Springer.

Kabir, M., Shahjahan, M. & Murase, K. (2012). A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications, 39*(3), pp. 3747-3763.

Kadoury, S. & Levine, M. (2006). Finding faces in gray scale images using locally linear embeddings. In H. Sundaram, M. Naphade, J. Smith & Y. Rui

(Eds.), *Image and video retrieval*, (pp. 221-230). Berlin Heidelberg: Springer.

Kanan, H. & Faez, L. (2008). An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. *Applied Mathematics and Computation, 205*(2), 716-725.

Kanan, H., Faez, K. & Taheri, S. (2007). Feature selection using ant colony optimization (ACO): A new method and comparative study in the application of face recognition system. In P. Perner (Ed), *Advances in Data Mining. Theoretical Aspects and Applications,* (pp. 63-76). Berlin Heidelberg: Springer.

Kapp, M. N., Sabourin, R. & Maupin, P. (2012). A dynamic model selection strategy for support vector machine classifiers. *Applied Soft Computing, 12*(8), 2550–2565.

Kapp, M., Sabourin, R. & Maupin, P. (2009). A PSO-based framework for dynamic SVM model selection. Proceedings of The 11th Annual Conference on Genetic and Evolutionary Computation, 8-12 July 2009, Montréal Québec, Canada, 1227-1234.

Khalid, S. & Razzaq, S. (2012). Frameworks for multivariate *m*-mediods based modelling and classification in Euclidean and general feature spaces. *Pattern Recognition, 45*(3), 1092-1103.

Khashei, M., Hamadani, A. Z. & Bijari, M. (2011). A fuzzy intelligent approach to the classification problem in gene expression data analysis. *Knowledge-Based Systems, 27*, 465–474.

Khashei, M., Zeinal Hamadani, A. & Bijari, M. (2012). A novel hybrid classification model of artificial neural networks and multiple linear regression models. *Expert Systems with Applications, 39*(3), 2606-2620.

Khushaba, R., AlSukker, A., Al-Ani, A. & Al-Jumaily, A. (2008a). A combined ant colony and differential evolution feature selection algorithm. In M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle & A. Winfield (Eds.), *Ant Colony Optimization and Swarm Intelligence*, (pp. 1-12). Berlin Heidelberg: Springer.

Khushaba, R., AlSukker, A., Al-Ani, A. & Al-Jumaily, A. (2008b). Intelligent artificial ants based feature extraction from wavelet packet coefficients for biomedical signal classification. Proceeding of The 3rd International Symposium on Communications, Control and Signal Processing, 12-14 March 2008, St Julians, 1366 – 1371.

Kong, M. & Tian, P. (2005). A binary ant colony optimization for the unconstrained function optimization problem. In Y. Hao, J. Liu, Y. Wang, Y.-M. Cheung, H. Yin, L. Jiao, J. Ma & Y.-C. Jiao (Eds.), Computational intelligence and security, (pp. 632-687). Berlin Heidelberg: Springer.

Kong, M. & Tian, P. (2006a). Application of ACO in continuous domain. In L. Jiao, L. Wang, X. Gao, J. Liu & F. Wu (Eds.), *Advances in Natural Computation*, (pp. 126-135). Berlin Heidelberg: Springer.

Kong, M. & Tian, P. (2006b). A direct application of ant colony optimization to function optimization problem in continuous domain. In M. Dorigo, L. Gambardella, M. Birattari, A. Martinoli, R. Poli & T. Stützle (Eds.), ant colony optimization and swarm intelligence, (pp. 324-331). Berlin Heidelberg: Springer.

Kovářík, O. (2006). *Ant colony optimization for continuous problems*. (Master Thesis, Czech Technical University in Prague, 2006). Retrieved from http://fakegame.sourceforge.net/lib/exe/fetch.php?media=ants-optimalizace.pdf.

Kumar, R. (2011). Research methodology: a step-by-step guide for beginners (3rd Ed.). London: Sage Publications Ltd.

Lázaro-Gredilla, M., Gómez-Verdejo, V. & Parrado-Hernández. Low-cost model selection for SVMs using local features. (2012). *Engineering Application of Artificial Intelligence, 25*(6), 1203-1211.

