ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Ferda SUNA DÖKME

APPLICATION OF PARTICLE SWARM OPTIMIZATION FOR COMPUTER AIDED DIAGNOSIS OF DISEASES

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2019

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

APPLICATION OF PARTICLE SWARM OPTIMIZATION FOR COMPUTER AIDED DIAGNOSIS OF DISEASES

Ferda SUNA DÖKME

MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was received and approved the award of degree of the Master of Science by the board of jury on

……	……	……....
Prof. Dr. Selma Ayşe ÖZEL	Asst. Prof. Dr. Serkan KARTAL	Asst. Prof. Dr. Esra SARAÇ EŞSİZ
SUPERVISOR	MEMBER	MEMBER

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

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

ABSTRACT

MSc THESIS

APPLICATION OF PARTICLE SWARM OPTIMIZATION FOR COMPUTER AIDED DIAGNOSIS OF DISEASES

Ferda SUNA DÖKME

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

Supervisor : Prof. Dr. Selma Ayşe ÖZEL
Year: 2019, Pages: 88
Juries : Prof. Dr. Selma Ayşe ÖZEL
: Asst. Prof. Dr. Serkan KARTAL
: Asst. Prof. Dr. Esra SARAÇ EŞSİZ

Data mining is used in order to obtain meaningful information from the data obtained in many different fields by applying several methods. Data mining is widely used to analyze medical data to make diagnosis of several diseases as this is very important topic and there exists large amount of available data in medical domain. In this study, Particle Swarm Optimization (PSO) is used to reduce the size of the medical data by making feature selection to perform better data analysis from healthcare datasets. To reach our goal, Breast Cancer Coimbra, Diabetic Retinopathy Debrecen, Self-Care Activities, and Lee Silverman Voice Treatment datasets that are used to diagnose breast cancer, diabetic retinopathy, children's self-care problems, speech disorders of patients having Parkinson disease, respectively, obtained from UCI Machine Learning Repository, are analyzed by using Naïve Bayes (NB), Support Vector Machines (SVM), and Random Forests (RF) classifiers. The experimental analysis has shown that the PSO based feature selection improves the classification accuracy of diagnosis of diseases for the NB and RF classifiers. The PSO based method is also compared with the well-known feature selectors that are information gain (IG), chi-square (CHI2) and Relief. It is observed that PSO based method has better performance than that of IG and CHI2 methods, and similar results with the Relief method.

Keywords: Data Mining, Particle Swarm Optimization, Classification, Feature Selection, Medical Data

ÖZ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR DESTEKLİ HASTALIK TEŞHİSİ İÇİN PARÇAÇIK SÜRÜ OPTİMİZASYONU TEKNİĞİNİN UYGULANMASI

Ferda SUNA DÖKME

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Prof. Dr. Selma Ayşe ÖZEL
Yıl: 2019, Sayfa: 88
Jüri : Prof. Dr. Selma Ayşe ÖZEL
: Dr. Öğretim Üyesi Serkan KARTAL
: Dr. Öğretim Üyesi Esra SARAÇ EŞSİZ

Veri madenciliği farklı yöntemler kullanarak, birçok alanda elde edilen verilerden anlamlı bilgi elde etmek için kullanılır. Veri madenciliği, çok önemli bir konu olduğundan ve tıbbi alanda büyük miktarda mevcut veri bulunduğundan, çeşitli hastalıkların tanısını koymak için tıbbi verileri analiz etmek amacıyla yaygın olarak kullanılmaktadır. Bu çalışmada, Parçacık Sürüsü Optimizasyonu (PSO), sağlık ile ilgili veri kümelerinden daha iyi veri analizi yapmak için öznitelik seçimi yaparak tıbbi verilerin boyutunu azaltmak için kullanılmıştır. Hedefimize ulaşmak için UCI makine öğrenim deposundan elde edilen Breast Cancer Coimbra, Diabetic Retinopathy Debrecen, Self-Care Activities ve Lee Silverman Voice Treatment verikümeleri kullanılarak sırasıyla meme kanserini, diyabetik retinopatiyi, çocukların öz bakım problemlerini, parkinson hastalığı olan hastaların konuşma bozukluklarını teşhis etmek için Naif Bayes (NB), Destek Vektör Makineleri (SVM) ve Rastgele Ormanlar (RF) sınıflandırıcıları ile analiz edilmiştir. Deneysel analiz, PSO tabanlı öznitelik seçiminin, NB ve RF sınıflandırıcıları için hastalık tanısının sınıflandırma doğruluğunu arttırdığını göstermiştir. PSO tabanlı yöntem, bilgi kazancı (IG), Ki-kare (CHI2) ve Relief olarak iyi bilinen öznitellik seçicileriyle de karşılaştırılmıştır. PSO tabanlı yöntemin IG ve CHI2 yöntemlerinden daha iyi bir performansa sahip olduğu ve Relief yöntemiyle benzer sonuçları olduğu gözlenmiştir.

Anahtar kelimeler: Veri Madenciliği, Parçaçık Sürü Optimizasyonu, Sınıflama, Nitelik Seçimi, Tıbbi veriler

EXTENDED ABSTRACT

Data mining aims to extract useful information from large-scale data. Various methods have been developed to extract meaningful information from these large-scale data. These methods are from many different disciplines such as statistics, database systems, machine learning, and artificial intelligence. Data mining is used in many different areas such as finance, telecommunication, energy, market basket analysis, education, lie detection, research analysis, criminal investigation, bio informatics, healthcare, etc. Today, as in many areas, data in the field of medicine and health are stored in digital environment and it is easily accessible. Applying data analysis tasks such as data mining for healthcare and medicine are the most important scientific research areas nowadays because of the size and the availability of the data in this domain, and the importance of healthcare related studies. The purpose of health information systems is to produce useful information from large amount of health-related data.

Classification (supervised learning) and optimization techniques are frequently used together in data mining to extract knowledge from datasets. The classification algorithms are applied on the data having the class label to learn a model which is then used for assigning class labels to new data instances. There are many classification algorithms. Naïve Bayes (NB), Support Vector Machine (SVM), Decision trees (DT), K nearest neighbors (KNN) are the most widely used classification techniques. The optimization techniques used with classification task in the literature include Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), etc. such that they are generally used for reducing the dimensionality of the datasets to make better classification, optimizing the parameters of the classifiers, or developing a new classifier.

The aim of this thesis is to apply PSO optimization technique to select optimal feature subset to classify medical datasets more efficiently. To reach our

goal, four datasets that are Breast Cancer Coimbra (BCCD), Diabetic Retinopathy Debrecen (DRDD), Self Care Activities (SCADI), and Lee Silverman Voice Treatment (LSVT) from UCI machine learning repository are used. These datasets are used to make diagnosis of breast cancer, diabetic retinopathy, children's self-care problems, and speech disorders of Parkinson patients. In the literature BCCD, DRDD, SCADI and LSVT datasets have been used for classification, however this thesis is the first study that applies PSO based feature selection to improve classification accuracy.

These datasets have varying feature and instance sizes such that BCCD has 10 attributes and has 116 instances; DRDD has 20 attributes and has 1151 instances, SCADI has 206 attributes and 70 samples; and LSVT has 309 attributes and 126 samples. Each of these datasets belong to identification of different diseases and they are recently donated datasets that have not been used with PSO based feature selectors before. Each instance in these datasets have a class label and the datasets are suitable for use in supervised learning such as classification.

In this thesis, Particle Swarm Optimization (PSO) is used for feature selection; and supervised learning methods that are Naïve Bayes (NB), Support Vector Machines (SVM), and Random Forest (RF) are employed for classification task to determine whether the specific disease exists or not in the patient. First, the datasets are divided into train and test sets. All datasets used in this study are divided as 70% train, and 30% test datasets. Then, classification accuracy is calculated for the test set by applying the classification model that is learned from the training set by using the above-mentioned classification methods. These results form our baseline results. After that, PSO is applied to the training set of each dataset to get the best feature subsets. For each dataset and classifier, the classification performance is measured by reducing the size of the training and test sets by using the selected features and the results are compared with the baseline results. The aim is to increase the classification accuracy by applying the feature selection.

To implement PSO based feature selector, a classifier is used to measure the fitness of the selected features. For this purpose, NB classifier is used. In this study the PSO optimization algorithm and the applied NB classifier are implemented in Python 3.7 using the Numpy package and the scikit-learn library.

After the feature selection process, performance of disease diagnosis is measured by applying NB, SVM, and RF classifiers to the reduced datasets that contain only the selected features. In this step, WEKA implementations of the NB, SVM and RF classifiers are used. To make experimental evaluation, the used hardware has the following features: Windows 10 (64 bit) operating system, Intel i7 Core processor 3.60Ghz, 16.0GB RAM.

The performance of the PSO based feature selector is also compared with the well-known feature selectors that are information gain (IG), chi-square (CHI2), and Relief. According to the experimental evaluation, it is observed that the success of the classification accuracy increases with applying feature selection. 80.5% accuracy with NB and RF classifier for BCCD dataset, 69.1% accuracy with RF classifier for DRDD dataset, 83.3% accuracy with NB classifier for SCADI dataset, and 87.2% accuracy with RF classifier for LSVT dataset are obtained after applying PSO based feature selection. On the other hands, the baseline results for these datasets are 63.8% and 77.7% accuracies with NB and RF classifiers for BCCD dataset, 65% accuracy with RF classifier for DRDD dataset, 79.1% accuracy with NB classifier for SCADI dataset, and 82% accuracy with RF classifier for LSVT dataset. Therefore, it is observed that applying the PSO based feature selector improves classification accuracy for all datasets.

When PSO based method is compared with IG, CHI2, and Relief feature selection methods, it is observed that the PSO based method has better performance than that of IG and CHI2 methods; and have similar performance with Relief method. All feature selectors used in this thesis improve classification accuracy for all datasets.

GENİŞLETİLMİŞ ÖZET

Veri madenciliği, büyük ölçekli verilerden faydalı bilgiler elde etmeyi amaçlar. Bu büyük ölçekli verilerden anlamlı bilgiler elde etmek için çeşitli yöntemler geliştirilmiştir. Bu yöntemler istatistik, veritabanı sistemleri, makine öğrenmesi ve yapay zeka gibi birçok farklı disiplindendir. Veri madenciliği finans, telekomünikasyon, enerji, market sepeti analizi, eğitim, yalan tespiti, araştırma analizi, ceza soruşturması, biyo bilişim, sağlık, vb. gibi birçok farklı alanda kullanılmaktadır. Günümüzde birçok alanda olduğu gibi tıp ve sağlık alanındaki veriler dijital ortamda depolanmakta ve kolaylıkla erişilebilir durumdadır. Sağlık ve tıp için veri madenciliği gibi veri analizi görevlerini uygulamak, bu alandaki verilerin büyüklüğü ve kullanılabilirliği ve sağlıkla ilgili çalışmaların önemi nedeniyle günümüzde en önemli bilimsel araştırma alanlarından biri haline gelmiştir. Sağlık bilgi sistemlerinin amacı, büyük miktarda sağlıkla ilgili verilerden faydalı bilgiler üretmektir.

Veri kümelerinden bilgi çıkarmak için sınıflandırma (gözetimli öğrenme) ve optimizasyon teknikleri sıklıkla veri madenciliğinde birlikte kullanılır. Sınıflandırma algoritmaları, yeni verilere daha sonra sınıf etiketleri atamak için kullanılan bir modeli öğrenmek için sınıf etiketine sahip veriler üzerinde uygulanır. Birçok sınıflandırma algoritması vardır. Naif Bayes (NB), Destek Vektör Makinesi (SVM), Karar ağaçları (DT), k-en yakın komşu (KNN) en yaygın kullanılan sınıflandırma teknikleridir. Literatürde sınıflandırma görevi ile kullanılan optimizasyon teknikleri arasında, daha iyi bir sınıflandırma yapmak için veri setlerinin boyutluluğunu azaltmak, sınıflandırıcı parametrelerini optimize etmek veya yeni bir sınıflandırıcı geliştirmek amacıyla kullanılan, Parçacık Sürüsü Optimizasyonu (PSO), Karınca Kolonisi Optimizasyonu (ACO), Arı Kolonisi Optimizasyonu (BCO) vb. bulunur.

Bu tezin amacı, tıbbi veri kümelerini daha verimli bir şekilde sınıflandırmak için optimal öznitelik alt kümesini seçmek üzere PSO optimizasyon

tekniğini uygulamaktır. Hedefimize ulaşmak için, Meme Kanseri Coimbra (BCCD), Diyabetik Retinopati Debrecen (DRDD), Öz Bakım Etkinlikleri (SCADI) ve Lee Silverman Ses Tedavisi (LSVT) olmak üzere UCI makine öğrenim deposundan dört veri seti kullanılmıştır. Bu veri setleri, meme kanseri, diyabetik retinopati, çocukların öz bakım problemleri ve Parkinson hastalarının konuşma bozukluklarını teşhis etmek için kullanılmaktadır. Literatürde BCCD, DRDD, SCADI ve LSVT veri setleri sınıflandırma için kullanılmıştır, ancak bu tez, sınıflandırma doğruluğunu geliştirmek için PSO tabanlı öznitelik seçimini uygulayan ilk çalışmadır.

Bu veri setleri çeşitli özelliklere ve örnek boyutlarına sahiptir, örneğin, BCCD 10 özelliğe ve 116 örneğe, DRDD 20 özelliğe ve 1151 örneğe, SCADI 206 özelliğe ve 70 örneğe ve LSVT 309 özelliğe ve 126 örneğe sahiptir. Bu veri kümelerinin her biri farklı hastalıkların tanımlanmasına aittir ve daha önce PSO tabanlı özellik seçicileriyle kullanılmamış olup, yakın zamanda veri kümesi olarak bağışlanmıştır. Bu veri kümelerindeki her örnek bir sınıf etiketine sahiptir ve sınıflandırma gibi gözetimli öğrenmede kullanım için uygundur.

Bu tezde, Parçacık Sürü Optimizasyonu (PSO) özellik seçimi için kullanılmıştır ve sınıflama görevi için spesifik hastalıkların var olup olmadığını belirlemek amacıyla Naif Bayes (NB), Destek Vektör Makineleri (SVM) ve Rasgele Orman (RF) gözetimli öğrenme yöntemleri uygulanmıştır. İlk olarak, veri kümeleri eğitim ve test veri kümelerine ayrılmıştır. Bu çalışmada kullanılan tüm veri kümeleri %70 eğitim ve %30 test veri kümeleri olarak ayrılmıştır. Daha sonra, yukarıda belirtilen sınıflandırma yöntemlerini kullanarak eğitim kümesinden öğrenilen sınıflandırma modelini uygulayarak test kümesi için sınıflandırma doğruluğu hesaplanmıştır. Bu sonuçlar bizim temel sonuçlarımızı oluşturmaktadır. Bundan sonra, en iyi özellik alt kümelerini elde etmek için PSO her veri kümesinin eğitim kümesine uygulanır. Her veri kümesi ve sınıflandırıcı için, sınıflandırma performansı, seçilen özellikleri kullanarak eğitim ve test kümelerinin büyüklüğü azaltılarak ölçülür ve sonuçlar, temel sonuçlarla karşılaştırılır. Amaç, özellik

seçimini uygulayarak sınıflandırma doğruluğunu arttırmaktır.

PSO tabanlı özellik seçiciyi uygulamak için, seçilen özelliklerin uygunluğunu ölçmek amacıyla bir sınıflandırıcı kullanılır. Bu amaçla, NB sınıflandırıcı kullanılmıştır. Bu çalışmada, PSO optimizasyon algoritması ve uygulanan NB sınıflandırıcısı, Numpy paketi ve scikit-learn kütüphanesi kullanılarak Python 3.7'de uygulanmıştır.

Özellik seçim işleminden sonra, yalnızca seçilen özellikleri içeren azaltılmış veri kümelerine NB, SVM ve RF sınıflandırıcıları uygulanarak hastalık tanı performansı ölçülmüştür. Bu adımda WEKA'da tanımlanmış olan NB, SVM ve RF sınıflandırıcıları kullanılmıştır. Deneysel değerlendirme yapmak için, kullanılan donanım şu özelliklere sahiptir: Windows 10 (64 bit) işletim sistemi, Intel i7 Core işlemci 3.60Ghz, 16.0GB RAM.

PSO tabanlı özellik seçicinin performansı, literatürde sıklıkla kullanılan bilgi kazancı (IG), ki-kare (CHI2) ve Relief özellik seçicileriyle de karşılaştırılmıştır. Deneysel değerlendirmeye göre, sınıflandırma doğruluğunun başarısının özellik seçimi uygulanarak arttığı görülmektedir. PSO tabanlı özellik seçimi uygulandıktan sonra BCCD veri kümesi için NB ve RF sınıflandırıcısı ile %80,5 doğruluk, DRDD veri kümesi için RF sınıflandırıcısı ile %69,1, SCADI veri kümesi için NB sınıflandırıcı ile %83,3 doğruluk ve LSVT veri seti için RF sınıflandırıcı ile %87,2 doğruluk elde edilmiştir. Öte yandan, bu veri setleri için temel sonuçlar, BCCD veri kümesi için NB ve RF sınıflandırıcıları ile %63,8 ve %77,7 doğruluk, DRDD veri kümesi için RF sınıflandırıcı ile %65, SCADI veri kümesi için NB sınıflandırıcı ile %79,1 ve LSVT veri kümesi için RF sınıflandırıcı ile %87,2 doğruluk değerleridir. Bu nedenle, PSO tabanlı özellik seçicinin uygulanmasının tüm veri kümeleri için sınıflandırma doğruluğunu arttırdığı gözlenmektedir.

PSO tabanlı yöntem IG, CHI2 ve Relief özellik seçim yöntemleri ile karşılaştırıldığında PSO tabanlı yöntemin IG ve CHI2 yöntemlerinden daha iyi performansa ve Relief yöntemiyle benzer performansa sahip olduğu gözlenmiştir.

Bu tezde kullanılan tüm özellik seçme yöntemleri, tüm veri kümeleri için sınıflandırma doğruluğunu artırmıştır.

