ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Ezgi ZORARPACI

A HYBRID APPROACH OF DIFFERENTIAL EVOLUTION AND ARTIFICIAL BEE COLONY FOR FEATURE SELECTION

DEPARTMENT OF COMPUTER ENGINEERING

ADANA, 2014

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

A HYBRID APPROACH OF DIFFERENTIAL EVOLUTION AND ARTIFICIAL BEE COLONY FOR FEATURE SELECTION

Ezgi ZORARPACI

MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on 24/12/2014.

…………………………….	………………………..……	…………………………….
Prof. Dr. Süleyman GÜNGÖR	Assoc. Prof. Dr. Selma Ayşe ÖZEL	Prof. Dr. Vedat PEŞTEMALCİ
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

Not: The usage of the presented specific declerations, tables, figures, and photographs either in this thesis or in any other reference without citiation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic

ABSTRACT

MSc THESIS

A HYBRID APPROACH OF DIFFERENTIAL EVOLUTION AND ARTIFICIAL BEE COLONY FOR FEATURE SELECTION

Ezgi ZORARPACI

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Prof. Dr. Süleyman GÜNGÖR
Year: 2014, Pages: 63
Jury : Prof. Dr. Süleyman GÜNGÖR
: Assoc. Prof. Dr. Selma Ayşe ÖZEL
: Prof. Dr. Vedat PEŞTEMALCİ

In this study, a hybrid method which combines Artificial Bee Colony Optimization Technique with Differential Evolution Algorithm is proposed for feature selection problem of classification tasks. The developed hybrid method was experimented on fifteen datasets from the UCI Repository which are commonly used in classification problems. The proposed hybrid feature selection method was also compared with the three most popular feature selection techniques that are Information Gain, ChiSquare and Correlation Feature Selection to evaluate its performance. The aim of this study is to reduce the number of features to be used during the classification process to improve run-time performance and accuracy of the classifier. The experimental results of this study showed that our developed hybrid method was able to select good features for classification tasks.

Key Words: Feature Selection, Hybrid Optimization, Artificial Bee Colony, Differential Evolution

ÖZ

YÜKSEK LİSANS TEZİ

NİTELİK SEÇİMİ İÇİN DİFERANSİYEL GELİŞİM VE YAPAY ARI KOLONİSİNİN BİR HİBRİT YAKLAŞIMI

Ezgi ZORARPACI

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Prof. Dr. Süleyman GÜNGÖR
Yıl: 2014, Sayfa Sayısı: 63
Jüri : Prof. Dr. Süleyman GÜNGÖR
: Doç. Dr. Selma Ayşe ÖZEL
: Prof. Dr. Vedat PEŞTEMALCİ

Bu çalışmada, sınıflandırma işlemlerindeki nitelik seçme problemi için Yapay Arı Kolonisi Optimizasyon Tekniği ile Diferansiyel Gelişim Algoritmasını birleştiren bir hibrit yöntem önerilmektedir. Önerilen hibrit yöntem sınıflandırma problemlerinde sıklıkla kullanılan, UCI Veri Kaynağı'ndan elde edilen on beş veri kümesi üzerinde test edilmiştir. Aynı zamanda, önerilen hibrit yöntem performans değerlendirmesi için popüler üç nitelik seçme tekniği olan Ki-kare (ChiSquare), Bilgi Kazancı (Information Gain) ve Korelasyon Nitelik Seçimi (Correlation Feature Selection) ile kıyaslanmıştır. Bu çalışmanın amacı çalışma zamanı performansını ve sınıflandırıcının doğruluğunu iyileştirmek için sınıflandırma işlemi sırasında kullanılan nitelik sayısını azaltmaktır. Bu çalışmada elde edilen deney sonuçları geliştirdiğimiz hibrit yöntemin sınıflandırma işlemleri için iyi nitelikleri seçebildiğini göstermiştir.

Anahtar Kelimeler: Nitelik Seçimi, Hibrit Optimizasyon, Yapay Arı Kolonisi, Diferansiyel Gelişim

ACKNOWLEDGEMENTS

First of all, I would like to thank to my advisor Prof. Dr. Süleyman GÜNGÖR and the member of MSc thesis jury Assoc. Prof. Dr. Selma Ayşe ÖZEL, for their supervision guidances, useful suggestions and spending of their valuable times for this work. And also, I would like to thank Prof. Dr. Vedat PEŞTEMALCİ who is the other member of MSc thesis jury for his suggestions and corrections.

Finally, I would like to thank my family for their endless support and encouragements for my life and career.

CONTENTS **PAGE**

LIST OF TABLES PAGE

LIST OF FIGURES **PAGE**

1. INTRODUCTION

Feature selection, also known as attribute selection or dimension reduction, is the method of selecting an optimum subset of relevant features which represents original feature set with the least error for learning model construction. Thanks to feature selection techniques, we have some benefits such as improved model interpretability, shorter training times, enhanced generalization by reducing overfitting when constructing learning models (He et al., 2009). As a result of these benefits, many feature selection methods have been proposed in literature. These methods are traditionally categorized as *wrapper* and *filter* techniques according to how the method is used. When a classifier is used to evaluate the generated feature subsets, it is called as *wrapper* approach. When feature subsets are evaluated according to their information content or statistical measures, it is called as *filter* method (Palasinamy and Kanmani, 2012). Filter methods are usually faster than wrapper methods since they have lower computational cost. However wrapper methods have usually better performance than filter methods because of selecting more representative features from the original feature set (Grande et al., 2007).

Theorically, a feature selection method must search through the subsets of features, and find the best one among the all candidate subsets according to a certain evaluation criterion. If n features exist in the feature set, optimal feature selection process needs to evaluate 2^n feature subsets to select the best subset. However, this procedure can not be possible. Since it is too costly and resrictive in general. So, instead of the best one, a (sub)optimum feature subset not reducing or least reducing classification accuracy may be accepted. Heuristic and random search methods can be applied to find these (sub)optimum subsets. Various metaheuristic search methods have been used to search solution space including Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Differential Evolution (DE), and Artificial Bee Colony (ABC) for feature selection problem (Frohlich et al., 2003). These heuristic models employ different strategies to keep the balance between exploration and exploitation of searching. The property of exploration provides to discover distinct areas in the search space. On the other hand, the property of exploitation allows to scan the local search space for better solutions. In some of these optimization techniques the exploration process performs well, while in others exploitation process performs well. For this reason, hybridization can be used to increase the performance of the algorithm. In hybridization, good properties of at least two techniques are combined to enchance the performances of these models. In this study, our aim is to use ABC and DE, which are recently developed good metaheuristic techniques, to develop a new hybrid wrapper method to improve the performance of general classification tasks. In literature, different forms for hybridization of heuristic algorithms have been developed for feature selection problem, but only a few of them includes combination of Artificial Bee Colony (ABC) and Differential Evolution (DE) algorithms.

Differential Evolution Algorithm (DE) (Storn and Price, 1997) is a metaheuristic algorithm, introduced by Rainer Storn and Kenneth Price, that optimizes a problem by iteratively trying to improve a candidate solution according to a given measure of quality. DE is used for multidimentional real-valued optimization problems. Also, DE presents several advantages compared to other optimization techniques. It requires only fewer parameters setting, it is fast, robust and applicable to high-dimensional complex optimization problems. Even though DE is efficient, it has some drawbacks such as unstable convergence and easy to hang out at local optimum (Wu et al., 2011).

Artificial Bee Colony Algorithm (ABC) (Karaboga, 2005), defined by Dervis Karaboga, is an optimization algorithm that mimics the intelligent foraging behaviour of honey bee swarm. ABC algorithm has good properties such as easy to implement, strong robustness, high flexibility and fewer control parameters (Bolaji et al., 2013). Also, ABC algorithm shows good performance at exploitation with onlooker bee processing phase. However, this process may result to increase in convergence time and algorithm can not exhibit it’s own real performance (Gao et al., 2011). To obtain better results than standart ABC and DE algorithms, strong properties of these two algorithms can be combined.

In this study, a hybrid method of ABC and DE, which uses logical operators for combination of ABC and DE algorithms, was proposed to find the best features and improve performance of the classification algorithms by reducing training and testing times without reducing the classification accuracy. In the hybrid method, DE processes which consist of logical mutation, recombination and selection were combined with logical producing of neighborhood of food source of ABC. Also, a modified onlooker bee process of ABC was embedded into the method to provide diversity of feature subset solutions. The proposed hybrid method was compared with standart ABC and DE algorithms in terms of stability and solution quality by running 10 times with identical seed values for each algorithm, and they were tested on fifteen standart datasets, commonly used by researchers for classification tasks and obtained from University Of California Irvine (UCI) Machine Learning Repository (http://archive.ics.uci.edu/ml/). Moreover, the selected features found by our proposed hybrid method were compared with that of three most popular attribute selection algorithms such as ChiSquare (CHI), Information Gain (IG) and Correlation Feature Selection (CFS) to evaluate the quality of them using different classication algorithms namely NaiveBayes and RBF Networks.

This thesis is organized as follows: Initially, previous works related to feature selection are described in Section 2. The proposed hybrid method of ABC and DE for feature selection is then presented in detail in Section 3. Experimental results obtained through the proposed hybrid method, standart ABC and DE algorithms, and the comparision of the proposed hybrid method with CHI, IG, and CFS feature selection techniques on the datasets are given and discussed in Section 4. At the end of thesis, final remarks and directions for future work are presented.

2. PRELIMINARY WORKS

A prominent problem of the classification tasks is the high dimensionality of the feature space (Shang et al., 2007). In this case we should incorporate only relevant features and discard less relevant or irrelevant of them. To cope with this selection procedure, several search strategies have been developed. Referring to literature, search methods based on rankings with feature scoring measures such as Information Gain, ChiSquare etc. have been used for selecting attributes. But, a joint disadvantage of these filter methods is that they neglect the mutual effect with the classifier model, and this may lead to decreased classification performance when compared to other types of feature selection methods. While filter methods handle with the issue of finding a good feature subset without depending on the classifier model, wrapper methods accomodate the classifier model into their search strategy. In this manner, it is possible that a search procedure evaluates a specific subset of features by using classification model and can obtain better classification performance. Since the size of the feature subset space is proportional to 2^n, where n is the number of features in the feature set, the computational time for search strategy which finds the best feature subset among all 2^n feature subsets is too much in wrapper methods. Therefore meta-heuristic search methods, which will discover a (sub)optimum feature subset, can be used with a classifier model to construct a wrapper method (Saeys et al., 2007). In literature several meta-heuristic search methods including ABC and DE have been proposed for feature selection problem.

2.1. Preliminary Works in Feature Selection Based on Artificial Bee Colony Optimization Technique and Differential Evolution Algorithm

Khushaba et al. (2008) have presented a novel feature selection algorithm based on Differential Evolution (DE). In this study, a real number optimizer has been used, and DE operators have been applied to indices of the features. Therefore, they have encountered the same features multiple times in the solution vector. This case is completely unacceptable for feature selection problems, as a certain feature is used more than one. In order to overcome this problem, they have proposed to employ feature distribution factors to replace duplicated features, hence a roulette wheel weighting scheme has been utilized. The proposed algorithm has been tested against other search algorithms like GA and PSO with a predetermined number of selected features that varies between 3 to 70. EEG signal classification dataset obtained from the Department of Medical Informatics, University of Technology, Austria has been used for the experiment, and 92.64% accuracy value has been recorded for the proposed method while 89.90% and 88.64% accuracy values have been obtained for GA and PSO respectively.

He et al. (2009) have suggested a discrete feature selection method using the DE algorithm. They have converted the real values obtained from applying of DE operators to the binary values, which are 0 and 1, by using Sigmoid Limiting Function. After the selection process, these selected features have been evaluated based on the idea of mutual information. Experiments have been performed using Support Vector Machine (SVM), C&R Tree and RBF Networks classifiers with six standart datasets namely Vote, Zoo, Flare, Breast, Lung, Exactly from UCI Repository. They have obtained accuracy values changing between 46.1% and 94.8% for the datasets.

Palanisamy and Kanmani (2012) have performed a wrapper based feature selection approach for classification problem. According to this approach, ABC is used as a feature selector and generates the feature subsets, and a classifier (e.g., J48) is used to evaluate each feature subset generated by the ABC algorithm. This study has been implemented and tested using 10 datasets from the UCI Repository. Hence, they have showed that algorithm has resulted in reduced feature size of the feature subset, increased classification accuracies, and low computational complexity.

Prasartvit et al. (2013) have proposed a novel method of ABC for data dimension reduction in classification problems. The proposed method applies ABC wrapping with a k-Nearest Neighbor (kNN) classifier. kNN has been used for evaluation criteria to compute the fitness value of the new feature subsets generated by ABC. In this method, employed bees and onlooker bees have generated new candidate food sources, which are the subsets of selected features, and kNN has been used to evaluate the classification accuracy (objective function) of the new candidate food sources. The proposed method has been validated in two distinct application domains: Gene Expression Analysis, and Autistic Behaviors. For Autistic Behaviors dataset, they have obtained 85% accuracy value with 25% features selected from original dataset. And for Gene Expression Analysis; the rates of genes (i.e., features) selected for Colon_Cancer, Acute_Leukemia, Hepatocellular_Carcinoma, High-grade_Glioma, and Prostate_Cancer datasets have been reduced to 3.15%, 3.39%, 4.38%, 3.61%, and 3.59% respectively with accuracy values changing between 89.5% and 100%.

Schiezaro and Pedrini (2013) have implemented a feature selection method using ABC to classification of different datasets. UCI datasets namely Heart-c, Hepatitis, Lung Cancer, Image Segmentation, Iris, Heart Statlog, Winsconsin, Labor and Diabetes have been used to demonstrate the effectiveness of the proposed method. With the proposed method, they have obtained accuracy values from 71.48% to 98.46%.

Uzer et al. (2013) have offered a hybrid approach that uses ABC for feature selection and SVM for classification. The purpose of this study was to examine the effect of elimination of unimportant and obsolete features from the datasets on the success of the classification process, using the SVM classifier. In the experiments, Hepatitis, Liver-Disorders, and Diabetes datasets from the UCI Repository have been used, and the proposed system has reached classification accuracies of 94.92%, 74.81%, and 79.29%, respectively.