Lei, L. & Qiao, G. (2012). Text categorization using SVM with exponent weighted ACO. Proceedings of The 31st Chinese Control Conference, 25-27 July 2012, Hefei, 3763 - 3768.

Lessmann, S., Stahlbock, R. & Crone, S. (2006). Genetically constructed kernels for support vector machines. In Hans H., Herbert K. & Jorn S. (Eds.), *Operations research*, (pp. 257-262). Berlin Heidelberg: Springer.

Li, N. & Zhou, R. (2011). A new approach to fault pattern classification of gasoline engine vibration. Proceedings of The 8th International Conference on Fuzzy systems and Knowledge Discovery, 26-28 July 2011, Shanghai, 1892 - 1896.

Li, S. & Tan, M. (2010). Tuning SVM parameters by using a hybrid CLPSO-BFGS algorithm. *Neuro Computing*, *73*(10-12), 2089-2096.

Li, W., Liu, L. & Gong, W. (2011). Multi-objective uniform design as a SVM model selection tool for face recognition. *Expert Systems with Applications, 38*(6), 6689-6695.

Li, X. & Chen, Z. (2010). Weed identification based on shape features and ant colony optimization algorithm. Proceeding of the International Conference on Computer Application and system Modeling, 22-24 October 2010, Taiyuan, 384-387.

Li, Y., Wang, G., Chen, H., Shi, L. & Qin, L. (2013). An ant colony optimization based dimension reduction method for high-dimensional datasets. *Journal of Bionic Engineering, 10*(2), 231-241.

Lian, H. (2012). On feature selection with principle component analysis for one-class SVM. *Pattern Recognition Letters, 33*(9), 1027-1031.

Liao, S., Yang, C. & Ding, L. (2011). Approximate parameter tuning of support vector machines. Proceedings of The 3rd International Workshop on Intelligent System and Applications, 28-29 May 2011, Wuhan, 1-4.

Liao, T. (2011). *Improved ant colony optimization algorithms for continuous and mixed discrete-continuous optimization problems* (Report No. 2010/2011). Retrieved from IRIDIA, Universite Libre de Bruxelles: Brussels, Belgium website: http://iridia.ulb.ac.be/~tliao/papers/DEAreport-Liao.pdf.

Liao, T., Oca, M., Aydin, D., Stützle, T. & Dorigo, M. (2011). *An incremental ant colony algorithm with local search for continuous optimization* (Report No. TR/IRIDIA/2011-005). Retrieved from IRIDIA, Universite Libre de

Bruxelles: Brussels, Belgium website: http://iridia.ulb.ac.be/IridiaTrSeries/rev/IridiaTr2011-005r002.pdf.

Lin, K. & Chien, H. (2009). CSO-based feature selection and parameter optimization for support vector machine. Proceedings of The Joint Conference on Pervasive Computing, 3-5 December 2009, Tamsui, Taipei, 783-788.

Lin, S., Lee, Z.-J., Chen, S.-C. & Tseng, T.-Y. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing, 8*(4), 1505-1512.

Lin, S., Ying, K., Chen, S. & Lee, Z. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications,* 35(4), 1817-1824.

Lin, X., Yang, F., Zhou, L., Yin, P., Kong, H. Xing, W., Li, X., Jia, L., Wang, Q. & Xu, G. (2012). A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information. *Journal of Chromatography B*, *910*, 149-155.

Liu, A., Cao, H., Chen, X., He, Z. & Shen, Z. (2013). Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings. *Neurocomputing, 99*, 399-410.

Liu, G., Zhou, D., Xu, H. & Mei, C. (2010). Model optimization of SVM for a fermentation soft sensor. *Expert Systems with Applications, 37*(4), 2708-2713.

Liu, H., Liu, L. & Zhang, H. (2010). Ensemble gene selection by grouping for microarray data classification. *Journal of Biomedical Informatics, 43*(1), 81-87.

Liu, J. & Yuan , X. (2009). Obscure bleeding detection in endoscopy images using support vector machines. *Optimization Engineering*, *10*(2), 289-299.

Liu, Q., Chen, C., Zhang, Y. & Hu, Z. (2011). Feature selection for support vector machines with RBF kernel. *Artificial Intelligence Review, 36*(2), 99-115.