ACKNOWLEDGEMENT

I would like to express my gratitude thank to my thesis advisor Professor Dr. Selma Ayşe Özel for her deepest care, endless support and encouragement at every stage of my thesis. When I fall into despair, her ideas lightened my way and given me hope. Thank you for the thesis jury suggestions and corrections. Thank you in all sincerely.

I have always felt their precious, kindly supports and love at my side. I would like to thank my family who always supported me throughout my education life and my husband Fatih Dökme, my son Yiğit Alaz Dökme and my daughter Cennet Lorin Dökme.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURE **PAGE**

LIST OF SYMBOLS AND ABREVIATIONS

UCI	: University of California Irvin
PSO	: Particle Swarm Optimization
BCCD	: Breast Cancer Coimbra Dataset
DRDD	: Diabetic Retinopathy Debrecen Dataset
AUC	: Area Under Curve
LR	: Logistic Regression
SVM	: Support Vector Machine
RF	: Random Forest
WBCD	: Wisconsin Breast Cancer Dataset
WDBC	: Wisconsin Diagnosis Breast Cancer
WPBC	: Wisconsin Prognosis Breast Cancer
DT	: Decision Tree
NN	: Neural Network
ANN	: Artificial Neural Network
ELM	: Extreme Learning Machine
K-NN	: K-Nearest Neighbor
IBK	: K-Nearest Neighbor
CKDD	: Chronic Kidney Disease Dataset
TSD	: Thoracic Surgery Dataset
WEKA	: Waikato Environment for Knowledge Analysis
NB	: Naive Bayes
SL	: Simple Logistic
IMC	: Input Mapped Classifier
NNET	: Neural Network
QDA	: Quadratic Discriminant Analysis
MLP	: Multi Layer Perceptron
PIDD	: Pima Indian Diabetes Dataset

PCA	: Principal Component Analysis
GA	: Genetic Algorithm
WHO	: World Health Organization
ROI	: Region On Interest
RAM	: Random Access Memory
CPU	: Central Process Unit
GNU	: General Public License
ARFF	: Attribute Relation File Format
BPSO	: Binary Particle Swarm Optimization
CPSO	: Continuous Particle Swarm Optimization
CSV	: Comma Separated Values
FS	: Feature Selection
RT	: Random Tree
S	: Second
AB	: Adaptive Boosting

1. INTRODUCTION

Data mining is the process of discovering interesting information, such as relationships, patterns, changes, deviations and trends, specific structures that are found among the data in very large databases. It is used to create meaningful data from complex and large datasets (Khan et al., 2013). Data mining is an intermediate system between the dataset and an application that converts data into available information (Yao, 2003). Traditionally, data mining was performed manually. As the time has passed the size of the data grew and it became impossible to process. Many methods are used to extract meaningful data from existing data in data mining. These include techniques from machine learning, artificial intelligence, statistics, databases, etc.

Today, data mining is used in many areas such as science and engineering, banking and finance, telecommunication and manufacturing, customer relations management, detection of fraud, security and intelligence, in the field of education, health and biomedical. Health and medicine are the most important scientific research areas of data mining. Data mining in healthcare are being used mainly for predicting various diseases as well as in assisting for diagnosis for the doctors in making their clinical decision (Jothi et al., 2015).

Feature selection in medical data is important for disease diagnosis. The feature selection is defined as the selection of the best subset representing the original dataset. Many studies have been done in this area and various techniques have been used for feature selection. Some of these techniques are filter based methods, statistical methods, machine learning based methods, optimization algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony Optimization (ABC), etc. based methods. Optimization algorithms have been widely used in the literature for feature selection.

Optimization is the most efficient use of the available resources in a system. Mathematically, optimization can be defined as minimizing or maximizing a function. Optimization methods and statistical methods are often used when selecting features in medical data. Using these techniques, the effect of the attributes in the dataset on classification accuracy is measured and thus, the most important features causing the disease are identified.

In this study, it is aimed to improve the classification performance by selecting the best feature subset with an optimization algorithm. The optimization algorithm used is PSO. The reason PSO is preferred is that it does not involve complex mathematical equations, it is easy to compute, and it is an algorithm that is used frequently in optimization problems and gives successful results (Bai, 2010).

The UCI machine learning repository is used as a source of dataset. UCI repository contains many disease related datasets. Breast cancer, diabetes, childhood diseases and Parkinson diseases datasets from the UCI repository were used in this thesis to show the effect of feature selection with PSO. The datasets used in this study are collected to identify some diseases by using machine learning techniques. These diseases are listed as follows: *i*) breast cancer is a common type of cancer that is seen mostly in women and sometimes causes fatal consequences; *ii*) diabetic retinopathy is a disease caused by damage to the eyes due to diabetes; *iii*) self-care problems are disorders that can develop in childhood; and *iv*) parkinson's disease (PD) is a neurodegenerative condition, an illness that affects nerve cells in the brain that control movement. All these datasets are evaluated by using the Particle Swarm Optimization (PSO) and a system is developed with high accuracy by feature selection done with PSO. The system is initialized with a population of random solutions and then searches for optima are performed by updating generations. Thus, accuracy of each generation is computed and the attributes that give the best accuracy at the end of the iterations are chosen as the best feature subset of the dataset. The size of the dataset is reduced by using only

the selected feature subset, then the classification step is applied to get better diagnosis of diseases.

In previous studies, various data mining methods have been used for disease diagnosis. For example, Akay (2009) used breast cancer dataset from UCI, evaluate this dataset using support vector machines and has reached the highest classification accuracy. Tokan, Türker and Yıldırım (2005) used echocardiogram dataset and applied artificial neural networks, and ROC analysis for all network structures. After that, they calculated the accuracy of their method by computing the sensitivity and the specificity values of the results. (Keleş and Keleş, 2008) used the thyroid disease data which was taken from the UCI machine learning repository to diagnose thyroid disease by using fuzzy systems. However, feature selection was made with PSO using different healthcare data. For example, Chen et al. (2012) studied eight different disease dataset from the UCI and applied the PSO + 1-NN method for feature selection.

In the literature, the datasets used in this study have been used with various data mining techniques, but Particle Swarm Optimization (PSO) based feature selection has not been used with these datasets. Therefore, using these datasets in this study, the existing classification success is tried to be increased by applying feature selection using Particle Swarm Optimization (PSO) optimization technique. Therefore, the contribution of this study is to show the effect of PSO based feature selection on the diagnosis of breast cancer, diabetes, childhood diseases, and Parkinson diseases.

The next sections are materials and methods, results and discussion and conclusion and future work. In the material and method section, the datasets and the methods used for the feature selection are explained. In the results and discussion section, the results of the feature selection using PSO and their comparisons with previous studies are explained. In the last section, the conclusion and future studies are explained.

2. RELATED WORKS

In this section previous studies using Breast Cancer Coimbra (BCCD), Diabetic Retinopathy Debrecen Dataset (DRDD), Self Care Activities (SCADI) and Lee Silverman Voice Treatment (LSVT) are summarized. In addition, studies that include feature selection for the similar datasets are examined.

Feature selection methods have important role in data mining. Feature selection helps you to select the minimum number of features from the whole feature set to reduce computing time, space needed, and improve performance of data analysis task (Sheena et al., 2016). Various methods are used for feature selection and examples from previous studies are given in this section of the thesis.

Pereira et al. (2018) have studied BCCD data and predicted the effects of Resistin, Glucose, Age and BMI on breast cancer. BCCD data has 116 samples and 10 features such as Glucose, Resistin, Age, BMI, HOMA, Leptin, Insulin, Adiponectin, MCP-1. These 116 sample are divided into two classes as, healthy controls and patient ones. 64 breast cancer patients and 52 healthy volunteers are included in the study. Using the dataset, two different analyzes are performed: Univariate analysis and Multivariate analysis. In Univariate analysis, ROC analysis for each property (Age, BMI, Glucose, Insulin, HOMA, Leptin, Adiponectin, Resistin and MCP-1) is made and confidence intervals of 95% is found for the respective Area Under Curve (AUC) values. Glucose, Insulin, HOMA and Resistin have AUC values that are greater than 0.5. Then, the sensitivity and specificity of the properties with AUC values greater than 0.5 are calculated. As a result, glucose with the higher sensitivity 77% and HOMA with the highest specificity 85% are found. In Multivariate analysis, dataset is divided into training and test datasets. The training set has 69.8% of total data (45 out of 62 breast cancer patients and 36 out of 52 healthy instances). Tree different classification methods are used on the training set: Logistic Regression (LR), Support Vector Machines (SVM) and Random Forests (RF). Using different attributes predictors are tested and

confidence intervals at a 95% level of confidence are computed in the test set for the AUC. The sensitivity and specificity values of the models are calculated. Best sensitivity combination - 95% CI [82.2%, 87.5%] - and specificity - 95% CI [84.5%, 89.7%] are found. Best values found with SVM. The attributes that yield this best result are: Glucose, Resistin, Age and BMI. R programming language has been used for this study.

Li and Chen (2018) worked breast cancer dataset from UCI in their study. One of the datasets is from Wisconsin Breast Cancer Database (WBCD) at the University of Wisconsin Hospitals, and the other dataset is Breast Cancer Coimbra Dataset (BCCD). BCCD is the same as the data used in this thesis. Five different classification algorithms that are Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR) and Neural Network (NN) are used in the study. RF is primarily applied to establish the classification model for predicting breast tumor class in female patients. Both breast cancer datasets are divided into random training and test data (70% training set and 30% test data). Firstly, the training data is used to model RF classification method and the test dataset is used to analyze the model. After the estimation done with RF, the results are compared with other methods such as DT, SVM, LR and NN. F-measure, accuracy and AUC value are calculated in all classification algorithms. No feature selection has been made by using any optimization algorithm in the study, and R programming language has been used. According to the experimental analysis RF classification algorithm is found as the best performer classifier among the five classifiers used in the study.

Aslan *et al.* (2018) have analyzed different machine learning methods for breast cancer diagnosis using the BCCD dataset taken from UCI repository. Machine learning techniques are important for early cancer diagnosis. In this study, 4 different algorithms such as ANN, Extreme Learning Machine (ELM), SVM and K-NN are used to diagnose early cancer detection with routine blood value analysis. In ANN, ELM, k-NN and SVM methods, hyperparameters with the least

error are determined by using Hyperparameter optimization technique. The results show that both the accuracy value and the running time of the ELM algorithm are better than the other methods applied.

The previous studies described in the above paragraphs contain BCCD data used in this thesis. It is seen that in these studies no feature selection has been made by using any optimization algorithm.

Antal and Hajdu (2014) have analyzed three different disease datasets that are Chronic Kidney Disease Dataset (CKDD), Diabetic Retinopathy Debrecen Dataset (DRDD), and Thoracic Surgery Dataset (TSD). CKDD consists of 25 features and 400 instances, DRDD consists of 20 features and 1151 instances, and TSD contains 17 features and 470 instances. CKDD and TSD datasets contain nominal and numeric values. On the other hand, DRDD contains only numerical attributes. They used Waikato Environment for Knowledge Analysis (WEKA) to calculate classification algorithm performance. Naive Bayes (NB), Simple Logistic (SL), K-Nearest Neighborhood (IBK), Bagging, Decision Tree (DT), Input Mapped Classifier (IMC), J48, Random Forest (RF), and Random Tree (RF) classification algorithms are used in the study. Accuracy, sensitivity, specificity and error rate values are calculated for each algorithm. Three different results are obtained by providing different combinations of these datasets. First, 400 samples are selected from each datasets and all selected classification algorithms are applied. The results showed that classification methods have higher accuracy for CKDD and TSD datasets than the DRDD. Secondly, DRDD dataset is increased by 1200 instances, CKDD and TSD sample sizes are not changed, and all selected classification algorithms are applied again. When DRDD samples are increased, accuracy values remain almost identical to the previous result. Thirdly, experiments are repeated by converting the numeric values to nominal values in the dataset. It is observed that nominal values give better results than numerical values. Consequently, it can be deduced that the application of algorithms on medical data yields more positive results if the concerned dataset contains more nominal attributes than numeric attributes.

Ramaswamy and Savitha (2016) have measured the success of various classification methods using the R programming language. Over several UCI datasets including Diabetic Retinopathy Debrecen dataset, Pima Indians Diabetes dataset, Chess dataset, Car evaluation dataset, and Liver disorders dataset. The classification of the datasets is done by using 8 different classification algorithms that are C5.0, Random Forest, Support Vector Machines, Gradient Boosting Machines, Naive-Bayes, AdaBoost Algorithm, and a Deep Learning based method. According to the experimental results Random Forest, Gradient Boosting Machines and Deep Learning based algorithm give the best classification accuracy rate for all datasets. All datasets are divided into 70% training and 30% test sets.

Maliha, Tareque and Roy (2018) have obtained the DRDD dataset from UCI. DRDD is the same as the data used in this thesis and the characteristics of the dataset is specified in the previous section of this thesis. The dataset is randomly divided into 80% training set and 20% testing set. They applied four machine learning algorithms; SVM, RF, KNN and Neural Networks (NNET) by using Python Pandas package. It is found that NNET algorithm gives the highest accuracy value as 75.32%.

Mohammadian, Karsaz and Roshan (2017) used Python Numpy package for classification of DRDD dataset. Nine classifiers that are Adaptive Boosting, DT, NB, Gaussian Process, KNN, Quadratic Discriminant Analysis (QDA), RF, SVM, Multi Layer Perceptron (MLP) are studied for comparative analysis. For the training and testing of the classifiers, 30 percent of the available data is randomly selected as the test set, while the remaining instances are used in the training set. The testing and training sets are kept the same in all classification algorithms. Parameter tuning is performed for the classifiers. Accuracy, Precision, Recall, and F1-Score values are calculated in each of these classification algorithms. Experimental studies showed that the Gaussian Process algorithm provides the best accuracy with 87% by selecting the optimum parameters. After the Gaussian Process, the second algorithm with the best accuracy is SVM with 84%. In their

study, DRDD dataset is used for classification without applying feature selection.

Buarque *et al.*, (2017) obtained 9 binary class datasets from the UCI repository for classification. These datasets are: Banknote, Bank, Climate, DRDD, Occupancy, PIDD, Spambase, VColumn, WDBC. They used Principal Component Analysis (PCA) to reduce data size. PCA is an unsupervised dimensionality reduction technique, but it is also used in supervised tasks such as classification and regression. The best value is calculated for each eigenvector in PCA. For their proposed feature selection method, a score is calculated for each feature and selecting *n* features with highest score are selected. However, in PCA, dimension of dataset is reduced by mapping the original feature set to a lower dimensional feature set by applying a kind of compression technique. First of all, they have proved that their proposed method is more successful than Principal Component Analysis by using an artificial dataset. DT, NB, Linear Discriminate and 1-NN classification methods are used to calculate mean accuracy on the artificial dataset, and then comparisons are done with the raw data. Using the DT classifier, raw data has been classified with 80.3% accuracy, when PCA is applied accuracy is reduced to 52.1%, and with their proposed method improves accuracy up to 93.9%. When NB classifier is applied raw data is classified at 78.5% accuracy, PCA reduces accuracy to 55.0%, and the proposed method improves to 94.5%. For the Linear Discriminate (LD) classifier raw data has 97.2% classification accuracy, while classification accuracy changes to 61.0% and 95.3% with the PCA and the proposed method, respectively. Finally, 1-NN classifier has classification accuracy as 93.0% without any dimensionality reduction, when PCA and the proposed method are applied, classification accuracy changes to 50.9% and 92.9%, respectively. Results of the proposed method are higher than the PCA. Then the two methods are applied to the nine real datasets. Usually the proposed method has higher accuracy than PCA. According to the experimental results, the accuracy of the proposed method is higher for the DRDD dataset compared to PCA for all classification methods.

So far, the summarized studies that use BCCD and DRDD datasets generally involve classification task without applying feature selection. In the following paragraphs, feature selection studies using similar dataset about diabetes and breast cancer are explained. And then, feature selection studies with PSO on healthcare data are examined.

Lavanya and Rani (2011) have analyzed the effects of feature selection methods for classification of breast cancer datasets. They have used the Weka program while selecting the features. They obtained tree different breast cancer datasets from UCI that are Breast Cancer, Breast Cancer Wisconsin (Original) and Breast Cancer Wisconsin (Diagnostic). In this study, CART is selected as one of the decision tree classification algorithms for the analysis of breast cancer data. Firstly, the classification accuracy is calculated using the CART algorithm without applying any feature selection. The accuracy values obtained without feature selection are as follows; for Breast Cancer accuracy value is 69.23%, for Breast Cancer Wisconsin (Original) accuracy value is 94.84%, and Breast Cancer Wisconsin (Diagnostic) accuracy value is 92.97%. Then the accuracy value is computed by using the feature selection algorithms in the Weka software and all results are compared. When compared to all feature selection methods, the SVMAttributeEval method is found as best for Breast Cancer Dataset with a accuracy of 73.03%, PrincipalComponentsAttributeEval method is the best for Breast Cancer Wisconsin (Original) Dataset with an accuracy of 96.99%, SymmetricUncertAttributesetEval method is the best for Breast Cancer Wisconsin (Diagnostic) Dataset with accuracy of 94.72%. All these results indicate that the feature selection method increases the accuracy of classification process.

Aalaei *et al.*(2016) have applied feature selection on three breast cancer datasets which are Wisconsin Breast Cancer Dataset (WBC), Wisconsin Diagnosis Breast Cancer (WDBC) and Wisconsin Prognosis Breast Cancer (WPBC) from UCI. In this paper, a feature selection method using a Genetic Algorithm (GA) is proposed to select the best feature subgroup for breast cancer diagnostic datasets.

Artificial Neural Network (ANN), Particle Swarm Classifier and GA Classifier are used to evaluate the feature selection method obtained by GA in Wisconsin Breast Cancer datasets. MATLAB is used for modeling this application. Datasets are divided into two sets that are 80% for training dataset and 20% as test dataset. Accuracy, sensitivity and specificity metrics are calculated with and without feature selection for evaluating performance of classification. As a result, it is observed that the selection of features increases the classification accuracy for all datasets.