Martinoyić et al. (2014) have been proposed a wrapper approach based on DE for dimensionality reduction. In this work, the feature subsets discovered by DE have been evaluated using a kNN classifier. Experiments have been conducted on twelve UCI Repository datasets. The classification accuracies for the proposed approach and some other optimization algorithms such as Angle Modulated DE (AMDE), GA, and DE have been compared. In this study, the accuracy values changing between 59% and 98% have been obtained by the proposed method outperforming AMDE, GA, and DE.

2.2. Preliminary Works in Feature Selection Based on Hybrid Meta-Heuristics

Oh et al. (2004) have developed a novel hybrid algorithm for feature selection. Local search operations that are Sequential Forward Search (SFS), Sequential Forward Floating Search (SFFS) and Polynomial-time Approximation (PTA) have been embedded into GA to better adjust the search process. The hybrid GA has showed better convergence property compared to standart GA. Experiments have been performed on various UCI Repository datasets including Glass, Wovel, Wine, Letter, Vehicle, Segmentation, WDBC, Ionosphere, Satellite, and Sonar. The results have showed that the proposed hybrid GA is more attractive for large-sized datasets compared to SFS, SFFS, PTA, and standart GA.

Chin (2007) has developed a new hybrid algorithm using GA and TS metaheuristics for feature selection of FUZZY ARTMAP NN classifier. In the proposed GA-TS algorithm, the memory structure of TS has been embedded into the searching process of GA. The frequency memory of TS has supported local search space in GA while the recency memory has supported global searching process in GA. A simulated noisy injection method has been used to evaluate the success of GA-TS hybrid algorithm. The new hybrid algorithm has been tested on medical diagnosis of Myocardinal Infraction and Acute Stroke, Thyroid, Wisconsin Prognostic Breast Cancer (WPBC), Pima Indian Diabetes (PID), Heart Disease, Ionosphere, German Credit Ranking, Single Proton Emission Computed Tomography, Hepatitis datasets obtained from public domain repositories. The results of the experiments have demonstrated that GA-TS algorithm performs better than conventinal GA in terms of accuracy and the number of reduced features.

Basiri and Nemati (2009) have suggested a novel hybrid algorithm for feature selection in text categorization by combining ACO and GA. In this study, after feature subsets have been discovered by using ACO, GA crossover and mutation operators have been applied to these subsets. After that, the feature subsets have been evaluated by using a classifier. These processes have been repeated iteratively by a predetermined number of times. They have tested the proposed algorithm on Reuters-21578 dataset by comparing with an existing ACO-based feature selection, IG, and CHI. The experiments have showed that the proposed method has outperformed ACO, IG and CHI algorithms.

Olabiyisi et al. (2012) have proposed a novel hybrid algorithm including GA and SA metaheuristics for feature extraction on timetabling problem. In the proposed algorithm, the selection process of SA has been used instead of GA selection process to prevent to hang out local optimum. The experimental results have showed that SA performs better than GA and hybrid of GA-SA in terms of optimality and run time. But for timetabling problem, runtimes of GA and SA are higher than runtime of the proposed hybrid method. Because of the running time performance, conclusions derived by the proposed hybrid algorithm has been more acceptable compared to GA and SA.

Akila et al. (2012) have built a hybrid wrapper and filter feature selection algorithm for classification problem using a combination of GA and Local Search (LS) technique. In this work, first LS has been performed using correlation based filter methods including discritizing, ranking and redundancy elimination with symmetrical uncertainity measure for feature subsets and then standart GA operators have been applied to these subsets. The success of the developed method has been tested on DNA Gene Analysis dataset obtained from UCI Repository and it has outperformed recent existing methods.

Jona and Nagaveni (2012) have proposed a new method to optimize the feature set by using a combination of PSO and GA, which is called Genetical Swarm Optimization (GSO) for Digital Mammogram datasets. Genetic operators have been used in PSO to prevent the disadvantages of PSO. In this study, experiments has been peformed on miniMIAS database and SVM classifier has been used for classification of mammograms. The success of GSO has been compared to GA and PSO through Receiver Operating Characteristics (ROC) curve. According to the results, the convergence of GSO has been better than GA and PSO. The accuracy values of 94%, 88% and 90% have been recorded by using selected attributes with GSO, GA and PSO respectively. After two years, they (Jona and Nagaveni, 2014) have developed a novel hybrid metaheuristic which is a combination of ACO and Cuckoo Search (CS), called Ant-Cuckoo Colony Optimization, for feature selection on Digital Mammogram datasets. To recruit the local search process of ACO, CS has been utilized. SVM classifier with Radial Basis Kernel Function (RBF) have been used to evaluate the feature subsets produced by ACO. The results have been tested on miniMIAS database by comparing with ACO and PSO algorithms. Ant-Cuckoo Colony Optimization has been showed 2% better accuracy than ACO and 4% better accuracy than PSO.

Babatunde et al. (2013) have constructed a hybrid algorithm of ACO and GA for classification task. They have used ACO algorithm for feature selection and applied GA operators (selection, crossover and mutation) to discover optimum feature subset. These feature subsets have been evaluated by using SVM classifier. The proposed algorithm has been tested on Face Detection dataset by comparing with ACO and GA. The results have demonstared that new hybrid technique outperforms ACO and GA.

Hasani et al. (2014) have developed a combination of Linear Genetic Programming (LPG) and Bees Algorithm (BA) for feature selection on Intrusion Detection Systems (IDS). In the proposed algorithm, LPG have been used to generate feature subset solutions and BA's neighborhood search process has been applied to these solutions. Finally, SVM has been used to evalute these feature subsets. The experimental results have demonstrated that LPG-BA method increases the accuracy and is more efficient than basic LPG and BA.

2.3. Preliminary Works Based on Hybrid ABC and DE Algorithms

Most of the population-based optimization techniques suffer from technique's shortcoming that requires a large population size to avoid premature convergence but resulting high computational time. This sometimes limits to explore and exploit the search space effectively. In order to obtain the most of the advantages of the nature inspired heuristic methods and to eliminate their disadvantages, hybridization is performed.

ABC is one of the most recent nature inspired algorithms based on foraging behavior of bees. ABC (Karaboga, 2005) has proven to be a robust and efficient algorithm for solving global optimization problems over continuous space. However, it has been observed that the structure of ABC supports exploration more in comparison to exploitation which has good local searching but, low convergence speed (Abraham et al., 2012). In order to create a balance between these two opposite factors, ABC can be used by combining with other techniques.

DE (Storn and Price, 1997) a novel version of GA, is a population-based direct search method for global optimization. DE uses real valued parameters, so it is easily applied to experimental problems changing from sciences to liberal arts. Also, DE has good convergence property (Gao and Lui, 2011). Hence, in literature some authors have preferred to combine ABC algorithm with DE.

A novel hybrid approach of ABC and DE has been presented for unconstrained optimization problems by Alizadegan et al. (2012). When combining these two algorithms, they have used ABC as a basic algorithm in their methods, and they have taken advantage of DE operators (i.e., population initilization, mutation and crossover operators/formulas). The structure of ABC-DE algorithm is the same as the ABC; but the new solution production process is similar to the DE. In fact, the way of producing new solutions in DE have been embedded into ABC. They have tested their proposed method using benchmark functions such as Sphere, Rosenbrock, Rastrigin, Griewank, Schwefel and Schaffer comparing with basic ABC and they have obtained better solutions for benckmark functions with hybrid approach.

Abraham et al. (2012) have employed a pipelining type hybrid method of ABC and DE. According to this method, for every generation of ABC, *n* best solution vectors from the current population based on the fitness values are selected to apply DE operators such as mutation, crossover, and selection, which allow scanning the local search space effectively. They have evaluated their proposal on 10 traditional benchmark functions, and compared to classical ABC and DE algorithms. According to their results, it has been observed that the performance of the hybrid algorithm has been superior to that of the classical ABC and DE over all benchmark functions.

Li et al. (2012) have proposed a hybrid algorithm of ABC and DE, in which the bee colony accelerating evolution process and the bee colony detecting operation have been added to the population evolution procedure of DE. In the algorithm, they have applied basic DE operators “mutation, crossover, selection” respectively to all individual in population. Then they have calculated fitness probability for each individual and multiplied with the number of individual in the population, which is called “cycling number”. After that, they have applied basic DE operators to each individual until their cycling numbers are reached. And finally, bee colony detecting operation has been performed. They have validated their method with the IEEE 14-bus and 30-bus systems data and better solutions have been found in ABC/DE hybrid algoithm compared to classical ABC and DE.

Xu et al. (2013) have proposed a new hybrid method by combining ABC optimization with DE algorithm. In this method, DE mutation operator has been applied to a certain percent of good solutions discovered by the ABC. Hence, they have presented a simple and efficient ABC algorithm and have compared this hybrid method with the classical ABC algorithm. They have observed that the experimental results are superior in the hybrid algorithm compared to ABC for benchmark functions.

Li et al. (2013) have proposed a hybrid of ABC and DE for solving reconfigurable antenna-array optimisation with quantised phase excitations. In this hybrid algorithm, they have constructed ABC algorithm with basic DE operators (mutation and crossover). In order to study the performance of the proposed hybrid method, they have compared the algorithm with the generalized ABC and DE. According to their experiments, better fitness values have been obtained in the hybrid algorithm compared to others.

Wang et al. (2013) have proposed a hybrid method of ABC and DE for portfolio optimization problem. According to this algorithm, for each iteration the phases of ABC such as employed bee, onlooker bee and scout bee processes have been applied to the solutions, after that these solutions have been subjected to the basic DE operators such as mutation and crossover. According to their analysis, the proposed hybrid method has higher accuracy values compared to other population based algorithms such as GA and PSO.

A new hybrid optimization method has been introduced to estimate the parameters in the biological models by Abdullah et al. (2013). The proposed method, which is called as Differential Evolutionary Bee Colony (DEBCO), has been developed to improve the neighboring searching strategy of the standard ABC using the evolutionary operations of DE. The proposed method has employed differential mutation and crossover operations to enhance the searching capability of the employed bee process of ABC. The performance of the proposed DEBCO method for parameter estimation problem has been evaluated using a complex biological model, namely the feedback regulation of lactose operons by bacterial cell. The results have showed that the proposed method is more capable of finding the parameters with significant accuracy and acceptable computational time compared to those of existing methods such as PSO, GA and ABC.

There exists only two study for feature selection problem including the combination of ABC and DE heuristics in literature. Yusoff et al. (2014) have proposed the hybrid algorithm of ABC and DE as feature selection technique. When combining these two algorithms, they have applied mutation and crossover operators of DE to the four food sources which can be abandoned in the scout bee process of ABC algorithm. Instead of moving randomly, they have moved by using existing solution in the search space. They have compared their results with other types of nature-inspired algorithms such as ACO and PSO. With their proposed hybrid method, they have obtained 100% and 98.44% of accuracy in training and testing phase respectively for the overian cancer dataset, and 100% and 94.44% of accuracy have been achieved for TOX dataset as well. Shanthi and Bhaskaran (2014) have suggested a new hybrid method which consists of ABC and DE. In this work, they have added the neighborhood production operator of ABC to the DE/current-to-rand/1 mutation operator to discover feature subset solutions. Self-adaptive Resource Allocation Network (SRAN) has been used for classification process with 10-folds crossvalidation. They have tested their method on Mamography Image Classification dataset. Performance of new hybrid method has been compared to ABC, ACO, and

GA. Their proposed hybrid method, ABC, ACO and GA has selected 42, 45, 56, 50 features from 84 attributes and has achieved 96.89%, 96.27%, 96.27% and 95.96% of accuracy respectively. However, the contributions of this thesis to the literature are described as follows:

i. The proposed hybrid method is the first in terms of usage of binary logical operators in hybridization of ABC and DE algorithms for feature selection problem.

ii. Generally, ABC and DE algorithms are used for real-valued problem solutions. Therefore, the operators of these algorithms are applied to real valued parameters of the problem, and then these real values are converted into binary values if problem solutions require binary form. However, these conversion processes can take a long time for the problems which have too many parameters. To eliminate this disadvantage, we have applied logical binary operators to the binary parameters in our study and we have eliminated the time required for real to binary value conversion process and thus, we have obtained the results in a short amount of time.

iii. In literature, the proposed methods have been experimented on at most ten UCI datasets in general. In our study, we have used more datasets and the proposed hybrid method has been experimented on fifteen UCI datasets.

iv. In literature, the hybrid algorithms of ABC and DE proposed for feature selection problem have not been compared with the popular filter feature selection methods namely CHI, IG and CFS. However, our proposed hybrid method has been compared with these filter feature selection techniques and by using different classifiers such as RBF Networks, and NaiveBayes.

3. MATERIAL AND METHOD

3.1. Material

This section consists of the UCI Repository datasets that were used in the experiments, Weka Data Mining Tool which was employed for evaluating the fitness of the selected features, ABC and DE optimization algorithms which were used in the proposed feature selection algorithm.

3.1.1 UCI Datasets

UCI datasets are frequently used by researchers for performance evaluation of the classification tasks. These datasets were downloaded from University Of California Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/). Totally fifteen datasets, namely Autos, Breast-w, Car, Glass, Heart-c, Dermatology, Hepatitis, ThoraricSurgery, Lymph, Credit-g, Sonar, Ionosphere, Liver-Disorders, Vote and Zoo were used in this study. The information about the number of classes, instances, and features for each dataset are presented in Table 3.1.