Liu, W. & Zhang, D. (2009). Feature subset selection based on improved discrete particle swarm and support vector machine algorithm. Proceedings of The Information Engineering and Computer Science, 19-20 December 2009, Wuhan, 1-4.

Liu, Y. & Zheng, F. (2006). FS-SFS: A novel feature selection method for support vector machines. *Pattern Recognition*, *39*(7), 1333-1345.

Lu, N., Zhou, J., He, Y. & Liu, Y. (2009). Particle swarm optimization for parameter optimization of support vector machine model. Proceedings of The 2nd International Conference on Intelligent Computation Technology and Automation, 10-11 October 2009, Changsha, Hunan, 283-286.

Luo, Z.-Y., Zhang, W.-F., Ye, B.-Y. & Cai, L.-Q. (2008). Wavelet SVM ensemble for pattern classification with quantum-inspired evolutionary algorithm. Proceedings of The International Conference on Wavelet Analysis and Pattern Recognition, 30-31 August 2008, Hong Kong, 485 - 490.

Madadgar, S. & Afshar, A. (2011). Forced water main design mixed ant colony optimization. *International Journal of Optimization in Civil Engineering, 1,* 47-71.

Maldonado, S. & Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences, 179*(13), 2208-2217.

Maldonado, S., Weber, R. & Basak, J. (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Science, 181*(1), 115-128.

Marcus, K. & Akila, M. (2010). Personal authentication based on keystroke dynamic using soft computing techniques. Proceedings of The 2nd International Conference on Communication Software and Network, 26-28 February 2010, Singapore, 334 - 338.

Mastrogiannis, N., Boutsinas, B. & Giannikos, I. (2009). A method for improving the accuracy of data mining classification algorithms. *Computers & Operations Research, 36*(10), 2829-2839.

Mesleh, A. & Kanaan, G. (2008). Support vector machines text classification system: using ant colony optimization based feature subset selection. Proceedings of The International Conference on Computer Engineering & Systems, 25-27 November 2008, Cairo, Egypt, 143-148.

Monmarch, M., Venturini, G. & Slimane, M. (2000). On how pachycondyla apicalis ants suggest a new search algorithm. *Future Generation Computer Systems,* 16(8), 937-946.

Moustakidis, S. & Theocharis, J. (2010). SVM-FuzCoC: A novel SVM-based feature selection method using a fuzzy complementary criterion. *Pattern Recognition, 43*(11), 3712-3729.

Nemati, S., Basiri, M., Ghasem-Aghaee, N. & Aghdam, M. (2009). A novel ACO-GA hybrid algorithm for feature selection in protein function prediction. *Expert System with Applications, 36*(10), 12086-12094.

Nguyen, H., Ohn, S., Chae, S., Song, D. & Lee, I. (2006). Optimizing weighted kernel function for support vector machine by genetic algorithm. In A. Gelbukh & C. Garcia (Eds.), *Advances in artificial intelligence*, (pp. 583-592). Berlin Heidelberg: Springer.

Nguyen, M. & Torre, F. (2010). Optimal feature selection for support vector machines. *Pattern Recognition, 43*(3), 584-591.

Oliveira, L., Sabourin, R., Bortolozzi, F. & Suen, C. (2003). A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), 17*(6), 1-35.

Örkcü, H. & Bal, H. (2011). Comparing performance of back propagation and genetic algorithms in the data classification. *Expert Systems with Applications,38*(4), 3703-3709.

Otero, F., Freitas, A. & Johnson, C. (2012). Inducing decision trees with an ant colony optimization algorithm. *Applied Soft Computing, 12*(11), 3615-3626.

Partridge, D., Wang, W. & Jones, P. (2001). Efficient and effective feature selection in the presence of feature interaction and noise. In S. Singh, N. Murshed & W. Kropatsch (Eds.), *Advances in pattern recognition* (pp. 196–204), Berlin Heidelberg: Springer.

Pellegrini, P. Favaretto, D. & Moretti, E. (2007). Multiple ant colony optimization for a rich vehicle routing problem: a case study. In B. Apolloni, R. Howlett & L. Jain (Eds.). *Knowledge-based intelligent information and engineering systems* (pp. 627-634), Berlin Heidelberg: Springer.