Gandhi and Prajapati (2014) examined feature selection methods using Pima Indian Diabetes Dataset (PIDD) obtained from UCI. PIDD contains 9 features, two classes and 768 instances. Various feature selection techniques and classification algorithms are used in the study. F-score, FCBF, Relief F, SVM – RFE, SVM, GA are used for feature selection and SVM and ANN are used for classification in the study. F-score feature selection technique with SVM classification method has the best classification accuracy which is equal to 98.92%. The GA feature selection technique with ANN classification method has the lowest accuracy result with 78.26%. As a result, F-score is the method of feature selection which gives the best results in all studies.

Vaishali *et al.* (2017) have analyzed the PIDD dataset, a publicly available dataset from UCI machine learning repository. This dataset is a compilation of the healthcare information of the Indian female population of Pima, aged 21 and over in Arizona and Phoenix. The PIDD dataset is divided into two parts: 70% training set and 30% test set. In this study, their aim is to increase the classification accuracy by selecting the features. Firstly, Genetic feature selection method is applied to decrease the number of features in the dataset. Then, several classification algorithms are used to measure the performance of the reduced and the original datasets. The classification algorithms used are: NB, J48 Graft, MLP NN and Multi-Objective Evolutionary Fuzzy Classifier (MOE Fuzzy). MOE Fuzzy is an evolutionary computation technique that simulates the nature of evolution. It

works according to the survival theory of the best. The fitness function is used to find the best value in the population. The experiments are made in WEKA Software. CFS Subset Evaluation technique is applied to reduce feature set. The classification accuracies of the algorithms are compared with and without FS. It is found that the selection of features increases the classification accuracy for all classifiers, and MOE Fuzzy has the best classification accuracy.

Various researchers have performed comparison on PSO feature selection to get better classification accuracy using various classifiers. Hamed *et al.* (2018) have analyzed feature selection with PSO using medical data. The medical data used in the study are: breast cancer, hepatitis, thyroid, diabetes, lung cancer, parkinson, dermatology, nervous system and similar health related data. Researchers used different classification techniques with PSO to achieve better classification accuracy and then compared all the results. It has been showed that the PSO based fast K-means algorithm has the best success with 99.39% accuracy.

Zarchi, Fatemi Bushehri, and Dehghanizadeh (2018) have analyzed the SCADI dataset, a publicly available dataset from UCI machine learning repository. They use two different expert systems in their work. The first one is the ANN based classifier and, the second one is the Decision Tree which uses the C4.5 algorithm. All features are used in both expert systems. In ANN experiment, it is repeated several times with different number of neurons for the hidden layer to obtain optimal ANN parameters. The best precision obtained with 40 ANN hidden neurons. Then 10-fold cross-validation approach is used to evaluate the proposed model and the best accuracy found in the ANN expert system is 83.1%. Decision tree is a rule based expert system and in this study, decision tree is used to derive self-care problem classification rules by IF-THEN rules.

El Moudden, Ouzir, and ElBernoussi (2017) have applied feature selection and extraction on LSVT dataset obtained from UCI machine learning repository. They used Principal Component Analysis (PCA) to apply dimensionality reduction to the dataset. PCA is one of the oldest and unsupervised techniques commonly

used to reduce data size. The best value is calculated for each eigenvector in PCA. In this study, PCA is used with both Logistic Regression (LR) and C5.0 Decision Tree (DT) algorithm. Firstly, the feature selection method is reduced the number of features to 114. Then the accuracy values for the 6 and 13 common features estimated by PCA + LR and PCA-C5.0 are found. The mean performance results reached to 90% by using PCA-LR model, and %91.72 with PCA+C5.0.

In Appendix 1, a table which compares the related work with this thesis study is presented. As shown in the table in the appendix, this thesis is the first study that applies the PSO based feature selection on the BCCD, DRDD, SCADI, and LSVT datasets to improve performance of diagnosis of breast cancer, diabetic retinopathy, children's self-care problems, and speech disorders of Parkinson patients, respectively.

3. MATERIALS AND METHODS

This section contains two sub-sections: materials and methods. Firstly, properties of all datasets used in this study and the software and programming languages employed are explained in detail in the material section. Later, the algorithms and techniques applied in the study are presented in the method section.

3.1. Materials

3.1.1. Datasets

In this study the UCI machine learning repository is used as a source of dataset. UCI repository contains many disease related datasets in the healthcare area. Four datasets related with four different diseases from the UCI are studied in this thesis. These datasets are Breast Cancer Coimbra Dataset (BCCD), Diabetic Retinopathy Debrecen Dataset (DRDD), Self-Care Activities Dataset based on International Classification Functioning – Children Youth (SCADI), and Lee Silverman Voice Treatment (LSVT) Dataset. The following subsections describe these datasets in detail. Table 3.1 shows the number of features, number of instances, and year information of datasets used in this thesis. As shown in Table 3.1, each dataset has different number of features and instances to make better evaluation of the methods applied.

Table 3.1. Characteristics of datasets received from UCI

Dataset name	# of classes	# of features	# of instances	Year
Breast Cancer Coimbra	2	10	116	2018
Diabetic Retinopathy Debrecen Dataset	2	20	1151	2014
Self-Care Activities Dataset	7	205	70	2018
Lee Silverman Voice Treatment	2	310	126	2014

3.1.1.1. Breast Cancer Coimbra Dataset

Breast cancer is a common form of cancer that usually appears in women. World health organization (WHO) announced that breast cancer cases are increasing every year and breast cancer cases reached 2 million in 2018. According to WHO, breast cancer causes the death of 15 out of 100 women (World Health Organisation, 2018). Breast cancer screening is an important method for early diagnosis. Scientists are working with artificial intelligence algorithms to help diagnose the disease.

In this study, BCCD dataset obtained from the UCI repository is used. This dataset includes blood analysis of women with and without breast cancer. BCCD dataset contains 10 features and 116 instances, and it is donated on March 6th, 2018. The dataset has numeric attributes, and it also contains class labels. Last attribute is the class label which consist of 1 and 2 so that 1 means healthy controls and 2 stands for patient. It has 9 attributes except the class label. The 9 attributes are described below:

i. Age: The women's age.
ii. BMI: Body mass index.
iii. Glucose: Shows the level of sugar in the blood.
iv. Insulin: A hormone that regulates the level of glucose in the blood.
v. HOMA: Homeostatic Model Assessment of Insulin Resistance.
vi. Leptin: Leptin is satiety hormone.
vii. Adiponectin: is a hormone that provides fat burning and regulates glucose levels in the blood.
viii. Resist: Blood flow resistance
ix. MCP.1: Monocyte Chemotactic Protein-1
x. Class Label: 1= Healthy controls, 2= Patient

The dataset is used to diagnose a women's breast cancer. The BCCD contains 116 data samples and no missing data. Dataset consists of 64 patients with breast cancer and 52 healthy women.

3.1.1.2. Diabetic Retinopathy Debrecen Dataset

Diabetes, also known as blood sugar, is a disease that occurs when blood sugar is high. Diabetes causes many damages to the human body such as nerves, kidneys, foots and eyes. The damage caused by diabetes in the eyes is called diabetic retinopathy. Diabetic retinopathy (DR), one of the most important preventable and / or treatable causes of blindness worldwide, is one of the most important complications of diabetes (Kurt et al., 2017). In the next 10 years, the number of diabetic patients is expected to increase, thus, diabetic diseases such as retinopathy will also increase (Nentwich, 2015). Diabetic retinopathy datasets are important sources that scientists use for their research.

DRDD dataset from the UCI repository is used in this study. This dataset contains attributes used to estimate whether an image contains signs of diabetic retinopathy or not. DRDD has different types of features that are integer and real. The value here refers to different retinal points of diabetes patients (Maliha et al., 2018). Dataset was donated to UCI repository on 2014-11-03. It contains a total of 20 attributes, one of which is a class label. The class labels consist of binary numbers 0 and 1: ‘0’ means no sign of retinopathy, and ‘1’ means patient has retinopathy. DRDD contains 1151 data samples and no missing data. Dataset contains 611 patients that have diabetic retinopathy, and 540 instances of no sign of diabetic retinopathy. Features are as follows:

i. q – Quality evaluation values 0 = bad, 1 = sufficient.
ii. ps - Shows pre-screening results,1 = retinal abnormality, 0= retinal lack
iii. nma.a - nma.f - Microaneurism detection result. Each attribute refers to the number of safe microaneurism.

iv. nex.a - nex.f – It contains the same information as nma.a – nma.f for fluid accumulated (exudates) in the body. Exudate is the fluid that accumulates in the body as a result of any injury and inflammation.
v. dd - It refers to the Euclidean distance of the center of the macula and the center of the optic disc to obtain information about visual damage in patients.
vi. dm – The optic disc diameters.
vii. amfm – The AM/FM-based classification result.
viii. Class – Class label. 0= patient without DR, 1= patient with DR.

3.1.1.3. Self-Care Activities Dataset

Self-care skill is the ability of children to undertake their own personal care in accordance with their developmental stages in preschool period. In this period, children are expected to develop these skills in accordance with their developmental characteristics. Some children may have inadequate self-care skills. In this case, self-care problems arise.

Self-care problems diagnosis and classification is an important challenge in exceptional children health care systems (Zarchi et al., 2018). The SCADI dataset from UCI repository, which is used in this study, is the only dataset that is available to classify self-care problems based on ICF-CY to this date. SCADI data contains 206 attributes, 70 samples and it was donated to UCI in year 2018. The 'class' label refers to children's self-care skills in physical and motor abilities, and there is no missing data. The attributes of SCASI are as follows:

- 1: gender: 1=male, 0= female.
- 2: age: Age in years.
- 3 to 205: self-care activities based on ICF-CY (1 = The case has this feature; 0 = otherwise)

Classes: (class1 = Caring for body parts problem; class2 = Toileting problem; class3 = Dressing problem; class4 = Washing oneself and Caring for body parts and Dressing problem; class5 = Washing oneself, Caring for body parts, Toileting, and Dressing problem; class6 = Eating, Drinking, Washing oneself, Caring for body parts, toileting, Dressing, Looking after one's health and Looking after one's safety problem; class7 = No Problem).

3.1.1.4. Lee Silverman Voice Treatment Dataset

Lee Silverman Voice Treatment (LSVT) is a form of treatment for speech disorders related to Parkinson's disease (https://en.wikipedia.org). LSVT dataset is composed of a range of biomedical speech signal processing algorithms from 14 people (8 male and 6 female) had an age range of 51 to 69 years who have been diagnosed with Parkinson's disease undergoing LSVT. LSVT data contains 309 attributes, 126 samples and it was donated to UCI in 2014. The class labels consist binary numbers 1 and 2: '1' means acceptable and '2' means unacceptable.

3.1.2. Hardware

Hardware is the most important requirement to develop the system used in this thesis. The qualifications of the hardware are very important to run the program in a short time. The hardware used in this thesis has the properties as given in Table 3.2.

Table 3.2. Hardware Details

Name	Detail
Operating System	Microsoft Windows 10 Pro
CPU	Intel(R) Core(TM) i7-4790
RAM	16,0GB
System Type	64 bit
Clock speed (Ghz)	3,60Ghz
Graphics Processing Unit (GPU)	NVIDIA Geforce 210

3.1.3. Software

To implement the methods used in this study Python programming language and Weka software are used.

3.1.3.1. Python

The Python programming language is written by a Dutch programmer who is called as Guido Van Rossum. Development started in 1990 (www.python.tc). Python works on many platforms such as Windows, Linux / Unix and Mac-OS. Python is a programming language that enables you to work quickly, integrate and effectively into your system (www.python.org). Therefore, it is a highly preferred programming language. It is the most widely used programming language in data mining and machine learning fields. In 2019, it is ranked 3rd in the world's most widely used programming language. Python is a programming language that is easy to learn, fast running, easy to process data files. There are many libraries to be used in data mining. Therefore, Python was chosen in this study. Table 3.3 shows the details of the Python packages that were used when performing this work. The Python packages specified in the table provided great convenience in implementing the application.

Table 3.3. Python Programming Details

Name	Type	Version
Python	Python Interpreter	Python 3.7
PyCharm	Python IDE	2018.2.4
Numpy	Python Package	1.16.3
Scikit-Learn	Python Package	0.20.0
Pandas	Python Package	0.23.4

3.1.3.2. WEKA Data Mining Tool

WEKA (Waikato Environment for Knowledge Analysis) (http://www.cs.waikato.ac.nz/ml/weka) is a widely used machine learning software in the world and it was developed at the University of Waikato (New Zealand). The software is written in the Java programming language and free software licensed under the General Public License (GNU). In Figure 3.1. WEKA's user interface is shown. Version 3.8.1 of Weka software is used in this thesis.

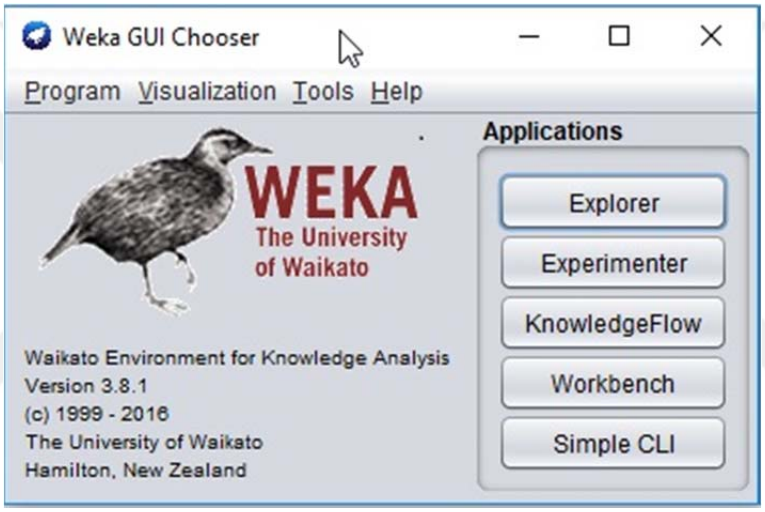

Figure 3.1. WEKA GUI Chooser

When you start WEKA, the GUI window appears and lets you to choose one of the five ways to work with WEKA (Explorer, Experimenter, Knowledge Flow, Workbench, Simple CLI) (Kaur and Dhiman, 2016). WEKA is a program that supports the basic tasks of data mining. These tasks are; data preprocessing, classification, clustering, regression, visualization and feature selection (Kaur and Dhiman, 2016). In this study, Weka Explorer options were used. Figure 3.2 shows the Explorer interface.

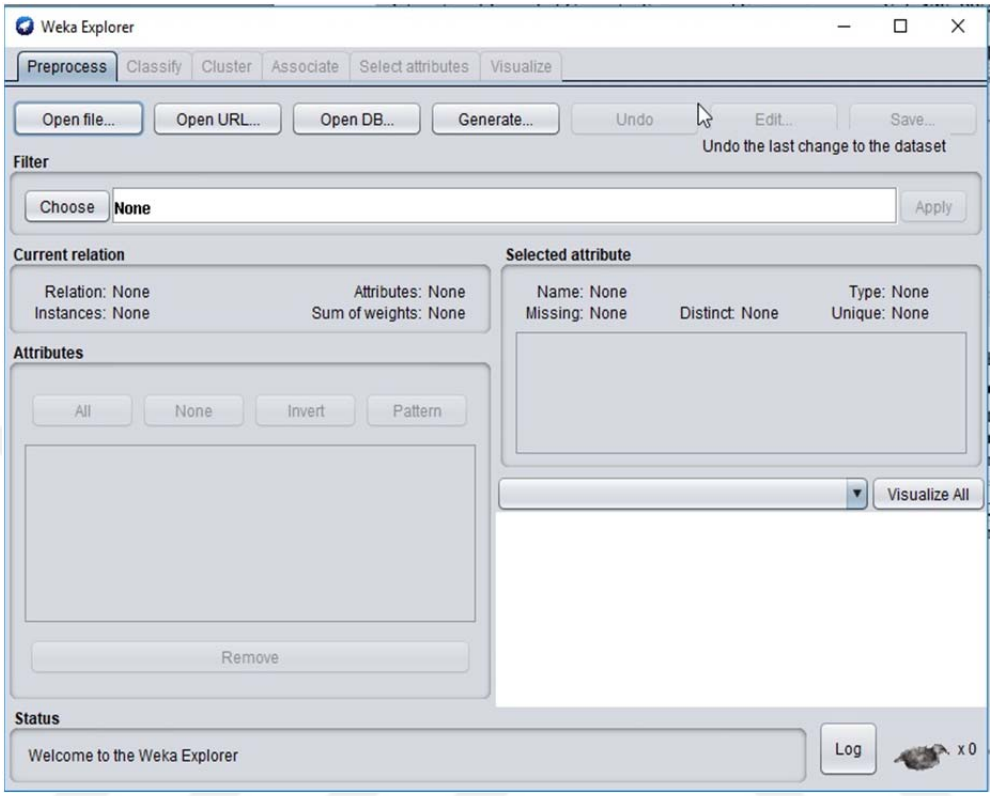

Figure 3.2. WEKA Explorer

WEKA uses a special file format. It is called Arff (Attribute-Relation File Format) which consists of 3 parts: relation, attribute, and data. Attributes are defined in the attribute section; examples are written in the data section in which each line represents a data instance. Attribute type can be numeric, real, or categoric. The Arff file format for dataset iris which has five attributes is shown in Figure 3.3. The class attribute is the class label of the dataset.