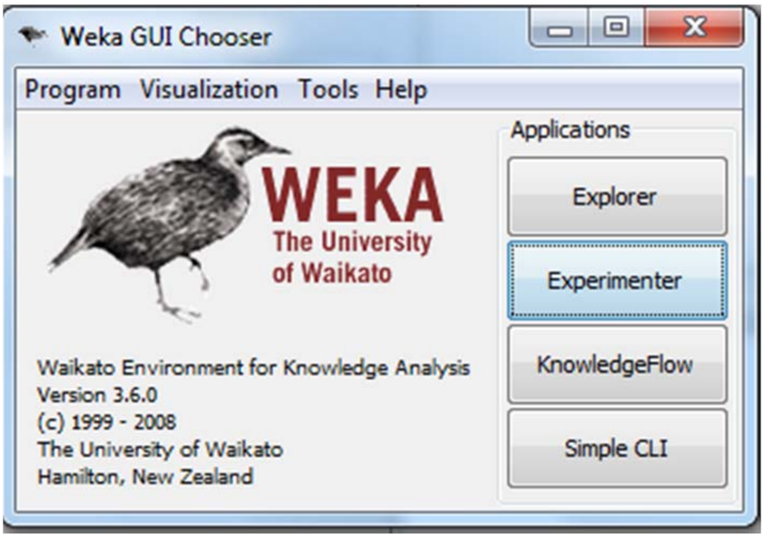
Table 3.1. Number of Classes, Instances and Features for each Dataset

Dataset	**# of Classes**	**# of Instances**	**# of Features**
Autos	6	159	25
Breast-w	2	683	9
Car	4	1728	7
Glass	6	214	9
Heart-c	2	296	13
Dermatology	6	358	34
Hepatitis	2	80	19
ThoraricSurgery	2	470	16
Lymph	4	148	18
Credit-g	2	1000	20
Sonar	2	208	60
Ionosphere	2	351	34
Liver-Disorders	2	345	6
Vote	2	222	16
Zoo	7	101	17

The data types of some UCI datasets are multivariate such as integer, real, and categorical; while those of some are univariate. For instance Autos, Heart-c, Dermatology, Hepatitis, ThoraricSurgery, Credit-g, Ionosphere, Liver-Disorders, and Zoo have multivariate data types, but other datasets have univariate data types.

3.1.2. Weka Data Mining Tool

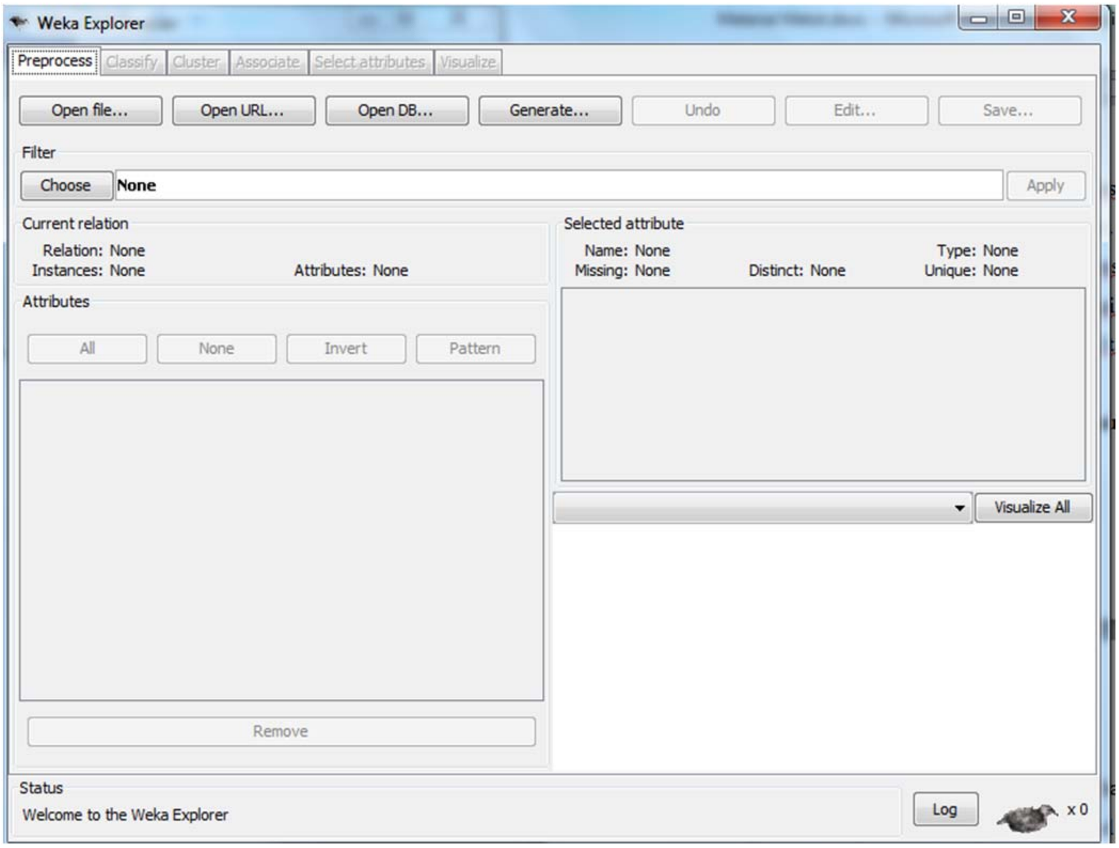
Weka (Waikato Environment for Knowledge Analysis) (http://www.cs.waikato.ac.nz/ml/weka) is a famous platform of machine learning software which was designed by using Java Programming Language and built in University of Waikato, New Zeland. It is free and open source software. The sample graphical user interface of Weka is shown in Figure 3.1. Although Weka can be used through its GUI, the source codes of it can also be called in our Java code. Weka includes feature selection, data preprocessing, clustering, regression, filtering, classification, and visualization tools.



Waikato Environment for Knowledge Analysis
Version 3.6.0
(c) 1999 - 2008
The University of Waikato
Hamilton, New Zealand

Figure 3.1. Sample graphical user interface of Weka

Weka Data Mining Tool has 4 general applications that are Explorer, Experimenter, KnowledgeFlow, and SimpleCLI with several subtasks. Explorer, Experimenter, and KnowledgeFlow have graphical user interface; while CLI has command line interface for performing data analysis. Explorer application consists of preprocessing, classification, clustering, association rule mining, attribute selection, and visualization main tasks. Preprocessing which is also called as “filters” can analyze and modify the data. Several classifiers (trees, rules, functions etc.) exist in the classification task. Clustering task includes different data clustering techniques such as SimpleKMeans etc. Association rule mining is performed by the associate task; whereas attribute selection algorithms are applied to data in the select attribute task. Finally, with visualization task, scatterplots for attribute values can be obtained. Weka Explorer Application GUI is shown in Figure 3.2.



Filter: None

Relation: None
Instances: None
Attributes: None

Name: None
Type: None
Missing: None
Distinct: None
Unique: None

Welcome to the Weka Explorer

Figure 3.2. Weka Explorer Application GUI

Experimenter component provides users to apply the same techniques in the Explorer part with different parameters or apply different analysis techiques to a data. Knowledge Flow task presents to users the data sources, data sinks, filters, classifiers, clusterers, associations, evaluations, and visulization processes. CLI is used if Weka is to run in command line interface. In this study, Weka was used for classification processes and filter based attribute selection processes that are CHI, IG and CFS to compare with our proposed feature selection technique.

3.1.3. Artificial Bee Colony Optimization Technique

Artificial Bee Colony (ABC) is a swarm based meta-heuristic algorithm that was introduced by Karaboga in 2005 for optimizing numerical problems (Karaboga, 2005). It simulates the intelligent foraging behavior of bees. Foraging model of honey bees includes some important constituent such as food sources, employed foragers, and unemployed foragers (Karaboga, 2005).

The quality of a food source is related to adjacency to the nest, its nectar concentration and convenience of extracting this nectar. Each employed forager exploits a particular food source and share informations such as nectar amount, distance and direction of own food source with other foragers. An unemployed forager always sights to exploit a food source. Unemployed foragers consist of scouts who look for new food sources and onlookers which wait to find a food source through information shared by employed foragers (Karaboga, 2005).

The sharing information among bees is performed in waggle dancing area. After an onlooker bee evaluates the information about rich sources, she makes a decision to exploit food source which is the most lucrative. Employed bees share information according to the proportional quality of food source (Karaboga, 2005). The foraging behaviour of bees is depicted in Figure 3.3.

Initially, a potential forager will be an unemployed bee. This forager does not have any information about the food sources around the nest. In this situation, two choices can be possible (Karaboga, 2005):

i. It can be a scout bee and searches for a food source according to some interior motivation or exterior clue. Scout bee is represented as “S” in Figure 3.3.
ii. It can move to find a new food source using waggle dances information. This bee is represented as “R” in Figure 3.3.

After finding the food source, bee will act as employed forager and exploit food source. Then this bee returns to hive for draining nectar to a food store. Following draining the nectar, the bee has three choices (Karaboga, 2005):

i. It can be an uncommited follower after abandoning the food source (UF).
ii. It can dance and then recruit nest mates before returning to the same food source (EF1).
iii. It can continue to exploit the food source without recruiting other bees (EF2).

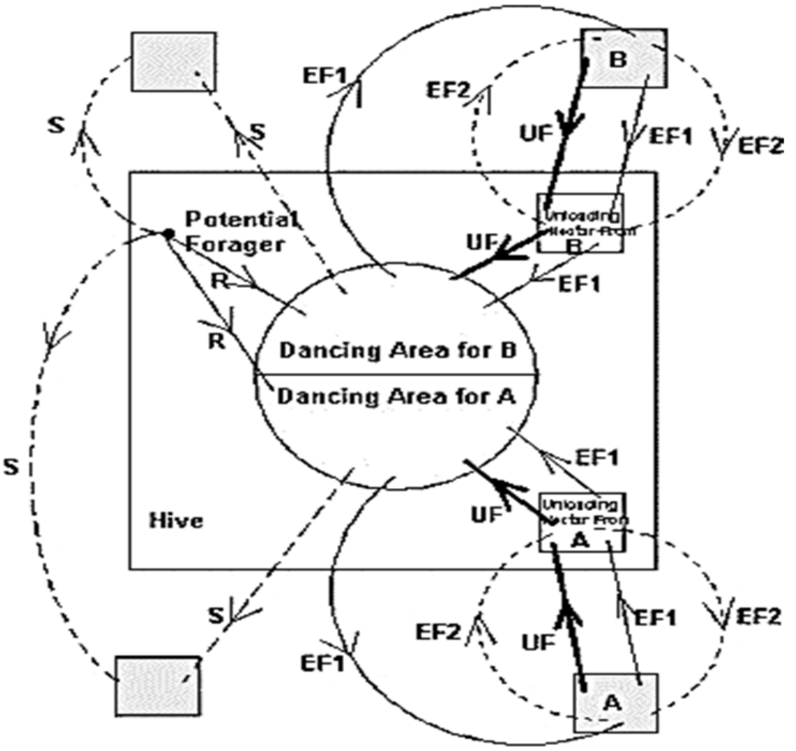


Figure 3.3. The foraging behavior of bees (Karaboga, 2005)

In ABC algorithm, the first half of the colony consists of the employed artificial bees and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources. Each employed bee goes to own food source and comes back to hive for dancing. Onlooker bees observe the dances of employed bees and specify food sources depending on the dances as well. When the food source of employed bee whose solutions cannot be improved through a predetermined number that is called “limit” has been abandoned, this employed bee turnes into a scout bee and it starts to search for finding a new food source.

According to the algorithm, a food source represents a potential solution (i.e., food source position) related with the problem and the nectar amount of a food source represents the quality (i.e., fitness) of the connected solution. The number of the employed bees or the onlooker bees and also food source are equal to the number of solutions in the swarm. The stages of ABC algorithm can be described as follows:

Step 1. *Initialization*: First, ABC generates a randomly distributed *SN* food source positions where *SN* denotes the size of employed bees or food sources. Each food source, defined as X_i, where $i=1,2,..,SN$ is a vector with dimension *D* which represents the number of parameters for the optimization problem. Generally, the beginning food source positions are randomly produced via equation 3.1.

$$X_i^j = X_{min}^j + rand(0,1).\left(X_{max}^j - X_{min}^j\right) \tag{3.1}$$

where, $j=1,2,\ldots,D$; X_{max}^j and X_{min}^j are the upper and lower bounds of the j th paremeter of the problem; $rand(0,1)$ is a random value between 0 and 1.

Step 2. *Nectar amount (i.e., fitness value) evaluations of the food sources:* In this step, the nectar amount (i.e., fitness value) is calculated for each food source.

Step 3. *Employed bee process:* After initialization, each employed bee goes to a food source and searches for a new food source having more nectar amount (i.e., quality) of own food source within its neighborhood. For an employed bee X_i, neighboring food source position is V_i which is produced by equation 3.2

$$V_i^{jrand} = X_i^{jrand} + rand[-1,1].\left(X_i^{jrand} - X_k^{jrand}\right) \tag{3.2}$$

where X_k is a randomly selected food source, $k \in \{1,2,\ldots,SN\}$ is randomly determined and has to be different from i, $jrand \in \{1,2,\ldots,D\}$ is a random integer number, and $rand[-1,1]$ is a random value between -1 and 1.

Step 4. *Nectar amount (i.e., fitness value) evaluation and selection*: After finding the neighboring food source, the nectar amount (i.e., quality) of new food source is calculated. If the nectar amount of the new food source is higher than that of the previous, the bee memorizes the new food source position (i.e., solution) and forgets the old. Otherwise she keeps the previous food source position in her memory.

Step 5. *Onlooker bee process:* After all employed bees have completed the search process; they share the nectar information of the food source with the onlooker bees. When an onlooker bee chooses a food source, it evaluates the nectar information taken from all employed bees and specifies a food source (X_i) with the probability value (p_i) related to its nectar amount. Therefore a random value which is between 0 and 1 is generated and compared with this probability value. If the probability value of a food source is higher than this random value, this food source is chosen by an onlooker bee and new neighboring food source is searched by using equation 3.2. For each X_i (food source), the probability value (p_i) is calculated by equation 3.3.

$$p_i = \frac{fitness_i}{\sum_{n=1}^{SN} fitness_n} \tag{3.3}$$

where $fitness_i$ is the quality (i.e., fitness value) of the food source (i.e., solution) i evaluated by its employed bee. After calculating the probability, onlooker bee X_i finds the neighboring food source according to the equation 3.2 and it evaluates the nectar amount of new candidate food source. If the nectar is higher than that of the previous, the bee memorizes the new source position (i.e., solution) and forgets the old.