Peng, Y., Wu, Z. & Jiang, Z. (2010). A novel feature selection approach for biomedical data classification, *Journal of Biomedical Informatics*, *43*(1), pp. 15-23.

Pham, D., Muhamad, Z., Mahmuddin, M., Ghanbarzadeh, A., Koc, E. & Otri, S. (2007). *Using the bees algorithm to optimise a support vector machine for wood defect classification*. *Conference on Innovative Production Machines and Systems Virtual*.

Phokharatkul, P. & Phaiboon, S. (2011). Control chart pattern classification using Fourier descriptors and neural networks. Proceedings of The 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce, 8-10 August 2011, Deng Leng, 4587 - 4590.

Polat, K. & Güneş, S. (2009). A new feature selection method on classification of medical datasets: kernel F-score feature selection. *Expert Systems with Applications, 36*(7), 10367-10373.

Polat, K. (2013). Data weighting method on the basis of binary encoded output to solve multi-class pattern classification problems. *Expert Systems with Applications, 40*(11), 4637-4647.

Pourtakdoust, S. & Nobahari, H. (2004). An extension of ant colony system to continuous optimization problems. In M. Dorigo, M. Birattari, C. Blum, L. Gambardella, F. Mondada, T. Stützle (Eds.), *Ant colony optimization and swarm intelligence,* (pp. 294-301), Berlin Heidelberg: Springer.

Qi, Z., Tian, Y. & Shi, Y. (2013). Robust twin support vector machine for pattern classification. *Pattern Recognition, 46*(1), 305–316.

Qian, Q., Chen, S. & Cai, W. (2011). Simultaneous clustering and classification over cluster structure representation. *Pattern Recognition, 45*(6), 2227–2236.

Qiu, D., Li, Y., Zhang, X. & Gu, B. (2011). Support vector machine with parameter optimization by bare bones differential evolution. Proceedings of The 7th International Conference of the Natural Computation, 26-28 July 2011, Shanghai, 263-266.

Rasmy, M., El-Beltagy, M., Saleh, M. & Mostafa, B. (2012). A hybridized approach for feature selection using ant colony optimization and ant-miner for classification. Proceedings of The 8th International Conference on Informatics and Systems, 14-16 May 2012, Cairo, Egypt, BIO-211 - BIO-219.

Reed, S., Reed, T. & Dascalu, S. (2010). High dimensional pattern recognition using the recursive hyperspheric classification algorithm. Proceedings of The World Automatic Congress, 19-23 September 2010, Kobe, 1-8.

Reif, M., Shafait, F. & Dengel, A. (2012). Meta-learning for evolutionary parameter optimization of classifiers. *Machine Learning, 87*(3), 357-380.

Ren, J., Shen, Y., Ma, S. & Guo, L. (2004). Applying multi-class SVMs into scene image classification. In B. Orchard, C. Yang & M. Ali (Eds.), *Innovations in applied artificial intelligence*, (pp. 924-934). Berlin Heidelberg: Springer.

Saini, L., Aggarwal, S. & Kumar, A. (2010). Parameter optimization using genetic algorithm for support vector machine-based price-forecasting model in national electricity market. *IET Generation, Transmission & Distribution, 4*(1), 36-49.

Salama., M. Hassanien, A. & Fahmy, A. (2010). Pattern-based subspace classification model. Proceedings of The 2nd World Congress on Natural and Biologically Inspired Computing, 15-17 December 2010, Fukuoka, 357 - 362.

Salama., M. Hassanien, A. & Fahmy, A. (2010). Uni-class pattern-based classification model. Proceedings of The 10th International Conference on Intelligent Systems Design and Applications, 29 November-1 December 2010, Cairo, Egypt, 1293 - 1297.

Saltan, M., Terzi, S. & Küçüksille, E. (2011). Backcalculation of pavement layer moduli and poission’s ratio using data mining. *Expert Systems with Applications, 38*(3), 2600-2608.

Samadzadegan, F., Soleymani, A. & Abbaspour, A. (2010). Evaluation of genetic algorithms for tuning SVM parameters in multi-class problems.