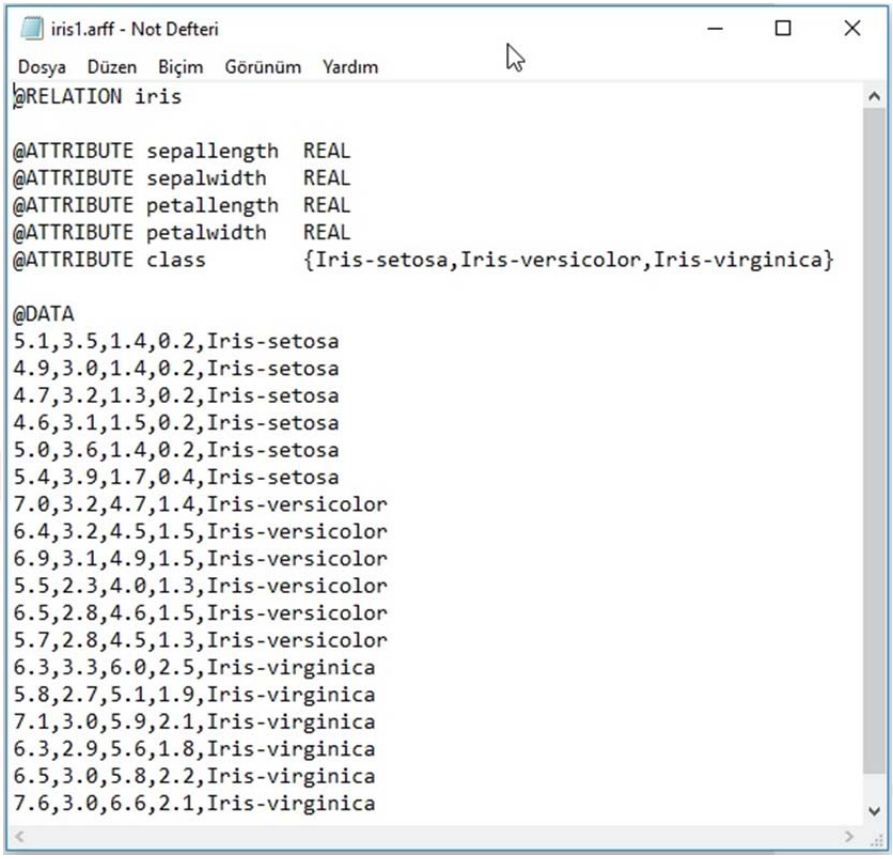

Figure 3.3. Arff file format

3.2. Methods

In this thesis, our aim is to apply PSO based feature selection method to improve classification performance of healthcare data to make diagnosis of some diseases. To reach our goal, we apply PSO based feature selection technique with Naïve Bayes classifier, and compare performance of the used feature selector with the well-known feature selection methods that are information gain, chi-square, and relief by applying classifiers not only Naïve Bayes classifier but also Support Vector Machines, and Random Forest. In Figure 3.4 the main architecture of the system developed is presented. In the below subsections, methods used for feature selection and classification are presented.

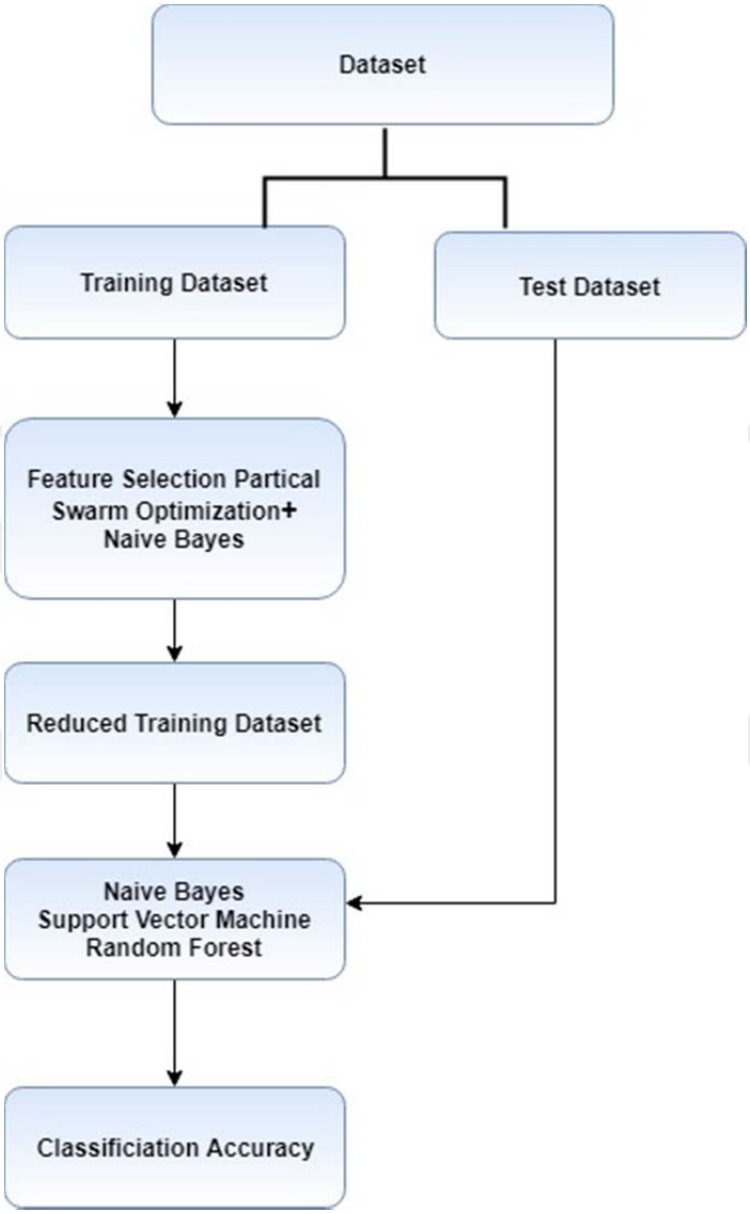

Figure 3.4. Flowchart of the applied feature selection and classification model

3.2.1. Particle Swarm Optimization

Nowadays, many methods that are inspired by biological systems are used to solve optimization problems (Tamer and Karakuzu, 2006). Social systems,

another type of biological systems, examine the interactions and common behavior of the individuals, especially with the environment and other individuals to solve some problems (Tamer and Karakuzu, 2006). This behavior is called as swarm intelligence. The swarm intelligence behavior is observed in ants, birds, bees, fish flocks and bacterial communities (Ordukaya, 2010).

Particle Swarm Optimization (PSO) is an optimization algorithm developed by Eberhart and Kennedy, inspired by movements that exhibit flocks in nature and first appeared in 1995 (Kennedy and Eberhart, 1995). PSO is a population-based optimization method inspired by two-dimensional behavior of flocks of birds and fish. PSO succeeds in solving non-linear problems. PSO has a simpler calculation than other conventional optimization methods, and does not include time-consuming derivative operations. Therefore, it works faster and the computation times are shorter and more preferred (Der et al., 1995). The PSO scenario is as follows: there is a flock of birds in an area with only one food. Birds are randomly located in this area of food, and no bird knows where the food is. But they know how close they are to food at the end of each iteration. In this case, it is a good decision to follow the bird closest to the food. PSO works according to this scenario and it is used to solve the optimization problems. Birds trying to find food in the problem space area are called “particle” at PSO. Each particle has fitness value and velocities that direct it to fly. These are calculated with the fitness function. Particles fly out of the problem space following the optimum particle at each iteration (Eberhart and Shi, 2002).

PSO is started with a group of random solution (particle swarm) and an optimum solution is found with updates. In each iteration, the particle positions are updated according to the two best values. The first best value is the coordinates that provide the best solution the particle has ever obtained. This value is kept in memory for later use and called ‘pbest’. The other best value is the coordinates that provide the best solution ever obtained by all particles in the population. This best value is global best and called ‘gbest’ (Eberhart and Shi, 2002). There are *n*

particles in a *D* dimensional problem space. *n* is the number of particles, *d* is the size of the space. In this case, particle population matrix has the form as shown in equation 3.1.

(Tamer and Karakuzu, 2006).

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nD} \end{bmatrix}_{nxD} \quad (3.1)$$

PSO initially has been proposed for solving continuous optimization problems. Kennedy and Eberhart developed a binary PSO (BPSO) that can be used for discrete problems (Tran et al., 2014a).

3.2.1.1. Continuous Particle Swarm Optimization

PSO is a population-based optimization technique. Initially, the particles in the solution space are randomly distributed. Then, in each iteration, the values are updated and they are directed to the best result. Kennedy and Eberhart first developed classical PSO which is called Continuous PSO (CPSO).

In CPSO, each particle in *D* dimensional problem space represents a solution. In matrix given in equation 3.1, the particle can be represented as $\boldsymbol{x}_i = [\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots\ldots, \boldsymbol{x}_{iD}]$. The best previous position $pbest_i$ (the position giving the best fitness value) of any particle i is recorded and represented as $pbest_i = [\boldsymbol{pbest}_{i1}, \boldsymbol{pbest}_{i2}, \ldots\ldots, \boldsymbol{pbest}_{iD}]$. The best particle index between all the particles in the population is represented by the *g* symbol, and the global best of $\boldsymbol{pbest}_i$ value is, $\boldsymbol{g} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots\ldots, \boldsymbol{g}_D]$ and called ***gbest.*** The velocity for the *i*th particle can be represented as $\boldsymbol{v}_i = [\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, \ldots\ldots, \boldsymbol{v}_{iD}]$ and is limited by V_{max} a user-defined variable (Shi and Eberhart, 2002).

Each particle moves through multi-dimensional search space and adjusting toward both:

i. the particle's best position found thus far, and
ii. the best position of the neighbor of that particle.

Each particle maintains the following information

- $\boldsymbol{x}_i$: The current position of the particle;
- $\boldsymbol{v}_i$: The current velocity of the particle;
- $\boldsymbol{pbest}_i$: The personal best position of the particle.

The best individual position of the particle refers to the best position the particle has visited until it reaches its current position x_i. The best location of the particles is determined by a fitness function. If it is a minimization problem, the position that best minimizes the function has the highest conformity value (Doğan, 2009). For the minimization problem, the best personal position of the particle *i* which is represented as $y_i^{(t+1)}$is calculated as in equation 3.2, where *i* is the number of the particle, *t* is the number of iterations, *f* is the fitness function, and y_i^t is the best personal position of the particle *i* in the previous iteration.

$$y_i^{(t+1)} = \begin{cases} y_i^t & if \ \ f(x_i^{(t+1)}) \geq f(y_i^t) \\ x_i^{(t+1)} & if \ \ f(x_i^{(t+1)}) < f(y_i^t) \end{cases} \tag{3.2}$$

After finding the values of *pbest* for each particle in the swarm, the global best *gbest* is found, and then the velocities and positions of the particles are updated according to the equations in 3.3 and 3.4.

$$v_i^{t+1} = w.v_i^t + c_1.rand_1^t.\left(pbest_i^t - x_i^t\right) + c_2.rand_2^t.\left(gbest^t - x_i^t\right) \quad (3.3)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3.4)$$

In equation 3.3, c_1 and c_2 are learning factors. c_1 and c_2 are constants expressing stochastic acceleration terms that pull each particle toward $pbest$ and $gbest$ positions. c_1 allows the particle to move according to its own experience and c_2 to move according to the experience of other particles in the swarm. Too small or too large values of c_1 and c_2 are a disadvantage in terms of not finding the target area. Researches show that usually c_1= c_2=2 gives good results (Tamer and Karakuzu, 2006). $rand_1$ and $rand_2$ are uniform random numbers between 0 and 1. The *t* value indicates the number of iterations. Inertia weight (w) is a parameter that allows to control the local research ability of the swarm. The range [0.9, 1.2] shows effective performance for the weight of inertia (Yaman, 2014). Particles can move out of solution space when moving in problem space. Therefore, the speed is limited in the range of [-V_{max}, V_{max}].

The application process of the algorithm is as described below.

i. Initialize random position and speed values for each particle. Speed values usually assigned to *'0'*.
ii. For each particle, the fitness value is calculated according to the fitness function.
iii. Each particle's fitness value is compared with its $pbest_i$ value. Update the best position of each particle with the equation in 3.1. If current value is better than $pbest_i$, set the $pbest_i$ value equal to the current values.
iv. The best $pbest_i$ value is assigned as the global best ($gbest_i$). If current value is better than, $gbest_i$, set the $gbest_i$ value equal to the current value.

v. Change the velocity and position of the particle with the equations 3.3 and 3.4.

The loop continues until the termination criteria are met. Termination criterion is usually the number of iterations or the best fitness value (Eberhart and Shi, 2002). The flowchart of the PSO is presented in Figure 3.5.

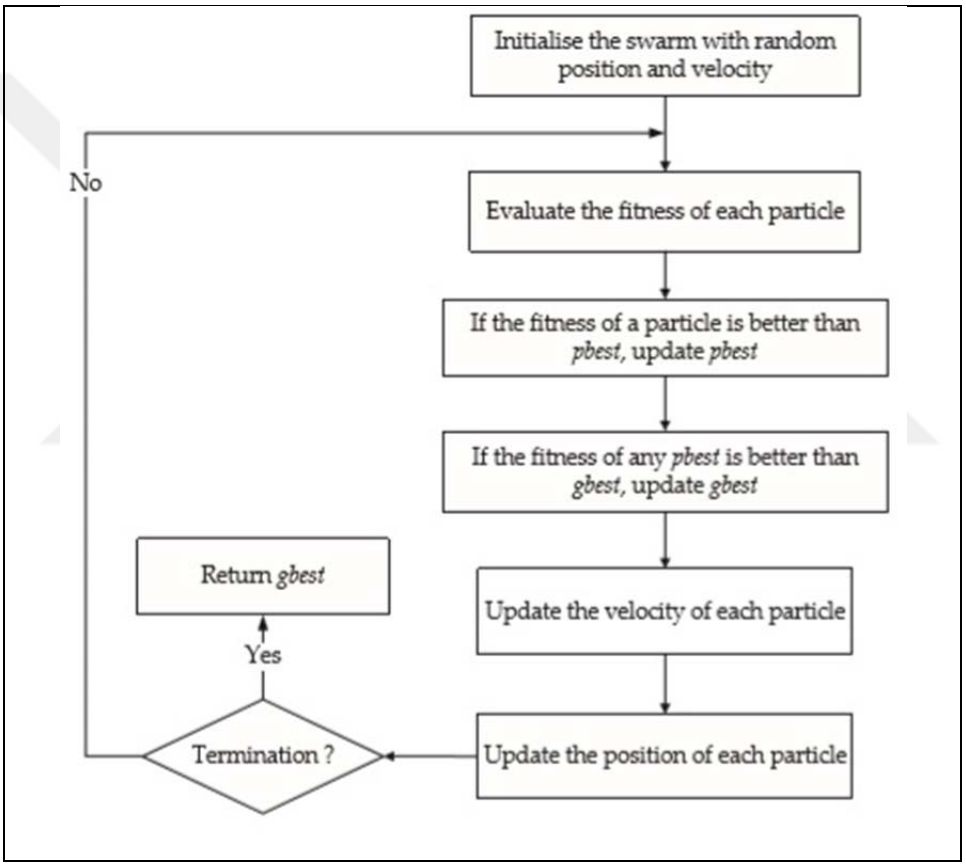

Figure 3.5. The Flowchart of PSO

3.2.1.2. Parameters of PSO

PSO is not an algorithm with many parameters, below is a list of parameters and their descriptions.

Particle number: *n* is the number of particles. It generally ranges from 20 to 40. In some difficult problems this value can be 100 or 200.

Particle size: It varies depending on the size of the problem to be optimized.

Particle range: Each dimension of the particle is initially defined as random value in the range [$-X_{max}, X_{max}$]. It depends on the problem to be optimized.

V_{max}: It determines the maximum change (velocity) that will occur in a particle. For example, if the range of a dimension of a particle is [-10, 10] , V_{max} can be 20.

Learning Factors: Usually $c_1 = c_2 = 2$ gives good results.

***W* (Inertia Weight):** It is a parameter that allows to control the local research ability of the swarm. It is generally in the range [0.9, 1.2].

Stop Criteria: The algorithm can be stopped if the specified maximum number of iterations or desired fitness value is reached.

3.2.1.3. Binary Particle Swarm Optimization

PSO was originally developed for real-number optimization problems, then Kennedy and Eberhart developed Binary PSO (BPSO) for the solution of discrete problems (Yang et al., 2008). BPSO, similar to the normal PSO algorithm, was proposed when the solution space is discrete. In step 1, it uses the equation 3.5, which is similar to equation 3.3 of velocity in the CPSO. Although the velocity equation of the particle is similar, the position update equation is completely different. After computing the velocities as in equation 3.5, the velocity value is sent to a sigmoid function (see equation 3.6) to maintain the speed values in the range [0, 1]. Speed value is compared to the *rand* variable in range [0,1] as in equation 3.7, then position values remain as 0 or 1. Then the location of the particle in the solution space is calculated (Kılıkçıer and Ersen, 2012).

$$v_{id}^{t+1} = w.v_{id}^{t} + c_1.rand_1^t.(pbest_i^t - x_i^t) + c_2.rand_2^t.(gbest^t - x_i^t) \quad (3.5)$$

$$S(v_{id}) = \frac{1}{1-e^{-v_{id}}} \quad (3.6)$$

$$x_{id} = \begin{cases} 1 & \text{if } rand < S(v_{id}) \\ 0 & otherwise \end{cases} \quad (3.7)$$

3.2.2. Feature Selection Methods

The dataset to be analyzed may contain many attributes, but not all of them are very important and relevant attributes. Therefore, these attributes must be eliminated from data without loss of the information (Sheena et al., 2016). The feature selection is defined as the selection of the best subset representing the original dataset. Feature selection is the process of selecting the best *k* attribute among *n* attributes in the original dataset. The purpose of the feature selection is to reduce the number of features in the dataset by selecting the most useful and important attributes in the original dataset (Budak, 2018). Advantages of feature selection are listed as below (Paul, 2005):

i. Reduces the size of the feature space in the dataset and improves the computation speed of the data analysis algorithm.
ii. Eliminates non-relevant and noisy data.
iii. Improves data quality.
iv. Makes the dataset more simple, identifiable and understandable.
v. Reduces the amount of memory required to store data.
vi. Reduce the computation time.
vii. Increases the accuracy of the model.
viii. Performance improvement, gain accuracy in estimation.