Step 6. *Memorizing the best food source:* In this step, the best food source which has the highest nectar amount (i.e., fitness) is stored.

Step 7. *Scout bee process:* In scout bee process, a new food source is determined by a scout bee and replaces the abandoned one. For this process, a counter is used for each bee in the swarm. If there is a bee that her counter value exceeds maximum limit, she abandones the food source (i.e., solution) and searches for new food source. To search for a new food source, a scout bee uses equation 3.1.

The steps through 3 to 7 are repeated until a predetermined termination criterion is met. The best solution found so far will be the (sub)optimum solution for the problem. To simplify the understanding of the algorithm, a flow chart is shown in Figure 3.4. Important properties of ABC optimization technique are summarized as follows:

i. If quality of a solution increases, the rate of producing a new solution from this solution increases too.
ii. The exploration process for a solution whose counter has exceeded the limit value is ceased.
iii. Exploring new solution is performed using a random search process.

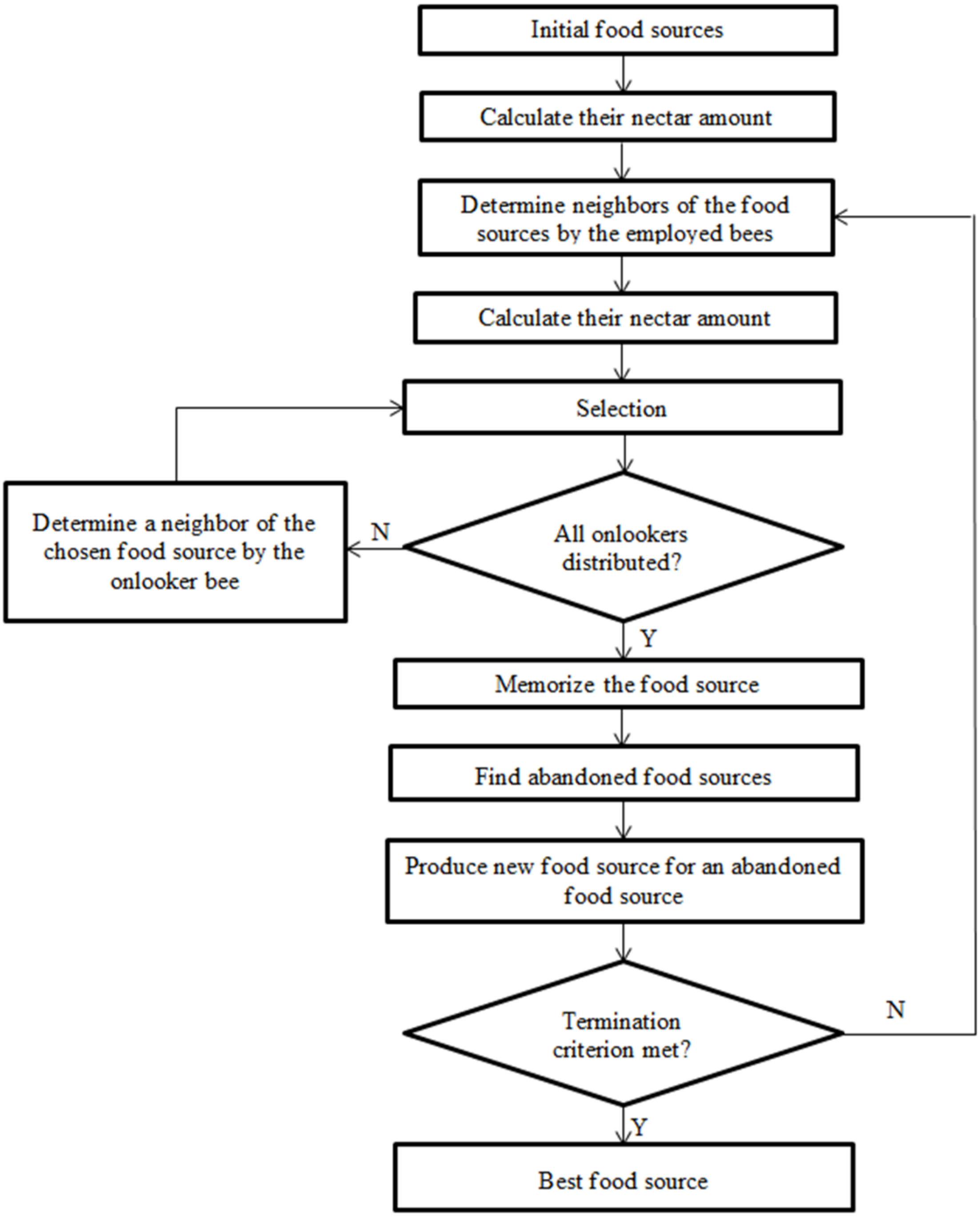


Figure 3.4. The flowchart of ABC algorithm (Zhang and Wu, 2011)

3.1.4. Differential Evolution Algorithm

Differential Evolution (DE) algorithm is a class of floating-point encoded evolutionary algorithms introduced by Storn and Price (1997) for global optimization problems over continous spaces (Price and Storn, 1997). It is a practical approach to global numerical optimization which is easy to understand, simple to implement, reliable, and requires only few control variables. Because of these reasons, DE has been a focus area of researchers in many scientific fields.

DE starts with the random initialization of a population of individuals in the search space, then the mutation and crossover operators and selection process are applied to individuals to generate a new population. The steps of DE are as follows:

Step 1. *Initilization*: The initial population should better cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum parameter bounds. Generally, this population is created by using equation 3.1.

Step 2. *Fitness evaluations of the individuals in the population:* In this step, fitness value is calculated for each individual in the population.

Step 3. *Mutation process:* DE employs the mutation operation to produce a mutant vector V_i with respect to *i*th individual in the population, X_i so-called source vector (i.e., individual), in the current population. The most frequently used mutation operator in the DE implementations is given in equation 3.4.

$$V_i^j = X_{r1}^j + F.\left(X_{r2}^j - X_{r3}^j\right) \tag{3.4}$$

where F is the scaling/mutation factor having values in the range of [0,2], X_{r1}, X_{r2}, X_{r3} are source vectors which are randomly chosen from the population, and $r1$, $r2$, $r3$ and i must be different from each other $(r1 \neq r2 \neq r3 \neq i)$.

Step 4. *Recombination (crossover) process*: After creating the mutant vector, crossover operation is applied to the source vector X_i and its corresponding mutant vector V_i to generate a trial vector U_i. This process is performed by using equation 3.5.

$$U_i^j = \begin{cases} V_i^j, & if(rand[0,1) \leq CR \; or \; j = j_{rand}) \\ X_i^j, & otherwise \end{cases} \tag{3.5}$$

where the crossover rate (*CR*) is a user-specified constant within the range of [0,1], which controls the fraction of parameter values copied from the mutant vector and j_{rand} is a randomly chosen integer in the range of [1, *D*]. D is the number of parameters of the related problem.

Step 5. *Fitness evaluation and selection*: The fitness function value of the trial vector is compared to that of source vector. If the trial vector has greater fitness function value than the source vector, the trial vector replaces the source vector and is included into the population of the next generation. Otherwise, the source vector remains unchanged in the population for the next generation. Steps 3, 4, and 5 are repeated for each individual until a predetermined termination criterion is met. The best solution in the population is the (sub)optimum solution. The flow chart of the DE algorithm is shown in Figure 3.5.

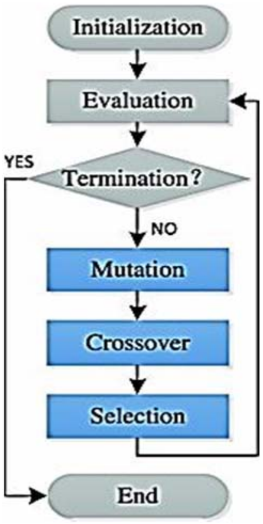


Figure 3.5. The flowchart of the DE algorithm (Deng et. al, 2013)

3.2. Method

This section explains our proposed hybrid ABC and DE algorithm. Detailed information about the combination of ABC and DE algorithms, the datasets used in the experimental evaluation and the classification algorithm applied for fitness evaluation are presented in this section of the thesis. The general components of the proposed method are shown in Figure 3.6.

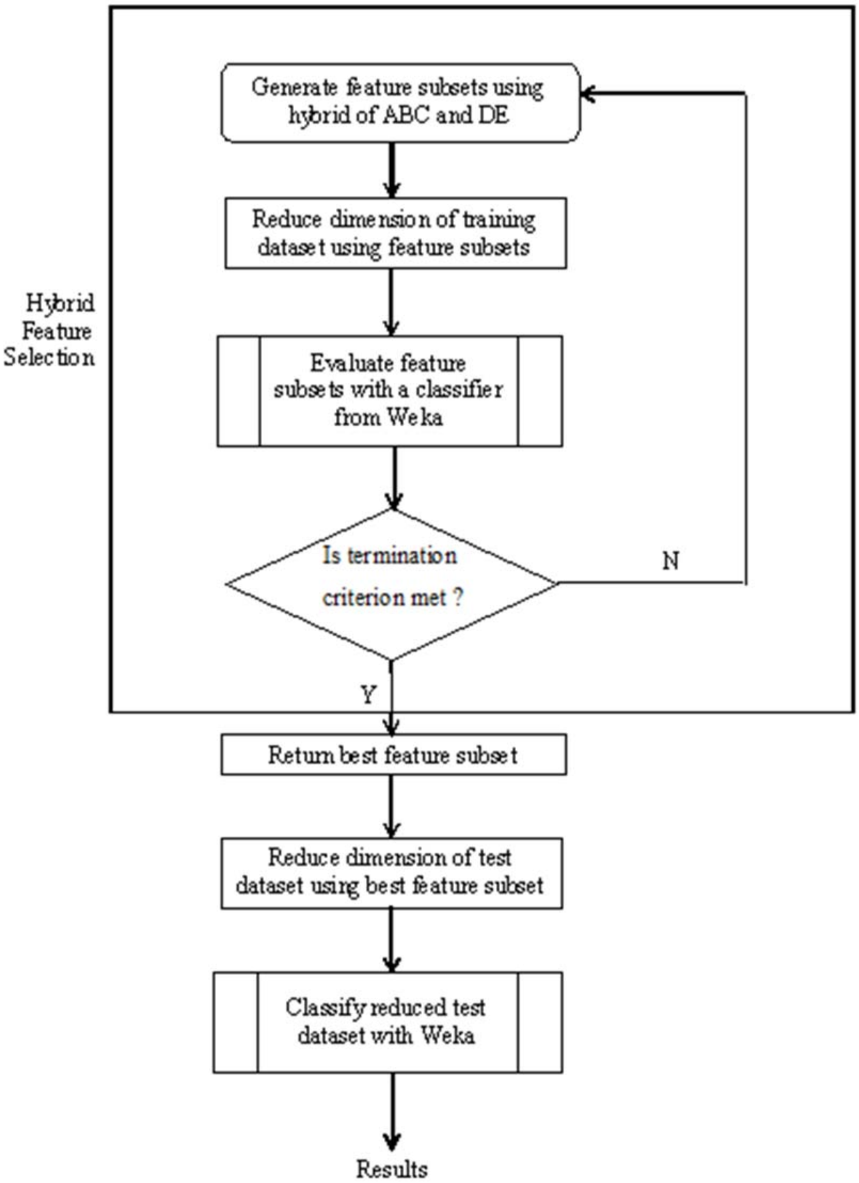


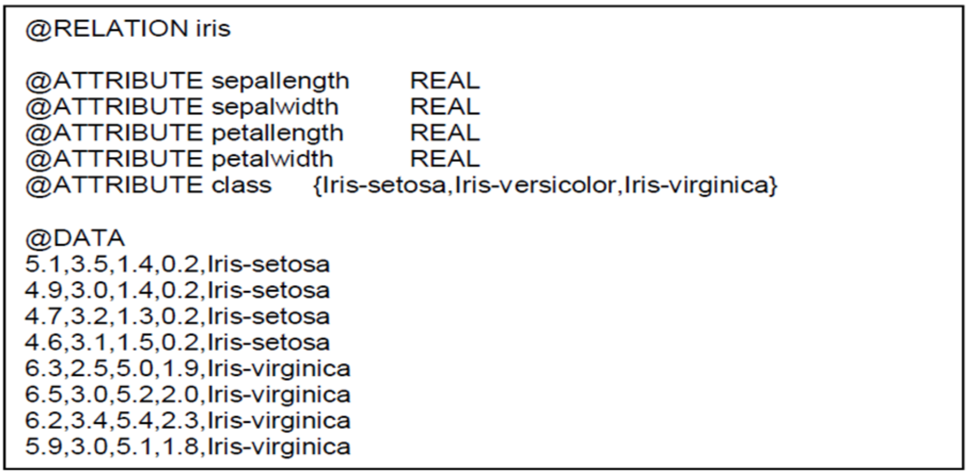
Figure 3.6. The general structure of the proposed algorithm

According to the architechure of our proposed method, first of all, the training dataset which is in the ATTRIBUTE-RELATION FILE FORMAT (ARFF) is read, and then our hybrid feature selection method is applied over the training data and the feature subset is determined. After that, the dimension of test dataset is reduced by using the selected set of features by the hybrid feature selection algorithm. Finally, the reduced test dataset is classified by using the Weka Data Mining Tool to measure the performance of the selected set of features.

Weka Data Mining Tool is used both in the hybrid feature selection algorithm, and in the test phase and the details of our proposal are explained in the following subsections.