Proceedings of The 11th International Symposium on Computational Intelligence and Informatics, 18-20 November 2010, Budapest, 323-328.

Santana, L., Canuto, A. & Silva, L. (2011). Bio-inspired meta-heuristic as feature selector in ensemble systems: a comparative analysis. Proceedings of The international Joint Conference on Neural Networks, 31 July-5 August 2011, San Jose, CA, 1112 - 1119.

Santana, L., Silva, L., Canuto, A., Pintro, F. & Vale, K. (2010). A comparative analysis of genetic algorithm and ant colony optimization to select attributes for an heterogeneous ensemble of classifiers. Proceedings of The Congress on Evolutionary Computation, 18-23 July 2010, Barcelona, 1-8.

Sanz, S., Cumplido, M., Cruz, F. & Calzon, C. (2002). Feature selection via genetic optimization. In J. Dorronsoro (Ed.), *Artificial Neural Networks*, (pp. 547–552). Berlin Heidelberg: Springer.

Sarafrazi, S. & Nezamabadi-pour, H. (2013). Facing the classification of binary problems with a GSA-SVM hybrid system. *Mathematical and Computer Modelling, 57*(1-2), 270-278.

Sartakhti, J. S., Zangooei, M. H. & Mozafari, K. (2011). Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA). *Computer Methods and Programs in Biomedicine, 108*(2), *570-579*.

Schaffernicht, E., Stephan, V. & Groß, H. (2007). An efficient search strategy for feature selection using chow-liu trees. In J. de Sá, L. Alexandre, W. Duch,

D. Mandic (Eds.). *Artificial neural networks*. (pp. 190-199). Berlin Heidelberg: Springer.

Seçkiner, S., Eroğlu, Y., Emrullah, M. & Dereli, T. (2013). Ant colony optimization for continuous functions by using novel pheromone updating. *Applied Mathematics and Computation, 219*(9). 4163-4175.

Sim, K. & Sun, W. (2003). Multiple ant colony optimization for load balancing. In J. Liu, Y.-M. Cheung & H. Yin (Eds.). *Intelligent data engineering and automated learning,* (pp. 467-471). Berlin Heidelberg: Springer.

Shahjahan, M. & Murase, K. (2009). An efficient feature selection using ant colony optimization algorithm. In C. Leung & J. Chan (Eds.), *Neural information processing*, (pp. 242-252). Berlin Heidelberg: Springer.

Shieh, M. & Yang, C. (2008). Multiclass SVM-RFE for product from feature selection. *Expert Systems with Applications*, *35*(1-2), 531-541.

Sivagaminathan, R. & Ramakrishnan, S. (2007). A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert Systems with Applications, 33*(1), 49-60.

Socha, K. & Blum, C. (2006). Ant colony optimization. In E. Alba & R. Martí (Eds.), *Metaheuristic procedures for training neutral networks*, (pp. 153-180). Springer: US.

Socha, K. & Dorigo, M. (2008). Ant colony optimization for continuous domain. *European Journal of Operational Research, 185*(3), 1155-1173.

Socha, K. (2004). ACO for continuous and mixed-variable optimization. In M. Dorigo, M. Birattari, C. Blum, L. Gambardella & F. Mondada (Eds.), *Ant*

Colony Optimization and Swarm Intelligence, (pp. 25-36). Berlin Heidelberg: Springer.

Socha, K. (2008). *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.*

Stützle, T. & Hoos, H. (1997). MAX-MIN ant system and local search for the traveling salesman problem. Proceedings of the International Conference on Evolutionary Computation. 13-16 April, Indianapolis, 309 – 314.

Su, C. & Yang, C. (2008). Feature selection for the SVM: An application to hypertension diagnosis. *Expert Systems with Applications, 34*(1), 754-763.

Sun, J., Zheng, C., Li, X. & Zhou, Y. (2010). Analysis of the distance between two classes for tuning SVM hyperparameters. *Transaction on Neural Networks, 21*(2), 305-318.

Tan, J., Zhang, Z., Zhen, L., Zhang, C. & Deng, N. (2012). Adaptive feature selection via a new version of support vector machine. *Neural Computing & Applications*.

Tan, P.-N., Steinbach, M. & Kumar, V. (2006). Introduction to data mining. Addison Wesley.