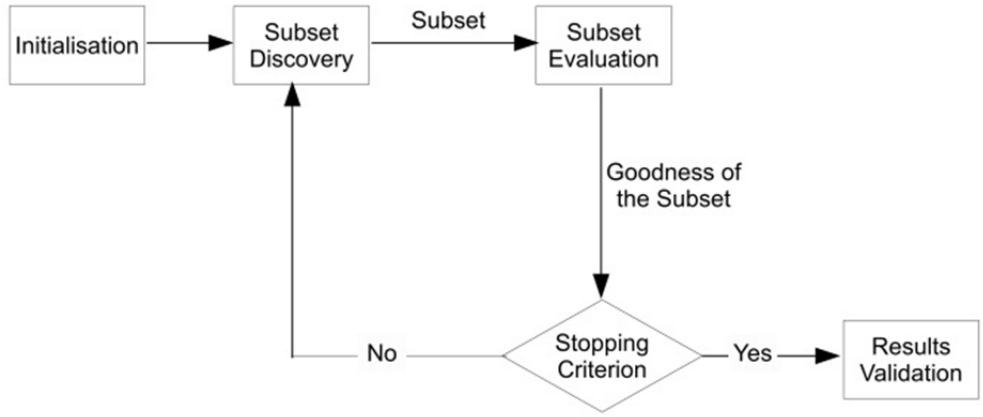

Figure 3.6. General Feature Selection Process

Figure 3.6 shows the five basic steps of feature selection (Tran et al., 2014a).

i. Initialization: Feature selection starts with a method that considers all the original features.
ii. Best Subset Search: This is a subset search method. Subset search can start with no features, all features or a random set of features.
iii. Subset Evaluation: Evaluates whether the generated subset yields a good result.
iv. Stopping Criteria: The subset selection can be stopped if the specified maximum number of iterations or desired fitness value is reached.
v. Validation: A validation is performed to check if the subset is valid.

3.2.2.1. Feature Selection Approaches

Feature selection is usually divided into 3 categories: filter, wrapper, and embedded methods. In filter methods, several statistical methods are used before the data mining algorithm is run. In the wrapper method, the feature selection depends on the data mining algorithm. Embedded method is an approach where

both data mining algorithm and statistical methods are used together (Budak, 2018).

i. Filter Methods

The method that selects features without using the classification algorithm is called filter methods. Data mining algorithms are used after finding the best attributes. Filter methods can sort individual attributes or evaluate all feature subset. Several statistical methods are used for filtering. These are: information, distance, similarity, consistency and statistical measures. The filter method has two types: univariate filter and multivariate filter. Univariate feature filters are based on a single feature and multivariable feature filters are based on the entire attribute subset (Jović et al., 2015). This method has some advantages and disadvantages. The advantages are they are simple and independent of the classification method to be used. Attribute selection is applied only once. The disadvantage is that it ignores interaction with the classifier and treats each feature separately (Sheena et al., 2016). In Figure 3.7 basic steps of using filter based feature selection method is presented.

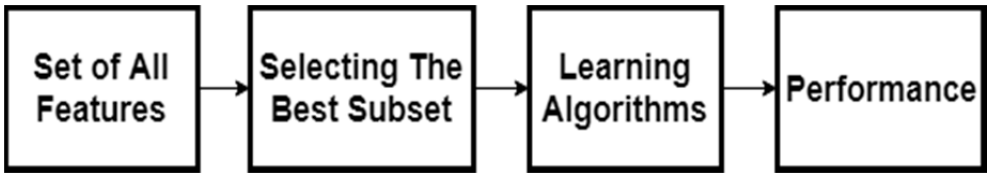

Figure 3.7. Filter Method

ii. Wrapper Method

The selection of feature in the wrapper method depends on the data mining algorithm to be employed. Feature selection is done by using the classification accuracy. It is decided whether the features are good according to the result of the classifier (Sheena et al., 2016). The wrapper method is more successful than the filter method in obtaining a subset of the best features. However, the cost of

computation is higher than the filter method (Budak, 2018). In a wrapper method, the quality of the subset of the attribute is directly measured by the performance of the classifier. Different from the filter method, it interacts with the classifier algorithm and also takes the attribute dependencies into consideration (Sheena et al., 2016). In a wrapper model, the feature selection algorithm interacts with a classification algorithm and the classification algorithm is the "black box" of the feature selection algorithm. Compared to filtering algorithms, wrappers generally provide better classification performance in the feature selection process. This is because each selected subset is sent to the classification algorithm and the attributes are evaluated according to the classification result (Tran et al., 2014b).

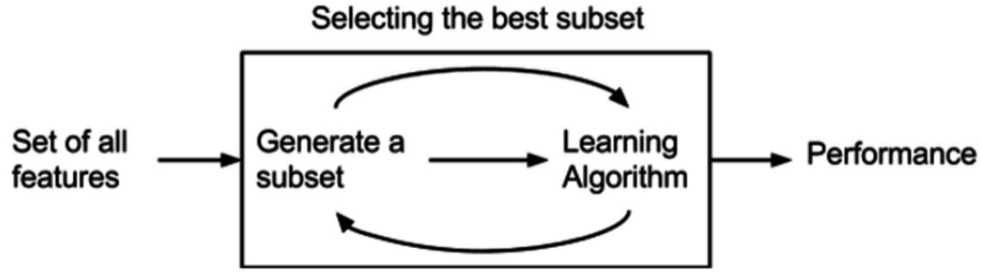

Figure 3.8. Wrapper Method

iii. Embedded Method

Embedded methods accommodate both classification algorithm and attribute selection algorithm, perform classification and attribute selection processes simultaneously. Embedded methods have a higher computational cost than filtering methods, just as in wrapper methods (Budak, 2018). Embedded methods are not a new method. Decision trees such as CART, for instance, have a built-in mechanism to make variable selections (Guyon and Elisseeff, 2003). This method aims to find the most appropriate subset as others. Feature subset is selected with a feature selection algorithm then, classifier is applied to the selected subset. After that, the performance of the classifier is evaluated and if the performance is good, the method is terminated.

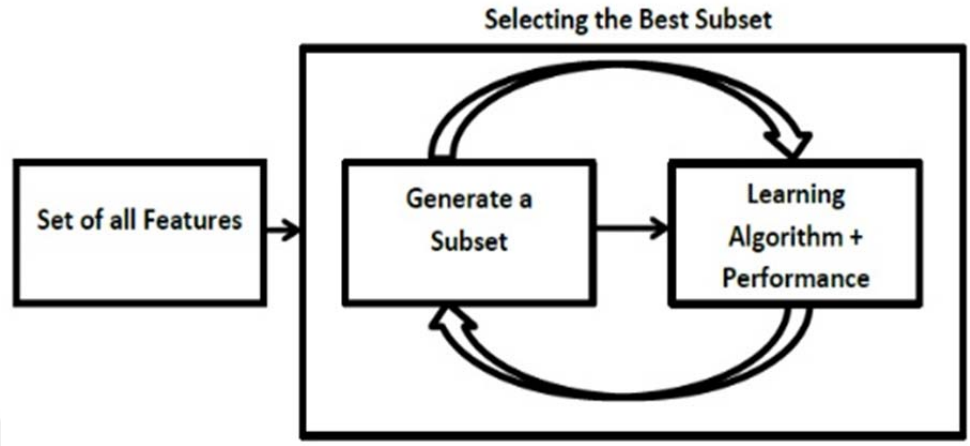

Figure 3.9. Embedded Method

3.2.3. Particle Swarm Optimization for Feature Selection

Both the CPSO and the BPSO are used for the feature selection. In CPSO, each particle in the swarm is expressed as the *n* dimensional real number vector. But, in BPSO the representation of a particle is an *n*-bit binary string. If BPSO is used to solve feature selection problems, the representation of a particle must be an *n*-bit binary string. (Tran et al., 2014a). Each bit represents a feature. If the bit value is "1", this feature is selected and if the bit value is "0", this feature is not selected (Aghdam and Heidari, 2015). The features represented by "1" constitute the new feature subset.

3.2.4. Naïve Bayes Classification

Naïve Bayes classification algorithm is a statistical method used for classification and categorization problems. This algorithm calculates a probability set by counting the frequency and combinations of values in a given dataset, and it is based on the Bayes' Theorem (Patil and Sherekar, 2013). It is a simple classification model. Therefore, it is preferred in analysis of large datasets. Because it is a simple method, it gives good results compared to other methods. For this reason, there is a very common usage area (Arslan, 2018). However, there are

many disadvantages due to the fact that data types are very simple and pure. The Bayesian classification method is used in analysis of data from many different fields such as health, genetics and industry (Gemici, 2012).

The Bayesian theorem is a statistical method that defines the outcome probabilities of related (dependent) events using the concept of conditional probability. Let *A* and *B* be two non-discrete events. The probability that two events, *A* and *B* will occur is P ($A \cap B$), which is the product of the probability of *A* to occur is P (*A*), with the probability of *B* given that *A* has occurred P($B \mid A$) as in equation 3.8.

$$P(A \cap B) = P(A)\, P(B \mid A) \tag{3.8}$$

On the other hand, the probability of *A* and *B* is also equal to the probability of *B* times the probability of *A* given *B*.

$$P(A \cap B) = P(B)\, P(A \mid B) \tag{3.9}$$

Equating the two yields;

$$P(B)\, P(A|B) = P(A)\, P(B|A) \tag{3.10}$$

and therefore,

$$P(A|B) = P(A)\, \frac{P(B|A)}{P(B)} \tag{3.11}$$

In any problem, the probabilities of possible dependent states are calculated by the Bayes equation given above. In this equation, P(*A*) means the probability of input of the problem, P(*B*) indicates the probability of the possible

exit state and P(A | B) represents the probability of B output states versus the previous A input (Orhan and Adem, 2012). This equation is known as Bayes' Theorem and it is the basis of statistical inference. Arslan (2018) summarized the Bayes theorem as follows:

i. P (A | B) is the probability of A in class B.
ii. P (A) is the probability of A.
iii. P(B | A) is the probability of B in class A.
iv. P (B) is the probability of B.

The Bayesian classification method is based on the Bayes Theorem described above. Let $S = \{S_{1,}S_{2,}\dots S_m\}$ be a set of training data instances. Each S_i sample is assumed to be a n-dimensional vector of $\{x_1, x_2, \dots, x_n\}$. These x_i should be associated with properties $A_{1,}A_{2,\dots}A_n$, respectively (Gemici, 2012). There are k classes $C_{1,}C_{2,\dots}C_k$. When the classifier encounters an X sample, it will estimate that X belongs to the class with the highest posteriori probability (Leung, 2007). The probability of P (C_i | X) is computed by using the Bayes theorem as in equation (3.12).

$$P(C_i \mid X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)} \tag{3.12}$$

As P(X) is the same for each class, only multiplication of P ($X \mid C_i$) and P (C_i) needs to be maximum. This probability is calculated by equation 3.13.

$$P(C_i) = \frac{\#\text{ of sample in Class } C_i}{\#\text{ of samples in the dataset}} \tag{3.13}$$

P (X | C_i) is the probability of sample X given class C_i in large datasets. Therefore, NB classification is used for complex and large dataset. The probability P (X | C_i) is computed with the equation in 3.14.

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_t \mid C_i) \quad (3.14)$$

where P (x_t | C_i) is the probability of attribute value x_t for class C_i .

3.2.5. Support Vector Machines

Support Vector Machine was developed by Vladimir Vapnik and Alexey Chervonenkis at the end of the 1960s and is basically a machine learning method based on statistical learning theory (Akşehirli et al., 2013). It has been used frequently in data mining especially in classification problems in recent years. This method is considered to be a linear classifier for the solution of two-class problems, then generalized to solve linear or multi-class classification problems and has been widely used in the solution of these problems (Akşehirli et al., 2013). In general terms, SVMs are in the family of generalized linear models that perform classification and regression tasks using a linear combination of properties derived from variables (Tekin, 2014).

There are two important approaches in the development of SVM. The first is the linear approach and the second is the nonlinear approach. The linear approach is determination of the most optimal hyperplane which will optimally separate the two classes. Nonlinear approach can be expressed as the transformation of nonlinear separable classification problem to linear separable problem. Figure 3.10 shows the linearly separable binary classification problem, which is not a possibility of incorrect classification data (Zunic and Peter, 2018).

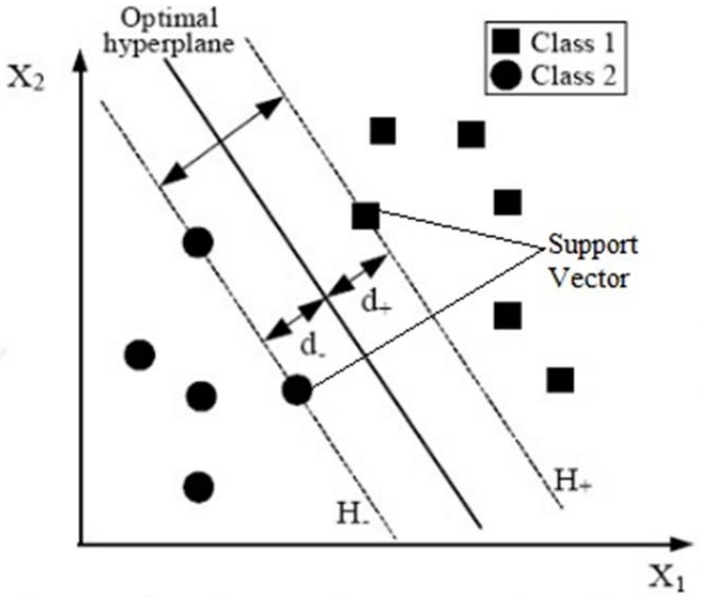

Figure 3.10. Linear SVM (Zunic and Peter, 2018)

SVM classifies by finding the separator hyperplane with the maximum margin between classes (Figure 3.10). The equation of the separating hyperplane is expressed in equation 3.15 where $w \in R^n$ is the weight vector, and b is the scalar constant (Güner and Çomak, 2011). Tuples $\{x_i, y_i\}$ can be represented as input property vectors and class label pairs, expressed as $i = 1,2, \dots N$, y_i belongs to (+1,1) and the classifier is in equation (3.16) (Zunic and Peter, 2018).

$$f(x) = w.x + b \tag{3.15}$$

$$y_i\ (w.x_i + b\ ) \geq 1, \qquad i = 1,2, \dots N \tag{3.16}$$

Referring to Figure 3.11, d + and d- is defined as margin. The margin will be maximized when d + = d- (Zunic and Peter, 2018). H + and H- hyperplanes are indicated by dashed lines. They are parallel hyperplanes to each other and to optimal hyperplane; and are located at equal distance from the optimal hyperplane.

There are no training examples exist between the H+ and H- hyperplanes. However, the existing training examples on the hyperplanes are Support Vectors and are examples of training closest to optimal hyperplane (Ayhan and Erdoğmuş, 2014). The distance between hyperplane H+ and H- is,

$$d_+ + d_- = \frac{2}{\|w\|} \tag{3.17}$$

In the nonlinear case, SVM will map the data in a lower dimensional space into a higher-dimensional space through kernel trick. The classifier is in equation (3.18) has $sgn\{\}$ which is the sign function, a_i is the Langrange multiplier, x_i is a training sample, x is a sample to be classified, K $(x_i.x)$ is the kernel. Kernel function includes linear function, polynomial function, and Radial Basis Function (RBF) (Yan and Jiao, 2016). Typical kernel formulation commonly used are listed in Table 3.4. (Zunic and Peter, 2018). In this study, polynomial kernel function is used when classifying with SVM in WEKA.

$$f(x) = sgn\{\sum_{i=1}^{n} a_i y_i (x_i.x) + b\} \tag{3.18}$$

Table 3.4. Kernel Function Formulation

Kernel	$K\ (x_i.x)$
Linear	$x^T.x_j$
Polynomial	$(x^T.x_j + 1)^d$
Gaussian RBF	$exp\left(\frac{-\|x - x_j\|^2}{2\sigma^2}\right)$

3.2.6. Random Forest

The Random Forest algorithm is a machine learning method that is used for classification and regression. Random Forest classifier consists of multiple decision tree collections. Tin Kam Ho created the first algorithm for random decision forests using random subspace method (Mishra and Ratha, 2016). Development of this algorithm was made by Leo Breiman and Adele Cutler (Mishra and Ratha, 2016). Nowadays, Random Forest algorithm is frequently preferred because it performs very well in classification. To create a tree with a RF classifier, you need 2 parameters. First, the number of variables used in each node to determine the best partition (m), the second is the number of trees (N) to be developed. The initial value of m is selected randomly by the user and the next value of m is increased or decreased according to the generalized errors (Ok et al., 2011). Breiman argues that when the number of variables m is equal to the square root of the number of variables M, it usually yields the closest results to optimal results. Samples are created from 2/3 of the training dataset. 1/3 of the training dataset is used to test for errors. GINI index is used to measure homogeneity of classes. Class heterogeneity increases as GINI index grows, class homogeneity increases as GINI index decreases. The branch succeeds when the GINI index of a child node is smaller than the GINI index of a parent node. When the GINI index reaches zero, the tree branching process ends. Finally, the class of the test sample is determined based on the estimation results obtained from N trees (Akar and Güngör, 2013). The GINI index is given in equation 3.19. Where T is whole dataset, p_j is the probability of class j in the dataset.

$$Gini(T) = 1 - \sum_{j=1}^{n}(p_j)^2 \tag{3.19}$$

3.2.7. K-fold Cross Validation

K-Fold Cross Validation is widely used in machine learning to evaluate performance of the models learned. It divides the dataset into k equal parts, the first k-1 parts are combined and used as a set of training and the remaining part is used as a test set (Maliha et al., 2018). This process is repeated k times.