3.2.1. Construction of Training and Test Datasets

An ARFF is a text file that represents a list of instances with attributes' values and it is used by Weka machine learning software. A sample ARFF for Iris dataset which is obtained from the UCI repository is shown in Figure 3.7. As shown in the figure, an ARFF consists of two parts namely header and data. The header part of the ARFF contains the name of the relation and name and type of the attributes. The relation name is defined in the first line of the ARFF file. Each attribute is declared with @attribute statement. This statement specifies the name of that attribute and its data type. The order of the attribute in the header section gives the column number of this attribute in the data section of the ARFF file. In the data section, each instance is written on a single line, and class label of this instance is denoted at the end of this line. Attribute values for each instance are delimited by commas. Missing attribute values are shown with question mark character.



```
@RELATION iris

@ATTRIBUTE sepallength      REAL
@ATTRIBUTE sepalwidth       REAL
@ATTRIBUTE petallength      REAL
@ATTRIBUTE petalwidth       REAL
@ATTRIBUTE class     {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
6.3,2.5,5.0,1.9,Iris-virginica
6.5,3.0,5.2,2.0,Iris-virginica
6.2,3.4,5.4,2.3,Iris-virginica
5.9,3.0,5.1,1.8,Iris-virginica
```

Figure 3.7. Iris.arff (http://www.cs.waikato.ac.nz/ml/weka)

In this study, fifteen different datasets from the UCI Repository were used. The numbers of instances and classes for each dataset were presented in Section

3.1.1. Randomly chosen approximately 75% of the instances in each dataset were specified as training instances, and the rest of the instances were used in the testing phase. The distributions of classes and instances in the training and test data for each dataset are shown in Table 3.2.

Table 3.2. Distributions of Classes and Instances for Datasets

Dataset	**# of Classes**			**# of Instances**	
	Training	**Testing**	**Total**	**Training**	**Testing**
Autos	6	6	6	121	38
Breast-w	2	2	2	514	169
Car	4	4	4	1296	432
Glass	6	6	6	160	54
Heart-c	2	2	2	222	74
Dermatology	6	6	6	267	91
Hepatitis	2	2	2	61	19
ThoraricSurgery	2	2	2	352	118
Lymph	4	4	4	112	36
Credit-g	2	2	2	750	250
Sonar	2	2	2	156	52
Ionosphere	2	2	2	263	88
Liver-Disorders	2	2	2	259	86
Vote	2	2	2	175	57
Zoo	7	7	7	76	25

3.2.2 Classifier and Performance Measure

In this thesis, we used C4.5 classifier to evaluate fitness values of the selected feature subsets. C4.5 (Quinlan, 1997) is a classification algorithm utilized to produce a decision tree. C4.5 is a tongue of Quinlan's earlier ID3 algorithm (Quinlan, 1993). C4.5 establishes decision trees from a set of training data as in the case of ID3 algorithm, using the entropy value of the attributes. Entropy is a measure of the amount of impurity in the dataset. The formula of the entropy is shown in equation 3.6.

$$H(S) = -\sum_{x \in X} p(x) log_2 \, p(x) \tag{3.6}$$

In equation 3.6, S is the dataset, X is set of classes in the S, and $p(x)$ is the rate of elements in class x to the number of elements in the set S. Information gain $IG(A,S)$ is the change in entropy from before to after the S is split on an attribute A. To calculate the information gain, equation 3.7 is used.

$$IG(A,S) = H(S) - \sum_{t \in T} p(t)H(t) \tag{3.7}$$

In equation 3.7, $H(S)$ is the entropy of the S, T specifies the subsets produced from splitting S by attribute A, $p(t)$ is rate of elements in t to the number of elements in S, and $H(t)$ is entropy of subset t. The attribute with the highest information gain is selected to make the decision. The C4.5 algorithm then repeats on the smaller sublists. The pseudocode of the general algorithm constructing the decision trees is given in Figure 3.8.

```
1. Check for base cases
2. For each attribute a
        2.1. Find the normalized information gain from splitting an a
3. Let a_best be the attribute with highest normalized information gain
4. Create a decision node that splits on a_best
5. Recurse on the sublists obtained by splitting on a_best, and add those nodes
as children of the node
```

Figure 3.8. Pseudo code of a general algorithm for decision trees (Quinlan, 1993).

J48 is an open source Java application of the C4.5 algorithm in the Weka Data Mining Tool. The decision tree which is built by the J48 algorithm for Iris dataset is shown in Figure 3.9.

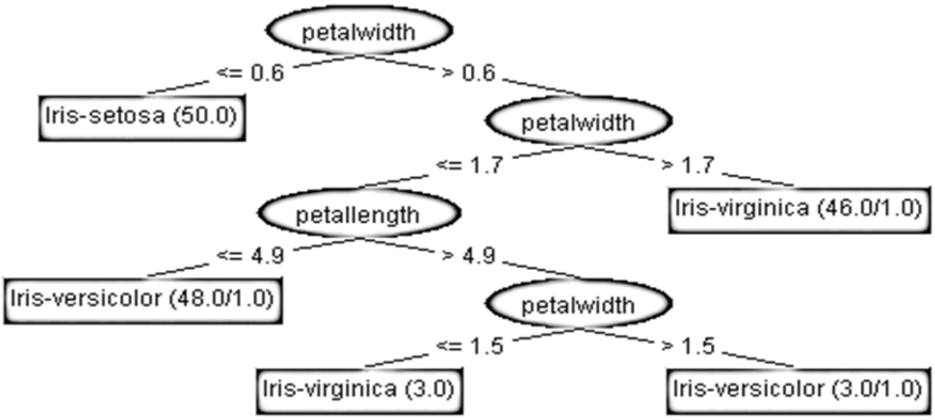


Figure 3.9. The decision tree built by J48 for Iris dataset.

In 2010, Saraç and Özel compared the classification performances of some classifiers namely J48, NaiveBayes, RBF Networks, Voted Perceptron, Threshold Selector, Voting Feature Intervals (VFI) for URL based Web page classification problem (Saraç and Özel, 2010). According to this study, J48 classifier had the highest classification F-measure for the Web page datasets. Therefore J48 classifier was chosen to evaluate the feature subsets of the proposed hybrid algorithm, standart ABC and DE algorithms.

In this study the feature subsets were evaluated with respect to F-measure values and the results for selected features were compared using F-measure values as well (Van Rijsbergen, 1979). F-measure value for a class is computed according to equation 3.10.

$$F - measure\ value = \frac{2 * precision * recall}{recall + precision} \tag{3.10}$$

In equation 3.10, precision for a class is the ratio of true positives to the total number of true positives and false positives. True positives are the number of items correctly labeled as belonging to the positive class, whereas false positives are the number of items incorrectly labeled as belonging to the positive class. Precision is the ratio of true positives to all items that are labelled as positives and its formula is given in equation 3.11. Recall is the ratio of true positives to all positive items in the dataset. False negatives are positive items which were labeled as not belonging to the positive class. The formula of recall is also given in equation 3.12.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{3.11}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3.12}$$

The popular metrics used to evaluate classification performance are accuracy and F-measure. Accuracy evaluates the classification performance by computing the proportion of the correctly classified instances among all instances. But, if the distributions of class labels are not equal in the dataset, accuracy value may be misleading. So, F-measure was used in this study.

3.2.3 The Proposed Hybrid Feature Selection Method

In this thesis, a new wrapper which is a hybrid method combining the superior properties of ABC and DE algorithms was proposed for feature selection problem of classification tasks. DE is simple and powerful evolutionary algorithm for global optimization problem (Storn and Price, 1997). It has been successfull in many fields of science. Owing to its appropriate structure to the parallel programming, it is applicable to the complex problems in large size. Also, DE surpasses many other optimization methods in terms of convergence speed and robustness. On the other hand, ABC algorithm and its different forms have been carried out successfully to unconstrained numerical optimization problems (Abraham et al., 2012). However, the low convergence speed of ABC may cause that the algorithm is unable to show its real performance. Becasue of these reasons, researchers have proposed various hybrid ways of these two algorithms for miscellaneous problem solutions in literature. But only two studies have been developed for solving the feature selection problem and none of them includes logical operators to resolve the feature selection problem. Therefore we thought that if we combine the high convergence speed of DE with the modified local search process of ABC using logical operators, we can obtain good results for feature selection problem in a short time. So, in this thesis we proposed a new hybridization of ABC and DE with logical operator to solve the feature selection problem.

The success of DE algorithm is very closely related to fine-tuning the control parameters of it (Sá et al., 2008). Specially, CR (Crossover Rate) is important in terms of balancing the local and global search processes. Low CR values support the local search process while high CR values support the global search process and decrease the time of convergence to optimum (Montgomery and Chen, 2010). Another important parameter is F (mutation/scaling factor). This factor is a real value which is between 0 and 2. The low values of F provide exploitation while the high values of it provide exploration and reduce the probability of dropping into local optimum (Mohamed et al., 2012).

ABC algorithm shows good performance at exploitation with onlooker bee processing phase which was described in section 3.1.3. But, this situation causes to increase the time of convergence and the algorithm can not show its real performance (Gao et al., 2011).

This study proposes a novel wrapper approach for feature selection of classification tasks by combining the strong exploration property of DE with a modified onlooker bee process of ABC algorithm. In this manner, we get rid of struggling to tune the control parameters for DE algorithm in order to solve the problems whose solutions are expressed in binary form. In addition to this, diversity in the population and exploitation process is supported for binary solution sets with the modified onlooker bee process of ABC algorithm. The main steps of the hybrid method are described as the following:

Step 1. *Determine the initial population of feature subset solutions*: In this step, feature subset solutions which are binary-coded solution vectors are created using binary random values 0 or 1. A sample of feature subset solution for a dataset which has 9 attributes is shown in Figure 3.10.

1	0	0	1	1	1	1	0	0

Figure 3.10. A sample of feature subset solution

Considering the feature subset solution structure which is shown in Figure 3.10, we have 9 features in total, and the selected feature indices are 1,4,5,6 and 7 and it means that classification task will be performed using these selected 5 features.

Step 2. *Fitness evaluations*: This step includes fitness evaluations of feature subset solutions. To calculate fitness value of a feature subset solution, first training data in ARFF format is read and the features which are indicated by 0 values on feature subset solution are removed from the training data. This filtered training data is classified using 3-folds crossvalidation with Weka J48 classifier function and the weighted average F-measure value which is returned from this function is assigned as the fitness value for the feature subset solution. In this step, for all feature subset solutions in the population, fitness value is calculated by the same way.

Step 3. *Calculating fitness probability values*: For each feature subset solution (i.e., individual) in the population, fitness probability value is calculated using equation 3.2 which was described in section 3.1.3.

Step 4. *DE/ABC producing of neighborhood operators*: In this step, the following two cases are considered for a feature subset solution (i.e., source individual):

i. If the fitness probability value of the source individual is greater than a random number which is produced between 0 and 1, DE mutation process is performed. So, three random individuals (X_{r1}, X_{r2}, X_{r3}) are chosen from the population for this source individual (X_i). These individuals should be different from each other and the source individual $(r1 \neq r2 \neq r3 \neq i)$. After selecting the individuals, the difference between two among the three randomly chosen individuals is found, this is called difference vector which is computed as follows: the jth dimension of difference vector will be 0 if the jth dimension of the first (X_{r1}) and the second (X_{r2}) vectors are equal to each other; otherwise the jth dimension of the difference vector is equal to the jth dimension of the first vector (X_{r1}). The formula of the difference vector is given in equation 3.13. Mutant vector is created for source individual after the computation of the difference vector. To construct mutant vector, "OR" logical operator is applied to the components of the randomly chosen third vector (X_{r3}) by using difference vector. Creating of mutant vector is described in equation 3.14. After creating the mutant vector, crossover between the source individual and the mutant vector is performed using equation 3.5 in section 3.1.4 and the trial vector is built. In our experiments, it is considered that CR is equal to 1. So, trial vector is equal to the mutant vector.

$$difference\ vector^j = \begin{cases} 0, & if\left({X_{r1}}^j = {X_{r2}}^j\right) \\ {X_{r1}}^j, & otherwise \end{cases} \tag{3.13}$$

$$mutant^j = \begin{cases} 1, & if\left(difference\ vector^j = 1\right) \\ {X_{r3}}^j, & otherwise \end{cases} \tag{3.14}$$

ii. Otherwise, to imitate the producing of neighborhood solution of ABC, for a source individual (X_i), two random individuals (X_{r1}, X_{r2}) are chosen from the population. These individuals should be different from each other and source individual $(r1 \neq r2 \neq i)$. After this selection process, a random component (*jrand*) for the source individual is specified and only for this component (*jrand*), the difference between the selected individuals (X_{r1} and X_{r2}) is found, which is called difference component. Equation 3.15 expresses how to find the difference component. Then a random value $r1(0,1)$ between 0 and 1 which simulates the coefficient of neighborhood solution production of ABC, is generated to decide whether "OR" logic operator will be applied to the component of source individual using difference component. Equation 3.16 explains how to construct the trial vector. In equation 3.16, $r2(0,1)$ is a random value which is generated between 0 and 1, j_{rand} is a randomly chosen integer in the range of [1, D]. D is the number of features in the dataset.

$$difference\ component = \begin{cases} 0, & if({X_{r1}}^{jrand} = {X_{r2}}^{jrand}) \\ {X_{r1}}^{jrand}, & otherwise \end{cases} \quad (3.15)$$

$$trial = X_i$$

$$trial^{jrand} = \begin{cases} 1, & if(difference\ component = 1\ and\ r1(0,1) > r2(0,1)) \\ {X_i}^{jrand}, otherwise & \end{cases} \quad (3.16)$$

Step 5. *Fitness evaluation and selection process*: The fitness value of trial vector is calculated using Weka J48 classifier with 3-folds crossvalidation; and weighted F-measure value returned from Weka is taken as the fitness value for the trial vector. After calculating the fitness value for the trial vector, this value is compared to that of source individual. If the trial vector has greater fitness function value than source individual or its fitness function value is equal to the source individual and the number of attribute of the trial vector is less than that of source individual, the trial vector will replace the source individual and enter into the population of the next generation. Otherwise, the source individual will remain in the population for the next generation.

Step 6. *Calculating fitness value probabilities*: After Step 4 and 5 are applied to all individual in the population, fitness probability value is calculated using equation 3.2 which was described in section 3.1.3 for each individual (feature subset solution).