Tang, Y., Guo, W. & Gao, J. (2009). Efficient model selection for support vector machine with Gaussian kernel function. Proceedings of The Symposium on Computational Intelligence and Data Mining, 30 March-2 April 2009, Nashville, TN, 40-45.

Theresa, M. & Raj, V. (2013). Fuzzy based genetic neural networks for the classification of murder cases using trapezoidal and lagrange interpolation membership functions. *Applied Soft Computing, 13*(1), 743-754.

Thomas, A. & Oommen, B. (2013). The fundamental theory of optimal "anti-Bayesian" parametric patter classification using order statistics criteria. *Pattern Recognition, 46*(1), 376-388.

Tsai, C. W., Liao, M. Y., Yang, C. S. & Chiang, M. C. (2011). Classification algorithms for interactive multimedia services: a review. *Multimedia Tools and Applications*, 1-29.

Tsai, C.-F. (2010). An evaluation methodology for binary pattern classification systems. Proceedings of The International Conference on Industrial Engineering and Engineering Management, 7-10 December 2010, Macao, 953 - 956.

Tsai, C.-F., Eberle, W. & Chu, C.-Y. (2013). Genetic algorithms in feature and instance selection. *Knowledge-Based systems, 39*, 240-247.

Tseng, V. S. & Lee, C. H. (2009). Effective temporal data classification by integrating sequential pattern mining and probabilistic induction. *Expert Systems with Applications, 36*(5), 9524-9532.

Twomey, C., Stutzle, T., Dorigo, M., Manfrin, M. & Birattari, M. (2010). An analysis of communication policies for homogeneous multi-colony ACO algorithms. *Information Sciences*, 180(12), 2390-2404.

Tzotsos, A. & Argialas, D. (2008). Support vector machine classification for object-based image analysis. In T. Huang, V. Kecman & I. Kopriva (Eds.), *Kernal based algorithms for mining huge data sets supervised, semi-supervised, and unsupervised learning*, (pp. 663-677). Berlin Heidelberg: Springer.

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.

Üney, F. & Türkay, M. (2006). A mixed-integer programming approach to multi-class data classification problem. *European Journal of Operational Research*, 173(3), 910-920.

Unler, A., Murat, A. & Chinnam, R. (2011). mr^{2}PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, *181*(20), 4625-4641.

Vapnik, V. & Chapelle, O. (2000). Bounds on error expectation for support vector machines. *Neural Computation, 12*(9), 2013-2036.

Vapnik, V. & Vashist, A. (2009). A new learning paradigm: learning using privileged information. *Neural Networks: The Official Journal of The International Neural Network Society, 22*(5-6), 544–557.

Vapnik, V. (1995). The nature of statistic learning theory. NY-USA: Springer.

Vapnik, V. (1999). An overview of statistical learning theory. *Transactions on Neural Networks, 10*(5), 988-999.

Varewyck, M. & Martens, J.-P. (2011). A practical approach to model selection for support vector machines with a Gaussian kernel. *Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 41*(2), 330-340.

Venkatesan, S. & Karnan, M. (2010). Edge and characteristic subset selection in image using ACO. Proceedings of The 2nd International Conference on Computer Research and Development, 7-10 May 2010, Kuala Lumpur, Malaysia, 369 - 372.

Verleysen, M., Rossi, F. & Francois, D. (2009). Advances in feature selection with mutual information. In M. Biehl, B. Hammer, M. Verleysen & T. Villmann (Eds.), *Similarity-based clustering recent developments and biomedical applications*, (pp. 52-69). Berlin Heidelberg: Springer.

Verma, P. & Yadava, R.D.S. (2012). Fuzzy C-means clustering based uncertainty measure for sample weighting boosts pattern classification efficiency. Proceedings of The 2nd National Conference on Computational Intelligence and Signal Processing, 2-3 March 2012, Guwahati, Assam, 31-35.

Vieira, S., Mendonça, L., Farinha, G. & Sousa, J. (2013). Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. *Applied Soft Computing, 13*(8), 3494-3504.