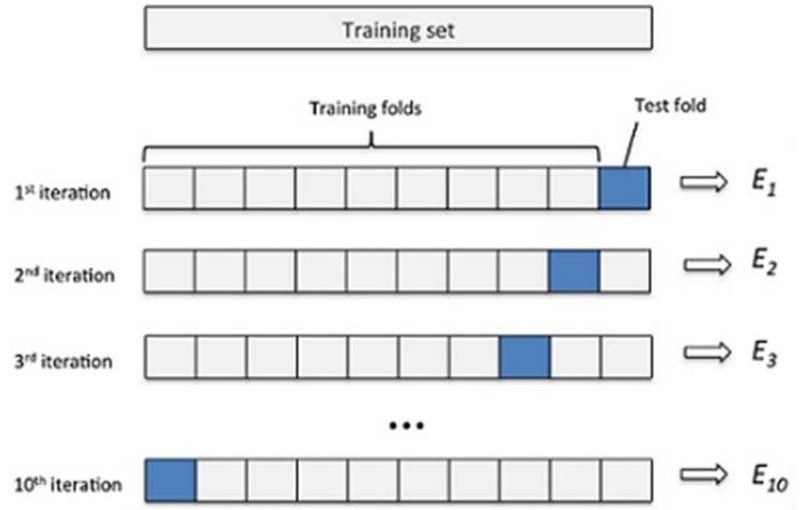

Figure 3.11. K-Fold Cross Validation

For 10-folds cross validation, the dataset is divided into 10 parts as shown in Figure 3.11. In each iteration, the blue painted area is the test set and the remaining part is the training set. At the end of each iteration, the performance measure values are added to the variable *E*. Then the arithmetic mean of the *E* value is computed for performance evaluation as in equation 3.20.

$$E = \frac{1}{10}\sum_{i=1}^{10} E_i \tag{3.20}$$

3.2.8. Feature Selection by Particle Swarm Optimization

In this section, the feature selection using PSO algorithm is explained. We create the initial swarm by assigning random position and speed values. Random position values are determined randomly by generating binary random numbers 0 and 1. Position of a particle is an *n* dimensional vector where *n* is the number of attributes in the dataset. Initial speed values are assigned to 0, because partitions are still constant and do not move. If position value is 1 for any i^{th} attribute this means that the attribute is selected. If position value is set to 0 then this attribute is

eliminated. If we assume that the dataset has 5 attributes in total, the position of any particle can be a vector with 5 elements. As an example, if position vector is [0, 1, 1, 0, 1], this means that the 2nd, 3rd and 5th attributes are selected, and the others are eliminated.

After computing binary position vector for each particle, the fitness value is calculated for each particle. To compute the fitness value of a particle, which shows a subset of features selected, a classifier is used. Therefore, the dataset is reduced by using the selected features by the particle, then classification is done, and the classification accuracy is taken as the fitness value of the particle. As the classifier, Naïve Bayes is used in this study, since it is one of the most accurate and fast classifiers.

While fitness values are obtained, *pbest* values are found for each particle. Then, the best *pbest* value is assigned as *gbest*. After computing *pbest* and *gbest* values, the speed value for each particle is calculated as in equation 3.5, and the position value is calculated as in equations 3.6 and 3.7. There is an important issue when making these calculations. While the particles move along the problem space, they are at risk of being away from the solution space. Therefore, the particle speeds should have a limit. When the speed of the particles is updated, the speed of each particle is limited in the range $[-V_{max}, V_{max}]$ where it is generally taken as [-10,10]. After this detail, all the position values are calculated, then the feature selection is performed by using the position value of each particle. This process is repeated until the termination criterion is met. Then the features selected in the *gbest* position are returned.

In this study, the classification is done by using the GaussianNB classification algorithm in the Python sklearn library. As described in the material and method section, NB classification algorithm is preferred because it is easy to implement, fast, and accurate. During the iterations, each new subset of the features are sent to the NB classifier. When applying the NB classification algorithm, k-fold cross validation is used to better compute the classification accuracy.

After completing the feature selection process, to test the real performance of the selected feature subset, test set which is the 30% of the dataset that is not employed during the feature selection process is used. For the test step, the training dataset which is reduced by using the selected features in the returned *gbest* is taken to learn the classification model. Then, the model performance is tested by using the test set. When testing the performance of the selected features, we use NB, SVM and RF classifiers in WEKA. Figure 3.12 shows the pseudo code of the attribute selection applied using the PSO algorithm and NB classification.

Procedure: Feature Selection by PSO algorithm and NB Classifier
Input: *mi* (Maximum number of iterations),
pn (Number of particles in the population).
Output: Best feature subset
Function newData(P_i)
P_{new} ← Select '0' values in P_i particle and assign '0' values to new data
F_{new} ← NBClassifier(P_{new}) // Compute fitness of the P_{new}
returns F_{new}
Function NBClassifier(P_{new})
D←Read Data file csv format
y← Set Dataset Class label
x← Drop(P_{new}) // Remove '0' features from Dataset D
A←GaussianNB(x,y)
return A //Accuracy

D //Dataset
A //Accuracy
F //Fitness Value (Accuracy)
S //Sigmoid function
Gbest.value ← 0 //initialize *Gbest* to 0.
Gbest.position ← [0, 0, ... ,0] // initially no feature is selected.

For *i* = 0 to *pn*
Begin
P[i].*position* ← Initialize particle position randomly by choosing binary random numbers.
P[i].*velocity* ← Initialize particle velocity to 0.
$Pbest_i.value$ ← **newData** (P[i].*position*)
Pbest$_i$.*position* ← P[i].*position*
If $Pbest_i.value$ > *Gbest.value*
Gbest.value=$Pbest_i.value$
Gbest.position = P[i].*position*
Endif
End
For *it* = 0 to *mi* //maximum number of iterations
For *i* = 0 to *pn* //number of particles in the swarm
P[i].*velocity* ←Calculate according to the equation 3.5
P[i].*velocity* ← Limit value to [-V_{max}, V_{max}]
P[i].*position*←Update according to the equation 3.7
$Pfitness_i$ ← **newData(***P*[i]**)**
If $Pfitness_i$> $Pbest_i.value$
$Pbest_i$.value= $Pfitness_i$
$Pbest_i$.*position*=*P*[i].*position*
Endif
If $Pfitness_i$> *Gbest.value*
Gbest.value=$Pfitness_i$
Gbest.position= *P*[i].*position*
Endif
End
End

Figure 3.12. Pseudo code for Feature Selection with PSO and NB

3.2.9. Mutation

The mutation used in genetic algorithms means the random change of a chromosomes. This change prevents individuals from being alike and can be one of the following (Şeker, 2009):

i. Inversion: It is the inversion of the value of a selected chromosome.
   1<u>1</u>0101 -> 1<u>0</u>0101
ii. Insertion: A new chromosome is added to the gene.
   110101 -> 11<u>0</u>0101
iii. Displacement: A chromosome leave from the gene.
   1<u>1</u>0101 -> 10101
iv. Reciprocal Exchange: It is the displacement of existing chromosomes.
   1<u>1</u>0101 -> 1001<u>1</u>1

3.2.10. Using PSO, NB and Mutation for Feature selection

Mutation is a method used to alter the current state of chromosomes in genetic algorithms. In this section, the hybrid method developed by combining PSO and mutation method for feature selection is explained. The purpose of mutating with PSO is to change the status of repetitive features throughout iterations. Changing the status of attributes is important to create different possibilities in search space. Inversion tecnique of the mutation is used as described in section 3.2.9. After 5 iterations, the selected attributes go to the mutation function as long as they are the same. The status of an attribute that is selected randomly from attributes is changed. The status of the randomly selected attribute is set to 0 if it is 1, and set to 1 if it is 0. The number of features to change depends on the size of the dataset.

3.2.11. Information Gain

Information gain (IG) measures the amount of information in bits about the class prediction. Concretely, it measures the expected reduction in entropy.

Entropy is expressed as a measure of uncertainty or unpredictability in a system (Budak, 2018). First the general entropy for the dataset *D* is calculated as in equation 3.21, then the entropy of the individual feature is calculated as in equation 3.22. At the end, the information gain of the features is calculated by taking the difference of the calculated values as in equation 3.23.

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2 p_i \tag{3.21}$$

where p_i is the probability of class i in the dataset, and m is the number of classes.

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{3.22}$$

where v is the number of distinct values that the attribute A can take, $|D|$ is the number of data instances in the whole dataset D, D_j is the subset of D where attribute A takes value equal to a_j, and $|D_j|$ is the number of data instances is D_j.

$$Gain(A) = Info(D) - Info_A(D) \tag{3.23}$$

Gain value is computed for all attributes in the dataset. Then attributes having the highest Gain values are chosen in the IG feature selection method.

3.2.12. Chi-Square

The Chi-Square test, which is based on whether the difference between observed and expected frequencies of a feature is significant, is one of the commonly used feature selection methods. It is checked whether the attribute in the dataset has an impact on the classification. As a result of the test, the features that

are not found to be related to the classification are removed from the dataset (Budak, 2018). The equations in 3.24 and 3.25 are computed by using a contingency table between the feature and class labels for the Chi-Square test. For each attribute, chi-square value is computed by using equations 3.24 and 3.25 where O_i is the observed frequency and E_i is the expected frequency for the attribute values for any class i. If the attribute has high chi-square value, it is important for the classification process.

$$\chi^2 = \sum_{i=1}^{m} \left(\frac{(O_i - E_i)^2}{E_i} \right) \tag{3.24}$$

$$E_i = \frac{row\ total \times column\ total}{grand\ total} \tag{3.25}$$

3.2.13. Relief

Relief is an example-based attribute filtering method proposed by Kira and Rendell (Budak, 2018). This method aims to find two features closest to a randomly selected instance. The closest sample of the same class is selected and is called the Hit sample (H). The closest missing sample from the different class is called the miss sample (M). Finally, the differences of the attributes between the samples are calculated as in equation 3.26 where W_i is the weight of the i^{th} property.

$$W_i = W_{i-1} - (x_i - nearHit_i)^2 + (x_i - nearMiss_i)^2 \tag{3.26}$$

3.2.14. Performance Measure

Confusion matrix is a method used to compare the test data with the estimation values and to measure the performance of the model. The confusion

matrix for a two-class data is given in Table 3.5. By using this table, accuracy of a classifier is computed as in equation 3.27 and it is used to measure the quality of the classification model learned.

Table 3.5. Confusion Matrix

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Classification Accuracy= Accuracy (ACC) is calculated as the number of all correct predictions by the total number of instances in the dataset.

$$Accuracy=TP+TN / (TP+TN+FP+FN) \quad (3.27)$$

4. RESULTS AND DISCUSSIONS

In this section, the experimental results of using PSO based feature selection for diagnosis of four different diseases are explained in detail. PSO based feature selection implementation was developed with the Python programming language (https://www.python.org/). PyCharm IDE for developing applications in Python and Numpy, pandas, sklearn Python packages were used to develop the application. After performing feature selection in Python, tests were performed using some classification algorithms found in WEKA Data Mining Tool. The hardware used has the following features: Windows 10 Pro Operating System, Intel (R) Core (TM) i7, 16 GB RAM and 3.60 GHZ processor. Many different steps have been applied during the experimental evaluation that are listed as follows;

i. Each dataset is separated into training and test sets by using WEKA.
ii. The best feature subset is determined for each dataset by using PSO algorithm and NB classification method in Python.
iii. After finding the best feature subsets, the reduced datasets are classified with NB in Python, SVM and RF classification methods in WEKA, and results are presented.
iv. Well-known feature selection methods that are information gain, chi-square, and relief from WEKA are also applied as feature selection. Their results are compared with that of PSO based method.

4.1. Preparing the Datasets for Analysis

In this thesis, a variety of datasets from different sources such as Kaggle (https://www.kaggle.com/), UCI etc. were investigated and healthcare related datasets were examined and selected. The datasets were selected according to below conditions:

i. Datasets that are developed for classification problems are selected.
ii. Balanced datasets with respect to class distribution are preferred.
iii. Datasets having sufficient number of attributes and samples are chosen.
iv. Datasets without missing values are taken.
v. Recently prepared new datasets are chosen.

At the end, BCCD, DRDD, SCADI and LSVT datasets with the above mentioned characteristics were selected from the UCI. After the dataset was selected, it was downloaded from UCI in csv format. Then, the csv files were converted into arff format by Weka Arffviewer, and the datasets were divided into train and test datasets for use in the study.

There are different methods to divide the data into training and test datasets. Dataset can be separated manually, by programming or by using WEKA without disturbing the homogeneity of classification. In this study, the datasets were divided into training and testing with the help of WEKA software. WEKA's filters feature was used when to split data. Weka-> Filter-> Supervised-> Resample is selected from WEKA's filter option. As in Figure 4.1, the percentage of the training set is entered in the “sampleSizePercent” value, “noReplacement” value is made “True”. Then we click Ok and leave the screen and "Apply" button is pressed. Thus, 70 percent of the dataset is chosen as the training set. For collecting test data, "Undo" button is pressed from the same screen and dataset returns to 100 percent. Then, the screen in Figure 4.1 is opened again. This time, only the "invertSelection" option is set to "True" to get the remaining 30 percent. Thus, 30 percent test data is generated.

Two datasets are also divided by the WEKA software in the same way. All of the datasets are divided into 70:30 percent. Table 4.1 shows the numbers of instances in the datasets in the training and test sets. Thus, our data are split without compromising homogeneity of classification. After the datasets are divided into training and test sets, the second phase is started.

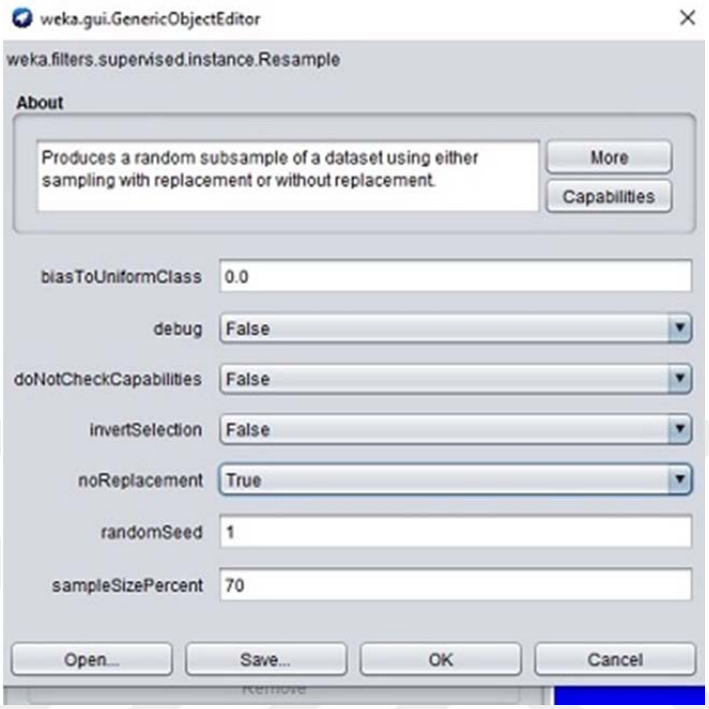

Figure 4.1. Weka Filters Supervised Instance Resample Editor

Table 4.1. Train and Test sets of the Datasets

Dataset name	# of train instances	# of test instances
Breast Cancer Coimbra (BCCD)	80	36
Diabetic Retinopathy Debrecen Dataset (DRDD)	805	346
Self-Care Activities Dataset (SCADI)	46	24
Lee Silverman Voice Treatment (LSVT)	87	39

In the below paragraphs, we show that the datasets are divided into train and test sets without disturbing the homogeneity of classification by performing a simple experiment. We apply NB classifier from Python and WEKA to the datasets in two ways: first we classify each dataset by combining train and test sets, and

applying 10 fold cross validation. Secondly, we learn classifier with the training set, and test the model with the test set. The results of this experiment for the BCCD, DRDD, SCADI and LSVT datasets are shown in Table 4.2 for the NB classifier in WEKA and Python. 100% refers to the original dataset (i.e., the case when we apply 10 fold cross validation) and 70%-30% refers to the training-test datasets obtained with the WEKA sample filter option.

According to Table 4.2, when NB classifier is used for raw data (100% case) and training-test partitioned data, the classification accuracy does not changed so much. In addition, the results show that the WEKA NB classifier, in general, yields better results than that of Python. This may be because the algorithm in WEKA uses better smoothing methods when computing probabilities. These values, which are shown in the Table 4.2, are then compared with the results when feature selection is applied.

Table 4.2. Classification Accuracy for Raw Set and Train-Test Sets for All Datasets

Dataset	Classification Accuracy of NB			
	WEKA 100%	WEKA 70%	Python 100%	Python 70%
BCCD	60.3%	60%	54.7%	55%
DRDD	56.8%	58.1%	59%	59.5%
SCADI	82.8%	84.7%	78.5%	79.5%
LSVT	56.3%	54%	42.8%	54.2%

4.2. Parameters of PSO and NB

In this section, the parameters used in selecting features for PSO algorithm and NB classification algorithm are discussed. Table 4.3 presents the parameter values used PSO and NB for feature selection and fitness function evaluation. The

parameter values given in Table 4.3 are tested and best parameter values are tried to be determined. Almost all parameters except the k-fold parameters belong to the PSO, k-fold parameters show how the k-fold cross validation is used in NB classification. The size of the position vector is taken as equal to the number of attributes of the dataset.

Table 4.3. Parameter Values of PSO and k-Fold Cross Validation of NB

Parameter	BCCD	DRDD	SCADI	LSVT
$c_1 = c_2$	[0.5-2]	[0.5-2]	[0.5-2]	[0.5-2]
$rand_1 = rand_2$	[0-1]	[0-1]	[0-1]	[0-1]
w (Inertia weight)	0.995	0.995	0.995	0.995
Iteration Number	500	500	500	500
Population Size	[50-200]	[50-200]	[50-200]	[50-200]
k-fold(n_split)	10	10	10	10
k-fold(shuffle)	True	True	True	True
Size of Position Vector	9	19	205	310
$[-V_{max}, V_{max}]$	[-5,5]	[-5,5]	[-5,5]	[-5,5]

In the next subsection, the performance of the PSO based feature selection system is presented for each dataset, for various values of the number of iterations, population size, c_1 and c_2 parameters.

4.3. Finding Best Feature Subset for All Datasets

In this section, the results of the feature selection experiments for the BCCD, DRDD, SCADI and LSVT datasets obtained from UCI are explained. The datasets are not preprocessed, because the datasets do not contain any missing data. Table 4.4 shows the range of the variables used for feature selection method. In this section, the results obtained by changing the values of these parameters are

evaluated. First of all, the results obtained by changing only the shuffle parameter for k-fold cross validation to 'False' or 'True' are shown in Table 4.5. The shuffle parameter is included in the kfold function in the Python numpy package. If shuffle is set to True, data instances in the dataset are shuffled then k-fold cross validation is applied, if shuffle is set to False, dataset is divided into *k* parts without changing the order of data instances and the training and test sets are created in order.