Step 7. *Modified onlooker bee process*: In modified onlooker bee process, the fitness probability value (p_i) of an individual (X_i) is subtracted from 1, maximum probability value, and if this value (i.e., $1 - p_i$) is greater than a random value which is generated between 0 and 1, "NOT" logical operator is applied to the random component of the source individual, $source^{jrand}$. "NOT" operator is described in equation 3.17. After determining trial vector, fitness evaluation and selection process are performed as described in Step 5. Step 7 is repeated n times where n is equal to the half of the total number of individuals in the population.

$$trial = source$$

$$trial^{jrand} = \begin{cases} 0, if \ (trial^{jrand} = 1) \\ 1, \qquad otherwise \end{cases} \tag{3.17}$$

Steps 3, 4, 5, 6 and 7 are repeated until a predetermined termination criterion is satisfied. The best solution in the population will be the optimum solution. Apart from that, producing of neighborhood of food source operator which is explained in the second condition in Step 4 was used to build the standart ABC algorithm, and the DE operators which are described in the first condition of Step 4 were employed to employ the standart DE algorithm. After all, the standart ABC and DE algorithms, CHI, IG and CFS feature selection techniques were compared with the proposed hybrid method.

4. RESEARCH AND DISCUSSION

In this section, experiments performed and their results are explained in detail. Hybrid feature selection method which combines ABC and DE algorithms was implemented in Java programming language under NetbeansIDE 7.2 platform (https://netbeans.org/community/releases/72/). The proposed hybrid method was developed on a computer having Windows 7 Home Premium operating system, 4 GB of RAM, and Intel Core i5-2430 M 2.4 GHz processor. The proposed hybrid method was tested on fifteen UCI machine learning repository datasets (http://archive.ics.uci.edu/ml/). Detailed information about the datasets were presented in section 3.1.1. For the proposed method, the ABC, and DE methods, number of iterations was set to 300. After 300 iterations, there were not any improvements on classification performance for the proposed hybrid method or the standart DE algorithm. To prove the robustness of the proposed hybrid method and better evaluate the results with other algorithms, all of the algorithms have been run 10 times. The proposed method was compared to standart ABC and DE algorithms. These three methods have been run with the same seed values and population sizes for 300 iterations. Population size was set to 50 for these methods because the population sizes of 30 to 50 are widespread for studies about the DE algorithm. When population size was set to 50, our standart DE algorithm produced good results. Our pupose in this thesis is to find the best feature subsets in a short amount of time. So, standart ABC and the proposed hybrid method have been run with a constant population size which was set to 50 as well. At the same time, 10-folds crossvalidation was utilized for all algorithms in the testing phase.

The best and worst weighted F-measure values obtained by J48 classifier at the end of 10 runs in the test phase for ABC, DE and the proposed hybrid method, and also F-measure values with all attributes (i.e., without making any feature selection) are shown in Table 4.1. Number of selected features for the best and worst F-measure values and total number of attributes for each dataset are given in Table 4.2.

Table 4.1. Best/Worst F-measure Values Obtained in the Test Phase at the end of 10 runs

Dataset	Best/Worst F-measure Values			F-measure Values without Feature Selection
	ABC	DE	Hybrid Method	
Autos	**0.777**/0.333	0.751/0.618	0.642/**0.642**	0.633
Breast-w	**0.923**/0.905	**0.923**/0.893	0.922/**0.922**	0.899
Car	**0.834/0.834**	**0.834/0.834**	**0.834/0.834**	0.817
Glass	**0.728**/0.608	0.721/0.623	0.721/**0.721**	0.663
Heart-c	**0.770**/0.699	0.755/**0.713**	0.713/0.675	0.703
Dermatology	**0.921**/0.774	0.910/0.856	0.910/**0.889**	0.878
Hepatitis	**0.944**/0.611	**0.944**/0.611	**0.944/0.944**	0.833
ThoraricSurgery	**0.765/0.765**	**0.765/0.765**	**0.765/0.765**	**0.765**
Lymph	0.675/0.506	**0.704**/0.599	0.688/**0.640**	0.609
Credit-g	**0.761**/0.671	0.740/0.671	0.727/**0.681**	0.738
Sonar	0.885/0.572	0.902/0.580	**0.942/0.592**	0.760
Ionosphere	0.931/0.837	0.919/0.845	**0.942/0.851**	0.870
Liver-Disorders	**0.617/0.617**	**0.617/0.617**	**0.617/0.617**	0.547
Vote	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**	**0.964**
Zoo	**0.754**/0.685	0.740/0.685	0.740/**0.730**	0.732

When we compare the algorithms using Table 4.1, the worst case performance of the proposed hybrid method is better than standart ABC and DE algorithms for 14 datasets in terms of the worst F-measure values. In general, the best F-measure values of the hybrid algorithm are better than DE, but worse than or neart to ABC.

Table 4.2. Number of Selected Features for the Best/Worst F-measure Values Obtained in the Test Phase at the end of 10 runs

Dataset	# of Best/Worst Selected Features			Total # of Attributes
	ABC	DE	Hybrid Method	
Autos	10/**9**	11/10	**9/9**	25
Breast-w	**3**/6	**3**/7	5/**5**	9
Car	**5/5**	**5/5**	**5/5**	7
Glass	5/6	**4**/5	**4/4**	9
Heart-c	**3**/6	10/10	6/**5**	13
Dermatology	13/16	20/19	**9/7**	34
Hepatitis	**2**/9	6/8	4/**4**	19
ThoraricSurgery	**1**/12	7/11	5/**6**	16
Lymph	11/9	8/9	**3/4**	18
Credit-g	13/10	11/11	**7/9**	20
Sonar	24/14	38/31	**14/10**	60
Ionosphere	**10/7**	18/20	11/11	34
Liver-Disorders	**5/5**	**5/5**	**5/5**	6
Vote	2/9	**1**/5	**1/1**	16
Zoo	8/9	8/8	**7/6**	17

When we analyze Table 4.1 and Table 4.2 together, the best F- measure values of the proposed hybrid method are less than the standart ABC or DE algorithms only for 6 datasets namely Autos, Glass, Dermatology, Lymph, Credit-g, Zoo out of 15 datasets. But, the hybrid algorithm decreases the best F-measure values by 1.73%, 0.96%, 1.19%, 2.2%, 4.7%, 1.85% respectively, while it reduces the number of features by 4%, 11.1%, 11.7%, 27.7%, 30%, 5.8% respectively for these datasets. According to these results, we can say that the proposed hybrid method performs better than standart ABC and DE algorithms for these 6 datasets as well. On the other hand the proposed hybrid method was not successfull only for two datasets namely Breast-w, and Heart-c in terms of the best F-measure values. The F-measure values reduced by 0.1%, and 7.4% respectively when the features selected by our hybrid algorithm was used for the Breast-w, and Heart-c datasets.

In addition to the best and worst F-measure values obtained by the algorithms, average F-measure values and average numbers of selected features at the end of 10 runs are given in Table 4.3 and Table 4.4 respectively.

Table 4.3. Average F-measure Values at the end of 10 runs

Dataset	Average F-measure Values			F-measure Values without Feature Selection
	ABC	DE	Hybrid Method	
Autos	0.606	**0.669**	0.642	0.633
Breast-w	0.918	0.914	**0.922**	0.899
Car	**0.834**	**0.834**	**0.834**	0.817
Glass	0.675	0.675	**0.721**	0.663
Heart-c	**0.736**	0.725	0.700	0.703
Dermatology	0.863	0.881	**0.893**	0.878
Hepatitis	0.811	0.809	**0.944**	0.833
ThoraricSurgery	**0.765**	**0.765**	**0.765**	**0.765**
Lymph	0.590	0.650	**0.657**	0.609
Credit-g	0.711	0.700	0.700	**0.738**
Sonar	0.706	0.738	**0.781**	0.760
Ionosphere	0.873	0.890	**0.893**	0.870
Liver-Disorders	**0.617**	**0.617**	**0.617**	0.547
Vote	**0.964**	**0.964**	**0.964**	0.964
Zoo	0.718	0.711	**0.737**	0.732

Table 4.4. Average # of Selected Features at the end of 10 runs

Dataset	# of Selected Average Feature			Total # of Attributes
	ABC	DE	Hybrid Method	
Autos	10	12	**9**	25
Breast-w	**5**	**5**	**5**	9
Car	**5**	**5**	**5**	7
Glass	5	5	**4**	9
Heart-c	7	8	**5**	13
Dermatology	14	19	**7**	34
Hepatitis	5	6	**4**	19
ThoraricSurgery	7	8	**5**	16
Lymph	9	9	**4**	18
Credit-g	11	12	**8**	20
Sonar	15	32	**13**	60
Ionosphere	**8**	19	9	34
Liver-Disorders	**5**	**5**	**5**	6
Vote	4	2	**1**	16
Zoo	8	8	**6**	17

According to Table 4.3, the proposed hybrid method is sucessfull on almost all datasets except the two datasets “Heart-c” and “Credit-g” in terms of average F-measure value at the end of 10 runs. But, the training times decrease with the proportion of approximately 96% and 95% for Heart-c and Credit-g datasets respectively while average F-mesure values decrease with the proportion of only 0.4% and 5.4% respectively. According to this result, the F-measure values for Heart-c and Credit-g datasets can be acceptable since training times quite a lot. On the other hand, when we analyze Table 4.3 and Table 4.4 together, the average F-measure value obtained by the proposed hybrid method for Autos dataset is approximately 4% smaller than that of standart DE algorithm while the average number of selected features is 25% smaller than that of standart DE algorithm. Because of this reason, we can say that the proposed hybrid method outperforms standart DE in terms of average F-measure value for Autos dataset as well.

As shown in Table 4.2, and 4.4, the (sub)optimum numbers of selected features were determined by the ABC, DE, and hybrid algorithms. Since all these methods are wrapper techniques, they are able to minimize the number of selected features while trying to maximize the F-measure values.

In addition to the above experiments, the proposed hybrid method was compared with three most popular feature selection techniques that are ChiSquare (CHI), Information Gain (IG) and Correlation Feature Selection (CFS). As CHI, IG, and CFS are filter techniques, they can only rank the features in a given dataset. To select features by using these filter techniques, we need to specify the number of features to be selected, and we take the top-ranked features . In the experiments we used the number of selected features by our hybrid algorithm to select features by using these ranking methods. For the best and worst F-measure values for the proposed hybrid method, performance of CHI, IG, and CFS techniques are shown in Table 4.5. When we specify the numbers of features to be selected for CHI, IG and CFS techniques, we use the numbers of features in hybrid method column in Table 4.2. Therefore the numbers of features, which provide the best F-measure values in hybrid method, were used to find the best F-measure values of CHI, IG and CFS. And the numbers of features, which produce the worst F-measure values in hybrid method, were used to find the worst F-measure values of CHI, IG and CFS as well.

Table 4.5. Results Obtained by CHI, IG and CFS Techniques with the Selected Features which provide the Best/Worst F-measure Values in Hybrid Method

Dataset	Best/Worst F-measure Values			
	CHI	IG	CFS	Hybrid Method
Autos	**0.658/0.658**	**0.658/0.658**	**0.658/0.658**	0.642/0.642
Breast-w	0.917/0.917	0.917/0.917	0.917/0.917	**0.922/0.922**
Car	**0.834/0.834**	**0.834/0.834**	**0.834/0.834**	**0.834/0.834**
Glass	0.520/0.520	0.520/0.520	0.520/0.520	**0.721/0.721**
Heart-c	0.702/**0.713**	0.702/**0.713**	0.702/**0.713**	**0.713**/0.675
Dermatology	0.732/0.729	0.732/0.729	0.732/0.729	**0.910/0.889**
Hepatitis	0.789/0.789	0.789/0.789	0.789/0.789	**0.944/0.944**
ThoraricSurgery	**0.765/0.765**	**0.765/0.765**	**0.765/0.765**	**0.765/0.765**
Lymph	0.651/**0.651**	0.651/**0.651**	0.651/**0.651**	**0.688**/0.640
Credit-g	0.684/**0.686**	0.684/**0.686**	0.684/**0.686**	**0.727**/0.681
Sonar	0.668/**0.689**	0.668/**0.689**	0.668/**0.689**	**0.942**/0.592
Ionosphere	0.872/**0.872**	0.872/**0.872**	0.872/**0.872**	**0.942**/0.851
Liver-Disorders	**0.617/0.617**	**0.617/0.617**	**0.617/0.617**	**0.617/0.617**
Vote	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**
Zoo	**0.754/0.754**	**0.754/0.754**	**0.754/0.754**	0.740/0.730

Acording to Table 4.5, when the numbers of selected features which provide the best F- measure values in hybrid method were used for feature selection process of CHI, IG and CFS techniques, the proposed hybrid method gave equal or better results on thirteen datasets while CHI, IG and CFS gave equal or better results on six datasets. At the same time, when the numbers of selected features which produce the worst F- measure values in hybrid method were used for feature selection process of CHI, IG and CFS techniques as well, the proposed hybrid method gave equal or better results on eight datasets while CHI, IG and CFS gave equal or better results on eleven datasets. Additionally, the average F-measure values obtained by the algorithms at the end of 10 runs are given in Table 4.6 to make a complete performance evaluation. When we performed feature selection processes of CHI, IG and CFS for a run, the number of feature to be selected by these algorithms was specified as equal to the number of feature selected at the end of this run of the proposed hybrid method.

Table 4.6. Average F-measure Values Achieved with CHI, IG, CFS and Hybrid Method at the end of 10 runs

Dataset	Average F-measure Values			
	CHI	IG	CFS	Hybrid Method
Autos	**0.658**	**0.658**	**0.658**	0.642
Breast-w	0.917	0.917	0.917	**0.922**
Car	**0.834**	**0.834**	**0.834**	**0.834**
Glass	0.520	0.520	0.520	**0.721**
Heart-c	**0.705**	**0.705**	**0.705**	0.700
Dermatology	0.678	0.678	0.678	**0.893**
Hepatitis	0.789	0.789	0.789	**0.944**
ThoraricSurgery	**0.765**	**0.765**	**0.765**	**0.765**
Lymph	0.649	0.649	0.649	**0.657**
Credit-g	0.686	0.686	0.686	**0.700**
Sonar	0.681	0.681	0.681	**0.781**
Ionosphere	0.865	0.865	0.865	**0.893**
Liver-Disorders	**0.617**	**0.617**	**0.617**	**0.617**
Vote	**0.964**	**0.964**	**0.964**	**0.964**
Zoo	**0.754**	**0.754**	**0.754**	0.737

According to Table 4.6, the proposed hybrid method has better or equal results for 12 datasets among 15 datasets. For Autos, Heart-c and Zoo datasets, the proposed hybrid method decreases average classification performance with the proportion of only 2.4%, 0.7% and 2.2% respectively.