Vieira, S., Sousa, J. & Runkler, T. (2007). Ant colony optimization applied to feature selection in fuzzy classifiers. In P. Melin, O. Castillo, L. Aguilar, J. Kacprzyk & W. Pedrycz (Eds.), *Foundations of Fuzzy Logic and Soft Computing*, (pp. 778-788). Berlin Heidelberg: Springer.

Wang, R. (2012). AdaBoost for feature selection, classification and its relation with SVM, a review. *Physics Procedia, 25*, 800-807.

Wang, S. & Meng, B. (2011). Parameter selection algorithm for support vector machine. *Procedia Environmental Sciences*, *11*(Part B), pp. 538-544, 2011.

Wang, T., Huang H., Tian S. & Xu J. (2010). Feature selection for SVM via optimization of kernel polarization with gaussian ARD kernels. *Expert Systems with Applications, 37*(9), 6663-6668.

Wang, X., Liu, X., Pedrycz, W., Zhu, X. & Hu, G. (2012). Mining axiomatic fuzzy association rules for classification problems. *European Journal of Operational Research, 218*(1), 202-210.

Wang, Z. & Chen S. (2007). New least squares vector machines based on matrix patterns. *Neural Processing Letters*, 26(1), 41-56.

Wei, K., Tuo, H. & Jing, Z. (2010). Improving binary ant colony optimization by adaptive pheromone and commutative solution update. Proceedings of The 5th Bio-Inspired Computing: Theories and Applications, 23-26 Septmber 2010, Changsha, 565-569.

Weiss, N. (2008). Introductory statistics (8th Ed.). Boston, USA: Pearson, Addison Wesley.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. & Vapnik, V. (2001). *Feature selection for SVMs*. Retrieved December 20, 2010, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.7476.

Wu, Z. & Li, C. (2006). Feature selection with transductive support vector machines. In I. Guyon, M. Nikravesh, S. Gunn & L. Zadeh (Eds), *Feature extraction foundation and applications,* (pp. 325-341). Berlin Heidelberg: Springer.

Xie, J. & Wang, C. (2011). Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases. *Expert Systems with Applications, 38*(5), 5809-5815.

Xiong, S., Wang, J.-Y. & Lin, H. (2010). Hybrid feature selection algorithm based on dynamic weighted ant colony algorithm. Proceedings of The 9th International Conference on Machine Learning and Cybernetics, 11-14 July 2010, Qingdao, 518 - 522.

Xiong, W. & Wang, C. (2008). Feature selection: a hybrid approach based on self-adaptive ant colony and support vector machine. Proceedings of The International Conference on Computer Science and Software Engineering, 12-14 December 2008, Wuhan, Hubei, 751 - 754.

Xiong, W. & Wang, C. (2009). A hybrid improved ant colony optimization and random forests feature selection method for microarray data. Proceedings of The 5th International Joint Conference on INC, IMS and IDC, 25-27 August 2009, Seoul, Korea, 559 - 563.

Xuan, W., Jiake, L. & Deti, X. (2010). A hybrid approach of support vector machine with particle swarm optimization for water quality prediction. Proceedings of The 5th International Conference on Computer Science and Education, 24-27 August 2010, Hefei, 1158 - 1163.

Xue, X., Sun, W. & Peng, C. (2010). Improved ant colony algorithm for continuous function optimization. Proceedings of The Control and Decision Conference, 26-28 May 2010, Xuzhou, 20 – 24.

Xusheng, G., Wei, C. & Yongxiang, L. (2010). Parameter selection for SVM using niches genetic algorithm in determined range. Proceedings of The 2nd International Conference on Information Management and Engineering, 16-18 April 2010, Chengdu, 492 - 496.

Yang, L., Fu, Z.-Q., Wang, D., Li, H.-L. & Xia, J.-B. (2010). An improved ant colony algorithm for continuous space optimization. Proceedings of The International Conference on Machine Learning and Cybernetics, 11-14 July 2010, Qingdao, 1829–1832.

Yang, Y., Han, C. & Han, D. (2009). A novel feature line segment approach for pattern classification. Proceedings of The 12th International Conference on Information Fusion, 6-9 July 2009, Seattle, WA, 490 - 497.