Table 4.4. Changeable PSO Parameters and Their Values for All Datasets

Parameter	Range
$c_1 = c_2$	[0.5-2]
Population Size (PS)	[50-200]

Table 4.5. Python PSO+ NB Accuracy with Shuffle False or True

Dataset	# Accuracy / Classification Time PS=200, IN=500, $c_1 = c_2$=0.5	
	kfold(shuffle=True) /Time	kfold(shuffle=False) / Time
BCCD	77.5% / 14.9min	61.2% / 15min
DRDD	66.8% / 32min	66% / 31min
SCADI	90% / 39.2min	84% / 38min
LSVT	60% / 64.8min	56% / 64min

The results presented in Table 4.5 are computed when population size (PS) is set to 200, number of iterations (IN) is equal to 500, and c_1 and c_2 parameters are equal to 0.5. As shown in Table 4.5, when shuffle=False, the feature selection process runs slightly faster, however it decreases classification accuracy. Therefore, shuffle value is considered as "True" to achieve better results when selecting the features in the subsequent experiments in this thesis.

In Table 4.6 to 4.15, performance of the PSO based feature selector is presented for all datasets for varying population size (PN), number of iterations (IN), c_1 and c_2 parameters. The first value given in each cell shows the classification accuracy when PSO based feature selector is applied, the second value is the running time of the PSO.

Table 4.6. Python PSO+ NB Accuracy for PN=50, IN=500, c_1= c_2=0.5

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	72.5% /4.7min	66.3%/9.5min	90%/11min	59.4%/17.6min

Table 4.7. Python PSO+ NB Accuracy for PN=50, IN=500, c_1= c_2=1

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	75%/4.7min	66.4%/9.3min	90%/11.2min	61.1%/17.5min

Table 4.8. Python PSO+ NB Accuracy for PN=50, IN=500, c_1= c_2=1.5

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	71.2%/4.4min	66.5%/8.8min	90%/10.6min	60.4%/16.7min

Table 4.9. Python PSO+ NB Accuracy for PN=50, IN=500, c_1= c_2=2

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	73.7%/4.65min	66.4%/9.5min	90%/11.2min	60.8%/17min

Table 4.10. Python PSO+ NB Accuracy for PN=100, IN=500, c_1= c_2=0.5

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	73.7%/9.6min	66.9%/18.6min	90%/22.3min	61.1%/35.5min

Table 4. 11. Python PSO+ NB Accuracy for PN=100, IN=500, c_1= c_2=1

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	75%/9.7min	66.8%/19min	90%/22.8min	60.2%/35.4min

Table 4. 12. Python PSO+ NB Accuracy for PN=100, IN=500, c_1= c_2=1.5

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	73.7%/10.9min	66.7%/19.7min	90%/23.2min	60.9%/35.7min

Table 4. 13. Python PSO+ NB Accuracy for PN=100, IN=500, c_1= c_2=2

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	73.7%/8.8min	66.8%/18.4min	90%/21.6min	60.9%/

Table 4. 14. Python PSO+ NB Accuracy for PN=200, IN=500, c_1= c_2=1

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	75%/15.5min	67%/33min	90%/39.2min	60%/65min

Table 4. 15. Python PSO+ NB Accuracy for PN=200, IN=500, c_1= c_2=1.5

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	75%/17.7min	66.7%/36.2min	90%/43.2min	61.1%/68.7min

Table 4. 16. Python PSO+ NB Accuracy for PN=200, IN=500, c_1= c_2=2

Best Results	BCCD	DRDD	SCADI	LSVT
Accuracy Rate	73.7%/17.6min	66.3%/35.9min	90%/42min	61.4%/66.5min

From Table 4.6 to 4.16, the number of iterations is kept constant at 500 and PN value changed between 50 to 200, and $c_1 = c_2$ value changed between 0.5 to 2. According to the results in these tables, BCCD dataset is classified with the highest accuracy which is equal to 77.5%, when PN = 200 and $c_1 = c_2 = 0.5$. DRDD dataset has the highest classification accuracy with 67%, when PN = 200 and $c_1 = c_2 = 1$. SCADI dataset is classified with the highest accuracy which is equal to 90.1% when PN = 200 and $c_1 = c_2 = 1.5$. LSVT dataset has the highest classification accuracy that is equal to 61.4%, when PN = 200 and $c_1 = c_2 = 2$. Table 4.17 shows the best accuracy results obtained using PSO + NB with best parameter values, and Table 4.18 shows the selected attributes that give the highest classification accuracy for the datasets.

Table 4. 17. Best Classification Accuracy with PSO + NB for all Datasets

Best Results	BCCD	DRDD	SCADI	LSVT
Optimum Parameter Values	PN = 200, $c_1 = c_2 = 0.5$	PN = 200, $c_1 = c_2 = 1$	PN = 200, $c_1 = c_2 = 1.5$	PN = 200, $c_1 = c_2 = 2$
Accuracy / Running Time of PSO	77.5%/14.9min	67%/33min	90%/42min	61.4%/66.5min

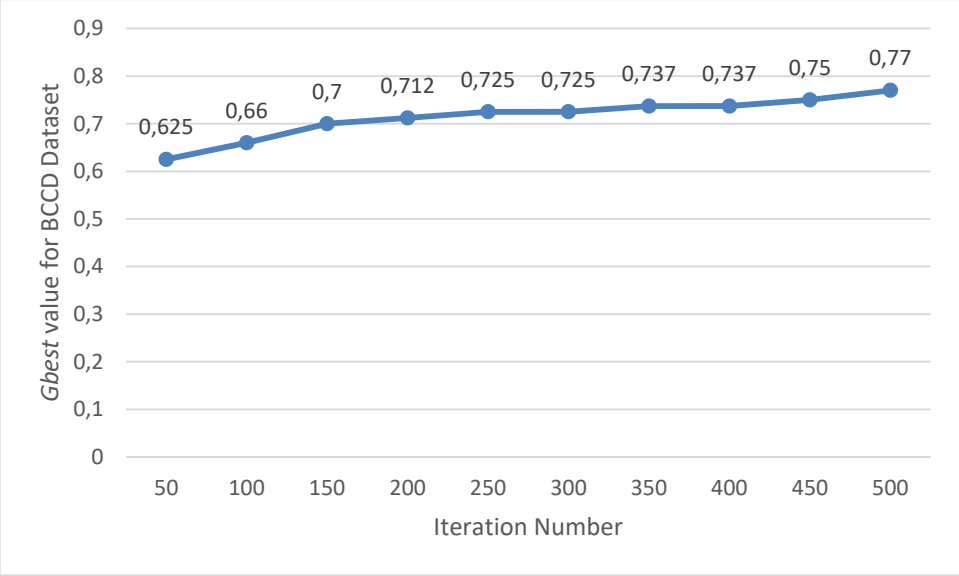

Figure 4.2. Gbest value based on the number of iterations for the BCCD dataset

Figure 4.2 shows the *gbest* value graph for the BCCD dataset with respect to the number of iterations when the best parameter settings for the PSO is used. PSO algorithm aims to achieve the best *gbest* value at the end of each iteration, so the *gbest* value usually tends to increase. As shown in Figure 4.2, during iterations, it is observed that the *gbest* value generally increases or at least remains constant in some places. Therefore Figure 4.2 shows that the *gbest* value converges through an optimum value as the iterations pass. Similar convergence of *gbest* values of the other datasets DRDD, SCADI, and LSVT, are also observed, and to save space they are not presented here.

Table 4. 18. Selected Features with PSO + NB from all Datasets

Best Results	Selected Features	Number of Selected Features
BCCD	['Age','BMI','Glucose','Resistin']	4
DRDD	[0,2,3,4,8,14,17,18]	8
SCADI	[0,1,2,4,5,6,9,10,11,12,13,14,15,18,19,20,21,22,24,25,26,27,28,29,31,32,33,35,42,43,44,45,47,50,51,57,58,63,64,65,67,71,73,76,77,80,84,85,87,88,89,90,91,92,93,98,100,101,103,107,109,111,112,113,114,115,118,120,121,123,125,126,127,128,132,133,134,137,142,145,149,150,151,155,156,157,159,160,161,162,165,166,169,170,172,173,174,175,177,179,182,184,185,186,188,191,194,204]	110
LSVT	[3,5,6,9,10,12,13,14,15,18,20,21,22,25,26,28,30,33,36,37,38,39,41,42,46,48,50,51,53,55,57,58,59,64,67,69,70,72,74,76,80,81,83,86,87,90,91,92,93,95,96,97,99,101,102,104,108,109,117,118,122,125,127,129,132,133,134,135,136,137,143,145,146,147,148,153,154,156,158,160,161,164,166,167,169,170,172,173,174,176,177,178,181,182,184,185,187,188,189,190,192,194,203,204,205,206,208,209,211,212,214,215,217,219,220,221,223,226,229,230,232,2322,236,237,238,241,242,243,244,248,251,252,253,254,256,257,259,263,264,266,268,275,278,279,284,287,290,291,294,300,302,303,305,306,308,309]	156

Table 4.17 shows the best classification accuracy values for all datasets and Table 4.18 shows the best features that provide this classification accuracy. Table 4.19 compares the classification accuracies for all datasets with and without feature selection. These results are obtained when NB classifier in Python is applied to i) the original dataset with 10 fold cross validation (this is the "without FS" case), and ii) the reduced training set, which is 70% of the original dataset, including only the selected features with the PSO. 10 fold cross validation is also applied to the reduced dataset. As it can be seen from Table 4.17, after the feature selection, the classification accuracy increases 22.8% for BCCD and 8% for DRDD, 12.5% for SCADI and 18.6% for LSVT. The feature selection method with PSO and NB classification improves classification accuracy in all datasets. The running time changes according to the number of iterations and number of individuals in the population in the PSO.

Table 4. 19. Classification Accuracy with and without Feature Selection

	BCCD		DRDD		SCADI		LSVT	
	Without FS	With FS	Without FS	With FS	Without FS	With FS	Without FS	With FS
Accuracy of NB in Python	54.7%	77.5%	59%	67%	78.5%	90%	42.8%	61.4%

4.4. Effects of Feature Selection for Various Classifiers

In this experiment, effects of feature selection method are tested by using various classifiers. First of all, classifier performance is measured without feature selection. Each classifier is learned from the training set which is the randomly chosen 70% of the dataset. Then, the classifier is tested with the test set which is the remaining 30% of the dataset. Secondly, by using the training set, PSO based feature selection is applied and the best feature set is determined. After that both

the training and test sets are reduced by using the selected features and classifier is learned from the training set, and tested with the test set. These processes are repeated for the NB, SVM, and RF classifiers in Python and WEKA environment for the four datasets.

In Table 4.20, experiments are performed by using the NB classifier of Python. According to Table 4.20, when feature selection is applied, the classification accuracies are increased 13.9% for BCCD, 3.8% for DRDD, 4.2% for SCADI, and 2.5% for LSVT datasets. The highest accuracy rate increase is observed in the BCCD dataset after feature selection.

Table 4. 20. Classification Accuracies for all Datasets with NB Classifier in Python

Best Results	BCCD		DRDD		SCADI		LSVT	
Accuracy Rate	Without FS	With FS	Without FS	With FS	Without FS	With FS	Without FS	With FS
Python	66.6%	80.5%	56.6%	60.4%	79.1%	83.3%	48.7%	51.2%

We test the performance of the feature selection method by using NB, SVM and RF classification algorithms from WEKA software. Previously, this experiment was performed with NB classifier in Python. In Table 4.21, the classification accuracy for NB classifier in WEKA using BCCD, DRDD, SCADI and, LSVT datasets are presented.

Table 4.21. Classification Accuracies for all Datasets with NB Classifier in WEKA

Best Results	BCCD		DRDD		SCADI		LSVT	
	Without FS	With FS	Without FS	With FS	Without FS	With FS	Without FS	With FS
Accuracy Rate	63.8%	80.5%	55.4%	60.1%	79.1%	79.1%	56.4%	53.8%

Default parameters of NB algorithm in WEKA are used for this experiment. The classification accuracy for BCCD dataset increases 16.7% after feature selection. In DRDD datasets, there is 4.7% increase in the accuracy after the feature selection. There is no change in the accuracy value of the SCADI dataset. However, in Table 4.20 the accuracy of the NB in Python for the SCADI dataset is higher than the accuracy found by NB in WEKA. In LSVT, there is 2.5% decrease in the accuracy after feature selection. It may be due to the fact that differences between the implementations of the NB classifiers in Python and WEKA. Also, to find best feature subset, NB in Python is used, however to test the performance of the selected features NB in WEKA is applied. When Table 4.20 and 4.21 are compared we observe that in majority of the cases feature selection improves classification accuracy, and there is no huge difference between the accuracies of the two NB implementations.

Table 4.22. Classification Accuracies for all Datasets with SVM Classifier in WEKA

Best Results	BCCD		DRDD		SCADI		LSVT	
	Without FS	With FS	Without FS	With FS	Without FS	With FS	Without FS	With FS
Accuracy Rate	69.4%	61.1%	66.7%	63.5%	79.1%	79.1%	71.7%	76.9%

In Table 4.22, the FS method is tested with the SVM classifier from WEKA. SVM classifier is called as SMO in WEKA. Default parameters of SVM algorithm in WEKA are used for testing. There is 8.3% accuracy decrease in BCCD dataset, and 3.2% accuracy decrease in DRDD dataset. There is no change in the accuracy value of the SCADI dataset. However, accuracy increases 5.2% in the LSVT dataset. These results occur because the feature subsets are selected with the NB classifier in Python, however, their performance is tested with the SVM classifier. So, we can conclude that the PSO selected the best feature subset for the

NB classifier, and these features are not suitable with the SVM classifier. When Table 4.21 and 4.22 are compared for without feature selection case, it is observed that SVM classifier generally has higher classification accuracy than the NB classifier.

Table 4. 23. Classification Accuracies for all Datasets with RF Classifier in WEKA

Best Results	BCCD		DRDD		SCADI		LSVT	
	Without FS	With FS	Without FS	With FS	Without FS	With FS	Without FS	With FS
Accuracy Rate	77.7%	80.5%	65%	69.1%	75%	79.1%	82%	87.2%

The same experiment is repeated with the RF classifier in WEKA and the results are listed in Table 4.23. Default parameters of RF algorithm in WEKA are used for testing. As shown in Table 4.23, the classification accuracy increases in all datasets after feature selection. RF algorithm has been widely used in data mining recently, because it performs well. These results showed successful performance of RF classification algorithm for all datasets.

Table 4.24. Performance of Classifiers for All Datasets

Best Results	BCCD		DRDD		SCADI		LSVT	
Classifier	Without FS	With FS	Without FS	With FS	Without FS	With FS	Without FS	With FS
NB-Python	66.6%	**80.5**%	56.6%	60.4%	79.1%	**83.3**%	48.7%	51.2%
NB-Weka	63.8%	**80.5**%	55.4%	60.1%	79.1%	79.1%	56.4%	53.8%
SVM-Weka	69.4%	61.1%	66.7%	63.5%	79.1%	79.1%	71.7%	76.9%
RF-Weka	77.7%	**80.5**%	65%	**69.1**%	75%	79.1%	82%	**87.2**%

In Table 4.24, results of all classifiers for the datasets are summarized and the best accuracy results for each dataset are written in boldface. Table 4.24 shows that PSO based feature selection have positively affects the accuracy of classification for all datasets. Testing of the selected features with SVM in WEKA does not yield positive results. In addition, the accuracy of the RF classifier increases after the feature selection for all datasets. Except for the SCADI dataset, RF has the best performance. Although RF uses features selected for the NB classifier, it has the best accuracy.

Table 4.25. Elapsed Time to Test Model on Supplied Test Set for All Datasets

Best Results	BCCD		DRDD		SCADI		LSVT	
Classifier	Without FS	With FS	Without FS	With FS	Without FS	With FS	Without FS	With FS
NB-Python	0.03s	0.01s	0.03s	0.02s	0.03s	0.02s	0.06s	0.05s
NB-Weka	0s	0s	0s	0s	0.01s	0s	0.01s	0s
SVM-Weka	0.01s	0s	0.02s	0.01s	0.04s	0.04s	0.01s	0 s
RF-Weka	0.02s	0.01s	0.21s	0.17s	0.01s	0.01s	0.05s	0.03s

Table 4.25 shows elapsed time to test model on supplied test set for BCCD, DRDD, SCADI and, LSVT with and without feature selection. Generally, classifier run times are shortened after feature selection.

4.5. Effect of Using Mutation in the PSO-based Feature Selector

In the previous experiments we observed that the PSO-based feature selection method generates the same set of features over multiple iterations. This may cause the PSO to be trapped into a local optimum. In order to get rid of the local optimum, we propose to include mutation operation from the Genetic Algorithms (GA) into the PSO during the position computation of the particles. If

at the end of the *n* consecutive iterations, the selected attributes are the same, mutation is performed to change the status of the attributes. We determine *n* as 5 experimentally. The number of attributes to be mutated is proportional to the number of attributes in the dataset. For example, 1 attribute in the BCCD dataset, 3 attributes in the DRDD dataset, 30 attributes in the SCADI dataset, and 60 attributes in the LSVT dataset are mutated.

The best parameter values of the PSO + NB feature selector are also used in the mutated version of the feature selector which we call as PSO + NB + Mutation feature selector. These parameters are given in Table 4.17. The classification accuracy results for both PSO + NB and PSO + NB + Mutation feature selection methods for all datasets are compared in Table 4.27. The experimental results show that except for the 1.4% increase in the DRDD dataset, the results with the mutation operator are not successful. This result may occur due to the fact that when mutation is applied particles may move far away from their best position. More experimental analysis may be done for mutation operation as future work.

Table 4.26. Comparison of Supplied test Accuracy of PSO+NB and PSO + NB + Mutation Feature Selection Methods for all Datasets

	BCCD		DRDD		SCADI		LSVT	
Classifier	PSO+ NB	PSO+ NB+M	PSO+ NB	PSO+ NB+M	PSO+ NB	PSO+ NB+M	PSO+ NB	PSO+ NB+M
Python-NB	80.5%	72.2%	60.4%	61.8%	83.3%	70.8%	51.2%	48.7%

4.6. Comparison with Information Gain Feature Selector

In this experiment, we compare the performance of the PSO-based feature selector with the well-known Information Gain (IG) feature selection method. To select features with IG, features with a gain value above 0 are selected by using the

Info Gain attribute selection module in WEKA. Attributes having gain value less than or equal to 0 are eliminated from the dataset.