In the above experiments, we employed J48 classifier in the test phase, since we used J48 classifier in the feature selection phase. However in the next experiment we applied other classifiers namely NaiveBayes and RBF Networks in the test phase to show the performance of our proposal.

NaiveBayes is one of the basic classification methods used for data mining and often performs well with its simple structure and ease of implementation. The other basic classification method often used for data mining is artifical neural networks. RBF Networks is especially popular in the pattern classification among artificial neural networks. So, we compared the results obtained by the proposed hybrid method, CHI, IG and CFS algorithms by using NaiveBayes and RBF Networks classifiers to make a complete performance evaluation. The classification performance for the best and worst set of selected features by the CHI, IG, CFS, and the proposed hybrid algorithms by using NaiveBayes classifier are shown in Table 4.7.

Table 4.7. Classification Performance using NaiveBayes Classifier for the Best/Worst Selected Features at the end of 10 runs

Dataset	Best/Worst F-measure Values with Selected Features			
	CHI	IG	CFS	Hybrid Method
Autos	**0.603/0.603**	**0.603/0.603**	**0.603/0.603**	0.435/0.435
Breast-w	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**
Car	**0.824/0.824**	**0.824/0.824**	**0.824/0.824**	**0.824/0.824**
Glass	0.444/0.444	0.444/0.444	0.444/0.444	**0.618/0.618**
Heart-c	**0.837/0.851**	**0.837/0.851**	**0.837/0.851**	0.783/0.797
Dermatology	0.732/0.740	0.732/0.740	0.732/0.740	**0.849/0.858**
Hepatitis	0.833/0.833	0.833/0.833	0.833/0.833	**0.944/0.944**
ThoraricSurgery	**0.757/0.757**	**0.757/0.757**	**0.757/0.757**	0.752/**0.757**
Lymph	**0.688/0.664**	**0.688/0.664**	**0.688/0.664**	0.605/0.628
Credit-g	**0.737/0.760**	**0.737/0.760**	**0.737/0.760**	0.714/0.724
Sonar	0.637/0.635	0.637/0.635	0.637/0.635	**0.713/0.668**
Ionosphere	0.801/0.801	0.801/0.801	0.801/0.801	**0.878/0.833**
Liver-Disorders	**0.544/0.544**	**0.544/0.544**	**0.544/0.544**	**0.544/0.544**
Vote	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**
Zoo	0.785/**0.767**	0.785/**0.767**	0.785/**0.767**	**0.845**/0.715

When we performed feature selection processes of CHI, IG and CFS algorithms for the best and worst F-measure values, the numbers of features to be selected were specified as equal to the numbers of selected features which provide the best and worst F-measure values in the proposed hybrid method respectively.

According to Table 4.7, the proposed hybrid method gave equal or better results on 10 datasets while CHI, IG and CFS gave equal or better results on 9 datasets in terms of the best F-measure values by using NaiveBayes classifier. Moreover the proposed hybrid methods, CHI, IG and CFS gave equal or better results on 10 datasets for the worst F-measure values by using NaiveBayes classifier. Also, average classification performance at the end of 10 runs with NaiveBayes classifier for these algorithms are given Table 4.8. In the construction of Table 4.8, to perform the feature selection processes of CHI, IG and CFS for a run, the number of feature to be selected by these algorithms was specified as equal to the number of feature selected at the end of this run of the proposed hybrid method, which is same as in the construction of Table 4.6.

Table 4.8. Average F-measure Values using NaiveBayes Classifier at the end of 10 runs

Dataset	Average F-measure Values with Selected Features			
	CHI	IG	CFS	Hybrid Method
Autos	**0.603**	**0.603**	**0.603**	0.435
Breast-w	**0.964**	**0.964**	**0.964**	**0.964**
Car	**0.824**	**0.824**	**0.824**	**0.824**
Glass	0.444	0.444	0.444	**0.618**
Heart-c	**0.841**	**0.841**	**0.841**	0.796
Dermatology	0.684	0.684	0.684	**0.863**
Hepatitis	0.833	0.833	0.833	**0.944**
ThoraricSurgery	**0.757**	**0.757**	**0.757**	0.754
Lymph	**0.664**	**0.664**	**0.664**	0.650
Credit-g	**0.757**	**0.757**	**0.757**	0.719
Sonar	0.645	0.645	0.645	**0.731**
Ionosphere	0.791	0.791	0.791	**0.861**
Liver-Disorders	**0.544**	**0.544**	**0.544**	**0.544**
Vote	**0.964**	**0.964**	**0.964**	**0.964**
Zoo	0.780	0.780	0.780	**0.806**

According to Table 4.8, the proposed hybrid method has better or equal results on 10 datasets while CHI, IG and CFS have better results or equal results on 9 datasets among 15 datasets. However, the proposed hybrid method decreases average classification performance with the proportion of only 0.39% and 2.1% respectively for ThoraricSurgery and Lymph datasets compared to CHI, IG and CFS. When we compared the results obtained by using J48 and NaiveBayes classifiers in terms of average F-measure values, the proposed hybrid method produce better or equal values on 12 datasets while CHI, IG and CFS produce better or equal values on 7 datasets using J48 classifier. Consequently, the results of a wrapper method are expected to be good only for the classifier used to evaluate feature subsets during the selection process but, the proposed hybrid method showed approximate performance with NaiveBayes as well. Although our hybrid method is a wrapper technique and selects features according to the J48 classifier, the selected features are also good for the Naive Bayes classifier.

The classification performance using RBF Networks with the best and worst selected set of features are presented in Table 4.9.

Table 4.9. Classification Performance using RBF Networks Classifier for the Best/Worst Selected Features at the end of 10 runs

Dataset	Best/Worst F-measure Values with Selected Features			
	CHI	IG	CFS	Hybrid Method
Autos	**0.631/0.631**	**0.631/0.631**	**0.631/0.631**	**0.631/0.631**
Breast-w	0.941/0.941	0.941/0.941	0.941/0.941	**0.976/0.976**
Car	**0.885/0.885**	**0.885/0.885**	**0.885/0.885**	**0.885/0.885**
Glass	0.572/0.572	0.572/0.572	0.572/0.572	**0.687/0.687**
Heart-c	**0.850/0.864**	**0.850/0.864**	**0.850/0.864**	0.756/0.797
Dermatology	0.780/0.769	0.780/0.769	0.780/0.769	**0.849/0.922**
Hepatitis	0.789/0.789	0.789/0.789	0.789/0.789	**0.944/0.944**
ThoraricSurgery	**0.765/0.796**	**0.765/0.796**	**0.765/0.796**	0.752/0.748
Lymph	0.592/0.575	0.592/0.575	0.592/0.575	**0.735/0.664**
Credit-g	0.692/0.735	0.692/0.735	0.692/0.735	**0.729/0.748**
Sonar	0.713/0.695	0.713/0.695	0.713/0.695	**0.730/0.764**
Ionosphere	0.849/0.849	0.849/0.849	0.849/0.849	**0.909/0.876**
Liver-Disorders	**0.556/0.556**	**0.556/0.556**	**0.556/0.556**	**0.556/0.556**
Vote	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**	**0.964/0.964**
Zoo	**0.813/0.821**	**0.813/0.821**	**0.813/0.821**	0.722/0.805

According to Table 4.9, the proposed hybrid methods gave equal or better results on 12 datasets while CHI, IG and CFS gave equal or better results on 7 datasets in terms of the best F-measure values by using RBF Networks classifier. At the same time, the proposed hybrid methods gave equal or better results on 12 datasets while CHI, IG and CFS gave equal or better results on 7 datasets for the worst F-measure values by using RBF Networks classifier. To evaluate the performances of the algorithms completely, average F- measure values for RBF Networks classifier at the end of 10 runs are given in Table 4.10.

Table 4.10. Average F-measure Values using RBF Networks Classifier at the end of 10 runs

Dataset	Average F-measure Values with Selected Features			
	CHI	IG	CFS	Hybrid Method
Autos	**0.631**	**0.631**	**0.631**	**0.631**
Breast-w	0.941	0.941	0.941	**0.976**
Car	**0.885**	**0.885**	**0.885**	**0.885**
Glass	0.572	0.572	0.572	**0.687**
Heart-c	**0.854**	**0.854**	**0.854**	0.766
Dermatology	0.718	0.718	0.718	**0.913**
Hepatitis	0.789	0.789	0.789	**0.944**
ThoraricSurgery	**0.774**	**0.774**	**0.774**	0.751
Lymph	0.574	0.574	0.574	**0.718**
Credit-g	0.729	0.729	0.729	**0.730**
Sonar	0.719	0.719	0.719	**0.755**
Ionosphere	0.864	0.864	0.864	**0.906**
Liver-Disorders	**0.556**	**0.556**	**0.556**	**0.556**
Vote	**0.964**	**0.964**	**0.964**	**0.964**
Zoo	**0.815**	**0.815**	**0.815**	0.747

When we analyze Table 4.10, the selected features using the proposed hybrid method also give satisfactory classification performance for RBF Networks classifier. As a result, the proposed hybrid method outperforms standart ABC and DE algorithms, CHI, IG and CFS by using J48 classifier and it also performs better than CHI, IG and CFS by using NaiveBayes and RBF Networks classifier. When we compared the classification performances of the classifiers for the proposed hybrid method, J48 is better than RBF networks and RBF Networks is better than NaiveBayes; but the classification performances of these classifiers are very close to each other.

One of the benefits of performing feature selection process is that it reduces the training time for learning the predictive model, and testing time for assigning class labels to new instances. To prove this, the average training times with average numbers of the selected features by the proposed hybrid method and the training times without making any feature selection for each dataset are compared and given in Figure 4.1 and the average testing times in testing phase are presented in Figure 4.2 as well.

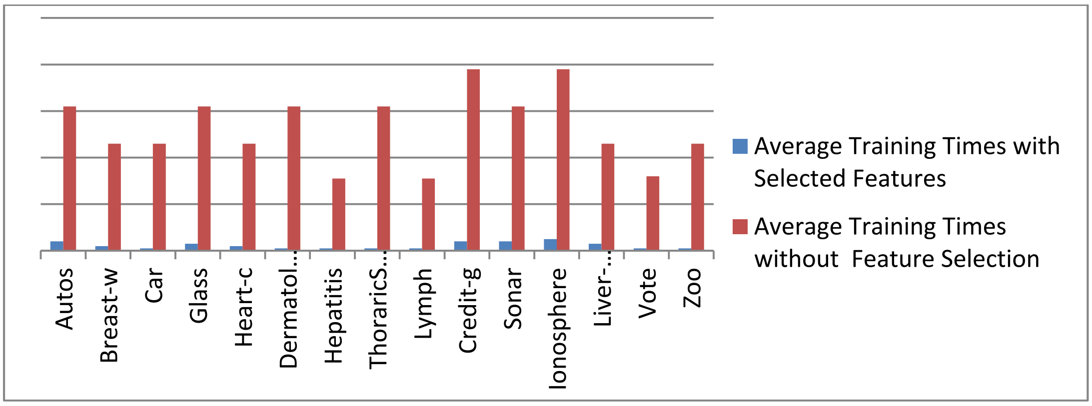


Figure 4.1. Average training times in seconds for the datasets

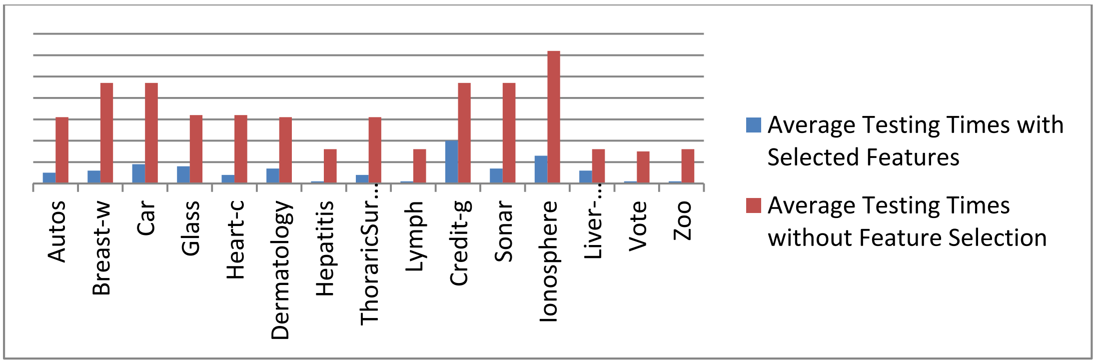


Figure 4.2. Average testing times in seconds for the datasets

As shown in Figure 4.1, Figure 4.2, Table 4.1, and Table 4.3 feature selection has reduced the training and testing times sharply while increasing the classification accuracy. When combining least two algorithms, the runtime of the new hybrid algorithm may increase. However, while we do this combining process, we try to unify the algorithms such that the hybrid method spends very little time to select features. In our study, we have taken account of this severe situation and so, we have tried to decrease the runtime of the algorithm by profitting from the modified onlooker bee process of ABC algorithm that spends quite little time and being applied of DE operators for only exploration process. The runtime of each algorithm is given in Table 4.11. As it is shown in Table 4.11, the differences of the runtimes of the algorithms are quite small.