Ye, Y., Chen, L., Wang, D., Li, T., Jiang, Q. & Zhao, M. (2009). SBMDS: an interpretable string based malware detection system using SVM ensemble with bagging. *Journal in Computer Virology, 5*(4), 283-293.

Yuxia, H. & Hongtao, Z. (2012). Chaos optimization method of SVM parameters selection for chaotic time series forecasting. *Physics Procedia, 25*, 588-594.

Zavar, M., Rahati, S., Akbarzadeh-T, M.-R. & Ghasemifard, H. (2011). Evolutionary model selection in a wavelet-based support vector machine for automated seizure detection. *Expert systems with Applications, 38*(9), 10751-10758.

Zhang, B., Pham, T. & Zhang, Y. (2007). Bagging support vector machines for classification of SELDI-TOF mass spectra of ovarian cancer serum samples. In M. Orgun & J. Thornton (Eds.), *Advances in artificial intelligence*, (pp. 820-826). Berlin Heidelberg: Springer.

Zhang, G. & Li, H.-X. (2010). A probabilistic fuzzy learning system for pattern classification. Proceedings of The International Conference on Systems Man and Cybernetics, 10-13 Oct. 2010, Istanbul, Turkay, 2336 - 2341.

Zhang, H. & Mao, H. (2009). Feature selection for the stored-grain insects based on PSO and SVM. Proceedings of The 2nd International Conference on Knowledge Discovery and Data Mining, 23-25 January 2009, Moscow, Russia, 586-589.

Zhang, H., Xiang, M., Ma, C., Huang, Q., Li, W., Xie, W., Wei, Y. & Yang, S. (2009). Three-class classification models of LogS and LogP derived by using GA-CG-SVM approach. *Molecular Diversity, 13*(2), 261-268.

Zhang, J., Chen, W., Zhong, J., Tan, X. & Li, Y. (2009). Continuous function optimization using hybrid ant colony approach with orthogonal design scheme. In Tzai-Der W. (Eds.), Simulated Evolution and Learning, *The Sixth International Symposium on Neural Networks (ISNN 2009)*, (pp. 126-133). Berlin Heidelberg: Springer.

Zhang, X. & Jiao, L. (2005). Simultaneous feature selection and parameters optimization for SVM by Immune Clonal Algorithm. In L. Wang, K. Chen & Y. Ong (Eds.), *Advances in Natural Computation*, (pp. 905-912). Berlin Heidelberg: Springer.

Zhang, X., Chen, X. & He, Z. (2010). An ACO-based algorithm for parameter optimization of support vector machines. *Expert System with Applications, 37*(9), 6618-6628.

Zhang, X., Chen, X., Zhang, Z. & He, Z. (2008). A grid–based ACO algorithm for parameters optimization in support vector machines. Proceedings of The International Conference on General Computing, 26-28 August 2008, Hangzhou, 805-808.

Zhang, Y. (2008). Evolutionary computation based automatic SVM model selection. Proceedings of The 4th International Conference on Natural Computation, 18-20 October 2008, Jinan, 66-70.

Zhang, Y., Zhang, C., Ma, P. & Su, X. (2010). A hybrid and ensemble intelligent pattern classification algorithm. Proceedings of The 1st International Conference on Pervasive Computing, Signal Processing and Applications, 17-19 September 2010, Harbin, 833 - 836.

Zhao, M., Fu, C., Ji, L., Tang, K. & Zhou, M. (2011). Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert System with Applications, 38*(5), 5197-5204.

Zhou, J., Zhang, A. & Bai, T. (2008). Client classification on credit risk using rough set theory and ACO-based support vector machine. Proceedings of The 4th International Conference on Wireless Communications, Networking and Mobile Computing, 12-14 October 2008, Dalian, 1-4.

Zhou, X. & Xu, J. (2002). A SVM model selection method based on hybrid genetic algorithm and empirical error minimization criterion. In H. Wang, Y. Shen, T. Huang & Z. Zeng (Eds.), *The Sixth International Symposium on Neural Networks,* (pp. 245–253). Berlin Heidelberg: Springer.

Zong, Q., Liu, W. & Dou, L. (2006). Parameters selection for SVR based on PSO. Proceedings of The 6th World Congress on Intelligent Control, 21-23 July 2006, Dalian, China, 2811-2814.