Table 4.11. Average Runtimes of the Algorithms

Dataset	**Average Runtime (seconds)**		
	ABC	**DE**	**Hybrid Method**
Autos	109.6	112.7	162.2
Breast-w	124.5	94.4	142.5
Car	122.7	151.8	245.2
Glass	99.7	90.9	218.7
Heart-c	68.2	91.8	114.6
Dermatology	82.6	104.3	136.9
Hepatitis	22.7	30.9	34.5
ThoraricSurgery	66.7	108.8	152.9
Lymph	38.6	36.8	49.4
Credit-g	430.1	387.1	580.5
Sonar	239.8	347.1	387.3
Ionosphere	314.8	477.3	438.4
Liver-Disorders	76.2	80.7	143.9
Vote	22.4	26.1	35.7
Zoo	21.5	32.3	38.1

4.1. Comparison with Earlier Studies

In this section, the proposed hybrid method, the ABC and DE algorithms are compared with the earlier studies.

He et al. (2009) used some UCI datasets namely Breast-w, Vote and Zoo in their studies. In this study, they used DE algorithm to generate the feature subsets. They obtained 94.8%, 89% and 87.5% accuracy values for these datasets respectively while our proposed hybrid method obtained 97.6%, 96.4% and 78.3% accuracy values using RBF Networks classifier respectively. Considering these results, we can say that our proposed hybrid method outperforms He et.al (2009)'s studies.

UCI datasets were used in Palanisamy and Kanmani (2012)'s studies. In their studies, ABC was used to generate feature subsets and J48 classifier was used to evaluate each feature subset generated by ABC algorithm. They obtained 81.26% accuracy value for Hepatitis dataset with 11 selected features among total 19 features. Our proposed ABC algorithm obtained 94.7% accuracy value with 2 selected features among total 19 features. Again, they obtained 98.6% accuracy value for Dermatology dataset with 24 selected features among total of 34 features. Our developed ABC algorithm reached 93.4% accuracy value for Dermatology dataset with 13 selected features among total 34 features. When we compare the differences between accuracy values and the number of selected features, our proposed ABC algorithm decrases the accuracy value with the proportion of only 5.27%, but it reduces the number of selected features with the proportion of 45.8%. At the same time, they obtained 86.9% accuracy value for Heart-c dataset with 7 selected features among total of 13 features and our proposed ABC algorithm obtained 77% accuracy value with 3 selected features among total 13 attributes. Compared these differences between accuracy values and the number of selected features, our proposed ABC algorithm reduces the accuracy value with the proportion of only 11.3%, but it reduces the number of selected features with the proportion of 57.1%. Based on these results, we can say that our proposed ABC algorithm has good performance as least as Palanisamy and Kanmani (2012)'s method.

Schiezaro and Pedrini (2013) developed a feature selection method using ABC to classify some UCI datasets namely Autos, Breast-w, Glass, Heart-c, Hepatitis. In their studies, they obtained 82.9%, 75.8%, 71.5%, 83.1%, and 87.1% accuracy values with 9, 4, 6, 7, and 9 average number of selected features respectively. On the other hand, our proposed hybrid method obtained 65.7%, 92.3%, 75.9%, 70.2%, and 94.7% accuracy values with 9, 5, 4, 5, and 4 average number of selected features respectively. According to these results, our proposed hybrid method outperforms Schiezaro and Pedrini (2013)'s studies for 3 datasets among 5 datasets.

Martinoyić et al. (2014) proposed a new feature selection method which uses DE algorithm to generate feature subsets. In their studies, they obtained 95.1%, 94.4%, 75.9%, and 87.9% accuracy values for Breast-w, Dermatology, Glass and Ionosphere datasets with 5, 18, 5 and 11 selected features while our proposed hybrid method obtained 92.3%, 90.5%, 75.9, and 89.5% accuracy values with 5, 7, 4, and 9 selected features respectively. When we evaluate the accuracy values with the number of selected features for 4 datasets, our proposed hybrid method has worse performance for only Breast-w dataset. Based upon these results, we can say that our proposed hybrid method outperforms Martinoyić et al. (2014)'s studies.

5. CONCLUSION

In this thesis we have developed a hybrid wrapper method by combining ABC optimization technique with DE algorithm for feature selection problem on classification tasks. We have used fifteen UCI repository datasets which are commonly used by researchers for classification problems. In our system, the optimum feature subsets have been discovered by the hybrid feature selection method. In general, wrapper feature selection methods are specific to the classifier which is used during feature subset evaluation and it is expected that classification accuracy is high for only this classifier. In this study, the selected features are used to classify the datasets with different classification techniques such as NaiveBayes and RBF Networks by comparing with three most popular attribute selection methods. Experimental results show that our proposed hybrid method outperforms CHI, IG, CFS, standart ABC and DE algorithms even for other classifiers that were not used in the feature selection process.

As future work, we plan to apply parallelization to our hybrid feature selection method to reduce runtime of the algorithm for very large datasets which have more than 1000 features. Hence, we hope to diminish number of selected features in a realtively short period of time.

REFERENCES

ABDULLAH, A., DERIS, S., and MOHAMAD, M. S. (2013). A New Hybrid Bee Evolution Algorithm for Parameter Estimation in Biological Model. ICIC Express Letters, Part B, Applications: An International Journal of Research and Surveys, 4(1), p. 1-6.

ABRAHAM, A., JATOTH, R. K., and RAJASEKHAR, A. (2012). Hybrid Differential Artificial Bee Colony Algorithm. Journal of Computational and Theoretical Nanoscience, 9(2), p. 249-257.

AKILA, M. M. (2012). Hybrid Local Feature Selection in DNA Analysis based Cancer Classification. Indian Journal of Computer Science and Engineering, 3(3).

ALIZADEGAN, A., MEYBODI, M. R., and ASADY, B. (2012). A Novel Hybrid Artificial Bee Colony Algorithm And Differential Evolution for Unconstrained Optimization Problems. Advances in Computer Science and Engineering, 8(1).

BABATUNDE, R. S., OLABIYISI, S. O., OMIDIORA, E. O. and GANIYU, R. A. (2013). Feature Dimensionality Reduction using a Dual Level Metaheuristic Algorithm. International Journal of Applied Information Systems, 7, p. 49-52.

BASIRI, M. E. and NEMATI, S. (2009). A Novel ACO–GA Hybrid Algorithm for Text Feature Selection. Evolutionary Computation, IEEE Congress, p. 2561-2568.

BOLAJI, A. L. A., KHADER, A. T., AL-BETAR, M. A. and AWADALLAH, M. A. (2013). Artificial Bee Colony Algorithm, Its Variants and Applications: A Survey. Journal of Theoretical and Applied Information Technology, 47(2), p. 434-459.

CHIN T. W. (2007). Feature Selection for the Fuzzy Artmap Neural Network using a Hybrid Genetic Algorithm and Tabu Search. University Sains MALAYSIA. Master Thesis.

DENG, F., CHEN, J., WANG, Y. and GONG, K. (2013). Measurement and Calibration Method for an Optical Encoder based on Adaptive Differential Evolution-Fourier Neural Networks. Measurement Science and Technology, 24(5).

FROHLICH, H., CHAPELLE, O., and SCHOLKOPF, B. (2003). Feature Selection for Support Vector Machines by means of Genetic Algorithm. In Tools with Artificial Intelligence, 15th IEEE International Conference, p. 142-148.

GAO, W. and LIU, S. (2011). Improved Artificial Bee Colony Algorithm for Global Optimization. Information Processing Letters, 111(17), p. 871-882.

GRANDE, J., DEL ROSARIO SUÁREZ, M., and VILLAR, J. R. (2007). A Feature Selection Method using a Fuzzy Mutual Information Measure. In Innovations in Hybrid Intelligent Systems, Springer, p, 56-63.

HASANI, S. R., OTHMAN, Z. A. and KAHAKI, S. M. M (2014). Hybrid Feature Selection Algorithm For Intrusion Detection System. Journal of Computer Science 10 (6), p. 1015-1025.

HE, X., ZHANG, Q., SUN, N. and DONG, Y. (2009). Feature Selection with Discrete Binary Differential Evolution. In Artificial Intelligence and Computational Intelligence, AICI'09 International Conference, 4, p. 327 330.

JONA, J. B. and NAGAVENI, N. (2012). A Hybrid Swarm Optimization Approach for Feature Set Reduction in Digital Mammograms. WSEAS Transactions on Information Science and Applications, 9, p. 340-349.

JONA, J. B. and NAGAVENI, N. (2014). Ant-cuckoo Colony Optimization Approach for Feature Selection in Digital Mammogram. Pakistan Journal of Biological Sciences, 17(2), p. 266-271.

KARABOGA, D. (2005). An Idea based on Honey Bee Swarm for Numerical Optimization. Erciyes university, engineering faculty, computer engineering department. Technical report.

KHUSHABA, R. N., AL-ANI, A. and AL-JUMAILY, A. (2008). Differential Evolution based Feature Subset Selection. In Pattern Recognition, ICPR 19th International Conference, p.1-4.

LI, X. and YIN, M. (2012). Hybrid Differential Evolution with Artificial Bee Colony and Its Application for Design of a Reconfigurable Antenna Array with Discrete Phase Shifters. IET Microwaves, Antennas and Propagation, 6(14), p. 1573-1582.

MARTİNOYİĆ, G., BAJER, D. and ZORİĆ, B. (2014). A Differential Evolution Approach to Dimensionality Reduction for Classification Needs. International Journal of Applied Mathematics and Computer Science, 24(1), p. 111-122.

MOHAMED, A. W., SABRY, H. Z. and KHORSHİD, M. (2012). An Alternative Differential Evolution Algorithm for Global Optimization. Journal of Advanced Research, 3(2), p. 149-165.

MONTGOMERY, J. and CHEN, S. (2010). An Analysis of the operation of Differential Evolution at High and Low Crossover Rates. In Evolutionary Computation (CEC), IEEE Congress, p. 1-8.

OH, I. S., LEE, J. S. and MOON, B. R. (2004). Hybrid Genetic Algorithms for Feature Selection. Pattern Analysis and Machine Intelligence, IEEE Transactions, 26(11), p. 1424-1437.

OLABIYISI S. O., FAGBOLA T. M., OMIDIORA E. O. and OYELEYE A. C. (2012). Hybrid Metaheuristic Feature Extraction Technique for Solving Timetabling Problem. International Journal of Scientific and Engineering Research, 3(8), p. 1-6.Optimizing Feature Selection. International Journal of Computer Science Issues, 9(3), p. 432-438.

PALANISAMY, S. and KANMANI, S. (2012). Artificial Bee Colony Approach for Optimizing Feature Selection. International Journal of Science Issues, 9(3), p. 432-438.

PRASARTVIT, T., BANHARNSAKUN, A., KAEWKAMNERDPONG, B. and ACHALAKUL, T. (2013). Reducing Bioinformatics Data Dimension with ABC-kNN. Neurocomputing, 116, p. 367-381.

QUINLAN, J. R. (1993). C4.5: Programs for Machine Learning. First Edition. Morgan Kaufmann Publishers. San Mateo, California. 302 p.

QUINLAN, J. R. (1997). Decision Trees and Instance-Based Classifiers. The Computer Science and Engineering Handbook, p. 521-535.

SÁ, Â. A., ANDRADE, A. O. and SOARES, A. B. (2008). Exploration vs. Exploitation in Differential Evolution. In AISB 2008 Convention Communication, Interaction and Social Intelligence, 1, p. 57-63.

SAEYS, Y., INZA, I. and LARRAÑAGA, P. (2007). A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics, 23(19), p. 2507-2517.

SARAC, E. and OZEL, S.A. (2010). URL-Based Web page Classification. ASYU Symposium, p. 13-17.

SCHIEZARO, M. and PEDRINI, H. (2013). Data Feature Selection based on Artificial Bee Colony Algorithm. EURASIP Journal on Image and Video Processing, 1, p. 1-8.

SHANG, W., HUANG, H., ZHU, H., LIN, Y., QU, Y. and WANG, Z. (2007). A Novel Feature Selection Algorithm for Text Categorization. Expert Systems with Applications, 33(1), p. 1-5.

SHANTHI, S. and BHASKARAN, V. M. (2014). Modified Artificial Bee Colony based Feature Selection: A New Method in the Application of Mammogram Image Classification. International Journal of Science, Engineering and Technology Research (IJSETR), 3(6), p. 1664-1667.

STORN, R. and PRICE, K. (1997). Differential Evolution–A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 11(4), p. 341-359.

UCI, University of California Irvine Machine Learning Repository: http://archive.ics.uci.edu/ml/

UZER, M. S., YILMAZ, N. and INAN, O. (2013). Feature Selection Method based on Artificial Bee Colony Algorithm and Support Vector Machines for Medical Datasets Classification. The Scientific World Journal, p. 1-10.

VAN RIJSBERGEN, C. J., 1979. Information Retrieval. Second Edition. Butterworth-Heinemann Publishers. London, UK. 224 p.

WANG, Z., OUYANG, R. and KONG, X. (2013). A Hybrid Artificial Bee Colony Algorithm for Portfolio Optimization Problem 1. Journal of Theoritical and Applied Information Technology, 49(1).

WEKA, Data Mining Software in Java. Available at: http://www.cs.waikato.ac.nz/~ml/weka/

WU, Y. C., LEE, W. P. And CHIEN, C. W. (2011). Modified the Performance of Differential Evolution Algorithm with Dual Evolution Strategy. International Conference on Machine Learning and Computing, 3, p. 57-63.

XU, Y., FAN, P. and YUAN, L. (2013). A Simple and Efficient Artificial Bee Colony Algorithm. Mathematical Problems in Engineering, p. 1-9.

YUSOFF, S. A. M., ABDULLAH, R. And VENKAT, I. (2014). Adapted Bio-inspired Artificial Bee Colony and Differential Evolution for Feature Selection in Biomarker Discovery Analysis. In Recent Advances on Soft Computing and Data Mining, p. 111-120.

ZHANG, Y. and WU, L. (2011). Optimal Multi-level Thresholding based on Maximum Tsallis Entropy via an Artificial Bee Colony Approach. Entropy, 13(4), p. 841-859.

CURRICULUM VITAE

Ezgi Zorarpacı was born in Ankara, in 1986. She has completed her elementary education in Gaziantep. She went to high school at Gaziantep Ayten Kemal Akınal Anatolian High School. She has completed university education at department of Computer Engineering of Selçuk University in 2011. Since 2012, she has been working as a research assistant at Computer Engineering Department of Çukurova University in Adana.