Table 4.27 shows the number of features selected with IG. In Table 4.28, PSO + NB and IG feature selectors are compared in terms of classification accuracy values. In the BCCD dataset, feature selection with PSO + NB gives better results than feature selection with IG. For the other datasets, when NB and RF classifiers are used, PSO + NB feature selector is more successful. For the SVM classifier, IG feature selector may have slightly more accurate results for DRDD and LSVT datasets. Table 4.28 shows that the overall performance of PSO + NB feature selection method is better than that of IG method.

Table 4.27. Number of Selected Features After Applying IG Based Feature Selection

Dataset	**# selected features**
BCCD	1
DRDD	10
SCADI	132
LSVT	84

Table 4. 28. Comparison of PSO+NB and IG methods using classifiers in WEKA

Best Results	BCCD		DRDD		SCADI		LSVT	
Accuracy Rate	PSO+NB	IG	PSO+NB	IG	PSO+NB	IG	PSO+NB	IG
NB	**80.5%**	66.6%	60.1%	56.3%	**79.1%**	**79.1%**	53.8%	53.8%
SVM	63.5%	55.5%	63.5%	64.1%	**79.1%**	**79.1%**	74.3%	82%
RF	**80.5%**	61.1%	**69.1%**	65.6%	75%	**79.1%**	**87.2%**	79.4%

4.7. Comparison with Chi-Square Feature Selector

In this experiment, PSO + NB feature selector is compared with the Chi-square (CHI2) feature selection method. For CHI2 feature selection method WEKA is used. When selecting the features with the CHI2 method, features having chi-square value greater than 0 are selected.

Table 4.29 shows the number of attributes selected with CHI2 method. In Table 4.30, classification accuracy values of PSO + NB and CHI2 methods are compared. In the selection of attributes with IG and CHI2, the numbers of attributes selected are the same. It also appears to be similar in classification accuracy with IG and CHI2. Comparing Table 4.28 and 4.30, it is seen that IG and CHI2 methods give similar accuracy results. In addition, feature selection with PSO + NB method is more successful than the CHI2 method.

Table 4.29. Number of Selected Features After Applying IG Based Feature Selection

Dataset	**# selected features**
BCCD	1
DRDD	10
SCADI	132
LSVT	84

Table 4.30. Comparison of PSO+NB and CHI2 methods using classifiers in WEKA

	BCCD		DRDD		SCADI		LSVT	
Classifiers	PSO+NB	CHI2	PSO+NB	CHI2	PSO+NB	CHI2	PSO+NB	CHI2
NB	**80.5%**	66.6%	60.1%	56.3%	**79.1%**	**79.1%**	53.8%	53.8%
SVM	63.5%	55.5%	63.5%	64.1%	**79.1%**	**79.1%**	74.3%	82%
RF	**80.5%**	61.1%	**69.1**%	69.07%	75%	**79.1%**	**87.2%**	84.6%

4.8. Comparison with Relief Feature Selector

In this experiment, feature selection is made by using Relief method in WEKA, and results are compared with that of PSO + NB feature selection method. When selecting the features with Relief method, features having scores that are greater than 0 are selected. According to Table 4.31, more features are selected with the Relief method than the IG and CHI2 methods. When Table 4.32 is examined, both PSO + NB and Relief feature selection methods have similar accuracy values. In general RF classifier has the best accuracy.

Table 4.31. Number of Selected Features After Applying Relief Based Feature Selection

Dataset	**# selected features**
BCCD	6
DRDD	16
SCADI	125
LSVT	257

Table 4.32. Comparison of PSO+NB and Relief methods using classifiers in WEKA

	BCCD		DRDD		SCADI		LSVT	
Classifiers	PSO+NB	Relief	PSO+NB	Relief	PSO+NB	Relief	PSO+NB	Relief
NB	80.5%	61.1%	60.1%	55.4%	**79.1%**	**79.1%**	53.8%	56.4%
SVM	63.5%	69.4%	63.5%	65.8%	**79.1%**	**79.1%**	74.3%	74.3%
RF	80.5%	**86.1%**	**69.1%**	65.6%	75%	**79.1%**	**87.2%**	84.6%

4.9. Comparison with Previous Studies and Discussions

In this section, the previous studies on the datasets used in this thesis are examined and compared with our results.

Pereira *et al.* (2018) have studied BCCD data. ROC analysis and Gini Coefficient are used to select the attributes that are important for the diagnosis of breast cancer. They used various classification algorithms such as SVM, RF and LR for classification of the BCCD dataset. The SVM algorithm gives the best accuracy value using Glucose, Resistin, Age and BMI attributes. In comparison with this study, they used different methods for feature selection and they selected BMI attributes differently from our study. Their best accuracy is achieved with 92% using SVM, and in our study it is 80.5% with RF and NB.

Li and Chen (2018) examined breast cancer datasets, one of them is the BCCD dataset. No feature selection has been made by using any optimization algorithm in this study. Data are compared using five different classification algorithms such as DT, SVM, RF, LR and NN. Common classification algorithms with our study are SVM, NB and RF. As shown in Table 4.33, while their studies achieved 71% success in SVM, our study achieved 80.5% success in RF and NB. Our study achieved higher accuracy by selecting features.

Table 4.33. Comparison of classification accuracies of Li and Chen (2018) with our study for BCCD dataset

	DT	SVM	RF	LR	NN	NB
Li and Chen (2018)	0.686	0.714	0.743	0.657	0.600	-
This Study	0.694	0.611	0.805	-	0.805	0.805

Aslan *et al.* (2018) analyzed different machine learning methods on BCCD dataset. These methods are ANN, ELM, k-NN and SVM. In ANN, ELM, k-NN and SVM methods, hyperparameters with the least error are determined by using Hyperparameter optimization technique. With the Hyperparameter optimization technique, ELM gives the best classification accuracy with 80%. They achieved 73.5% accuracy in SVM algorithm and achieved a better result than our results. Our study achieved 80.5% success in ANN.

Table 4.34. Comparison of classification accuracies of Aslan et al. (2018) with our study for BCCD dataset

	ANN	ELM	k-NN	SVM
Aslan et al.(2018)	79.43	80	77.5	73.5
This Study	80.5	-	86.1	61.1

Antal and Hajdu (2014) have analyzed three different datasets, one of which is the DRDD dataset. They used nine different classification algorithms in their analysis. The preferred classifiers in the study are: NB, SL, IBK, Bagging, DT, IMC, J48, RF, RT. No feature selection technique was used in the study, but classification was made with different sample sizes and results were evaluated. NB and RF algorithms were used in both studies. In Table 4.35, the results of the studies with 400 and 1200 samples are given and compared with our study. The accuracy value of NB, SL IBK, and IMC classifications are higher in our study than their study.

Table 4.35. Comparison of classification accuracies of Antal and Hadju (2014) with our study for DRDD dataset

	RF	NB	SL	IBK	DT	IMC	J48	RT	Bagging
Antal and Hadju (2014)(with 400 sample)	0.703	0.474	0.712	0.592	0.651	0.345	0.64	0.564	0.652
Antal and Hadju (2014) (with1200 sample)	0.665	0.507	0.711	0.614	0.617	0.368	0.64	0.617	0.661
This Study	0.690	0.604	0.708	0.630	0.615	0.532	0.63	0.589	0.635

Ramaswamy and Savitha (2016) have measured the success of classification of various datasets by machine learning algorithms using the R programming language, and one of these datasets is the DRDD dataset. No feature selection technique was used in the studies. They used eight different classification

algorithms in their analysis, only SVM, RF and NB algorithms are compared. Table 4.36 shows that accuracy of their studies is successful than our study.

Table 4.36. Comparison of classification accuracies of Ramaswawy and Savita (2016) with our study for DRDD dataset

	RF	SVM	NB	C5.0
Ramaswawy and Savita (2016)	0.921	0.734	0.629	0.854
This Study	0.690	0.635	0.604	0.632

Maliha, Tareque and Roy (2018) analyzed DRDD dataset with four different classification algorithms that are SVM, KNN, RF, NNET. Each classification algorithm is modeled with 10-fold cross validation using training dataset. SVM and RF algorithms were used in both our and their studies. As shown in Table 4.37, while our study is successful for RF and SVM, their results are more successful for KNN and NNET.

Table 4.37. Comparison of classification accuracies of Maliha, Tareque and Roy (2018) with our study for DRDD dataset

	RF	SVM	KNN	NNET
Maliha, Tareque and Roy (2018)	0.636	0.570	0.645	0.753
This Study	0.690	0.635	0.635	0.716

Mohammadian, Karsaz and Roshan (2017) also used DRDD dataset. Nine classifier algorithms are used for comparative analysis with Python Numpy package. Some methods have not used tunable parameters such as NB, QDA and Adaptive Boosting (AB) and however some other algorithms used tunable parameters. Each classification technique was optimized based on optimum parameters. The Gauss Process algorithm provided the best accuracy with 87% by

selecting the optimum parameters. After the Gaussian Process, the second algorithm with the best accuracy is SVM with 84%.

Table 4.38. Comparison of classification accuracies of Mohammadian, Karsaz and Roshan (2017) with our study for DRDD dataset

	RF	SVM	NB	DT	KNN	MLP	AB
Mohammadian, Karsaz and Roshan (2017)	0.802	0.844	0.728	0.731	0.760	0.812	0.831
This Study	0.690	0.635	0.601	0.632	0.635	0.716	0.618

Buarque *et al.* (2017) examined DRDD dataset obtained from UCI. They used two different methods, PM and PCA for dimensionality reduction. The details of these techniques are described in the related works section. These methods have been tested on nine different datasets, by using four different classification methods, we only compare DRDD results for NB, DT and KNN algorithms. In Table 4.39, feature selection with PSO yields better results than PM and PCA methods.

Table 4.39. Comparison of classification accuracies of Buarque et al. (2017) with our study for DRDD dataset

	NB	DT	KNN
Buarque et al. (2017) (Raw Data)	55.8%	61.0%	61.7%
Buarque et al. (2017) (PM)	62.7%	63.2%	62.5%
Buarque et al. (2017) (PCA)	58.2%	58.8%	58.5%
This Study PSO+NB	60.4%	63.2%	63.5%

Zarchi, Fatemi Bushehri, and Dehghanizadeh (2018) have analyzed SCADI dataset from UCI. They used ANN classifier and DT algorithm in their study. The best accuracy found belongs to ANN expert system which is equal to 83.1%.

According to Table 4.40, the results obtained by ANN and PSO + NB are not very different from each other.

Table 4.40. Comparison of classification accuracies of Zarchi, Fatemi Bushehri, and Dehghanizadeh (2018) with our study for SCADI dataset

	ANN
Zarchi, Fatemi Bushehri, and Dehghanizadeh (2018)	83.1%
This Study	79.1%

El Moudden, Ouzir, and ElBernoussi (2017) have applied feature selection and extraction on LSVT dataset obtained from UCI machine learning repository. They used PCA + LR and PCA + C5.0 method to make dimensionality reduction on the dataset. Using with these methods, 90% accuracy of PCA + LR and 91.2% accuracy of PCA + C5.0 are achieved.

Table 4.41. Comparison of classification accuracies of El Moudden, Ouzir, and ElBernoussi (2017) with our study for LSVT dataset

	LR	C5.0	NB	RF
El Moudden, Ouzir, and ElBernoussi (2017) (PCA)	90%	91.2%	-	-
This Study PSO	-	74.3%	53.8%	87.2%

It has been observed that there are some differences between the results of this study and the previous studies. Possible reasons of theses differences may be; the used parameter setting methods, optimization algorithms, classification algorithms, statistical methods, programming language in which the application is made, etc. However, for majority of the cases our results are compatible with the previous studies. Also as shown in the above paragraphs, applying feature selection

improves classification accuracy in majority of the cases and for majority of the classifiers.

When we compare the previous studies with this thesis as summarized in Appendix 1, in this thesis, it is observed that PSO method is used in feature selection unlike other studies.

5. CONCLUSION AND FUTURE WORK

In this study, it has been investigated how the selection of features with PSO using four different datasets affect the classification accuracy. The datasets used in the study are BCCD, DRDD, SCADI and LSVT datasets taken from the UCI database. In the literature, no feature selection has been made by using PSO in these datasets. However, many different classification algorithms have been studied and compared by using these datasets.

In this thesis, feature selection is made and its effect on the classification accuracy is evaluated. First, classification accuracies for each dataset are examined before applying feature selection. Then, for each dataset, an optimal subset of attributes is selected by using the PSO algorithm. Each particle in the PSO algorithm selects a set of features and the fitness of the selected features are determined by using the NB classifier. According to the computed fitness values, set of selected features of the particles are updated and fitness of the new feature subsets are then computed. These steps are repeated until the termination criterion is satisfied. At the end of the PSO algorithm the best subset of features found are returned.

For each dataset, PSO is used to select a best feature subset. Then the datasets are reduced by keeping the selected features and eliminating the others. After that, the reduced datasets are classified with NB, SVM and RF classifiers in Weka.

According to experimental evaluations, it is observed that PSO-based feature selection improves classification accuracy of the BCCD dataset by 13.9%, DRDD dataset by 3.8%, SCADI dataset by 4.2%, and LSVT dataset by 2.5%. This is an undeniable success in data mining. In addition, we obtained the highest classification accuracy values for the BCCD and LSVT datasets with respect to the previous studies in the literature. BCCD has 80.5% classification accuracy with RF and NB classifiers, and LSVT has 87.2% accuracy with RF classifier.

When PSO-based feature selection method is compared with the well-known feature selectors that are IG, CHI2, and Relief, it is observed that IG and CHI2 are very similar performances and their classification accuracies are lower than that of PSO based method in majority of the cases. However, Relief has better performance with respect to IG and CHI2; and similar performance with PSO.

Although the PSO based feature selection method uses NB algorithm to determine best feature subset, the selected features also yield very good classification accuracies with the RF classifier, but they have lower classification successes with the SVM classifier. It may be better if SVM is used in PSO when SVM is applied to classify the reduced dataset. This experiment may be performed as future work of this thesis.

There are many factors affecting the success of classification in data mining. These are; programming language, artificial intelligence algorithm, classification algorithm, dataset and parameters used. Only the change of the programming environment can change the accuracy value. As an example, NB classifier implemented in Python and WEKA have different results. Also applying shuffling or not before doing k-fold cross validation affects the classification accuracy. Therefore, data mining is a process with very different dynamics. In future studies, one or more of the factors affecting the performance can be changed and their effects can be evaluated experimentally. For example, the used implementation environments, datasets, and classification algorithms' parameters can be changed; existing algorithms can be modified, and different optimization algorithms can be applied to increase the success of classification of medical datasets.

REFERENCES

Aalaei, S., Shahraki, H., Rowhanimanesh, A., and Eslami, S., 2016. Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets. Iranian journal of basic medical sciences, 19(5): 476–82.

Aghdam, M.H., and Heidari, S., 2015. Feature Selection Using Partcle Swarm Particle Swarm Optimization for Feature Selection. JAISCR, 5(4): 231–238.

Akar, Ö., and Güngör, O., 2013. Rastgele orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması. Journal of Geodesy and Geoinformation, 1(2): 139–146.

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

Akşehirli, Ö.Y., Ankaralı, H., Aydın, D., and Saraçlı, Ö., 2013. Tıbbi Tahminde Alternatif Bir Yaklaşım: Destek Vektör Makineleri. Turkiye Klinikleri Journal of Biostatistics, 5(1): 19–28.

Antal, B., and Hajdu, A., 2014. Diabetic Retinopathy Debrecen Data Set Data Set.

Arslan, N., 2018. Özelleştirilmiş Naive Bayes Algoritması.

Aslan, M.A., Celik, Y., Sabancı, K., and Durdu, A., 2018. One-shot versus stepwise gas-solid synthesis of iron trifluoride: investigation of pure molecular F2 fluorination of chloride p. International Journal of Intelligent Systems and Applications in Engineering, 6(4): 289–293.

Ayhan, S., and Erdoğmuş, Ş., 2014. Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi. Eşkişehir Osmangazi Üniversitesi İİBF Dergisi, 9(1): 175–198.

Bai, Q., 2010. Analysis of Particle Swarm Optimization Algorithm Abstract. Computer and Information Science, 3(1): 180–184.

Buarque, T., et al., 2017. Principal Component Analysis for Supervised Learning : a minimum classification error approach. Journal of Information and Data Management, 8(2): 131–145.

Budak, H., 2018. Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22: 21–31.

Chen, L.F., Su, C.T., Chen, K.H., and Wang, P.C., 2012. Particle swarm optimization for feature selection with application in obstructive sleep apnea diagnosis. Neural Computing and Applications, 21(8): 2087–2096.

Der, O., Vural, R.A., and Yıldırım, T., 1995. Inverter Design Based on Particle Swarm Optimization. YTÜ , İstanbul, 4 pp. (unpublished).

Doğan, B., 2009. Parçaçık Sürü Optimizasyonuna Dayalı Yeni Bir Aritmi Sınıflandırma Yöntemi. İTÜ Yüksek Lisans Tezi,İstanbul, 255pp. (unpublished).

Eberhart, R.C., and Shi, Y., 2002. Particle swarm optimization: developments, applications and resources. IEEE, 81–86.

Gandhi, K.K., and Prajapati, P.N.B., 2014. Study of Diabetes Prediction using Feature Selection and Classification. International Journal of Engineering Research & Technology (IJERT), 3(2): 2581–2584.

Gemici, B., 2012. Veri Madenciliği ve Bir Uygulaması. Dokuz Eylül Üniversitesi, Yüksek Lisans Tezi, İzmir, 103pp. (unpublished).

Güner, N., and Çomak, E., 2011. Mühendislik Öğrencilerinin Matematik I Derslerindeki Başarısının Destek Vektör Makineleri Kullanılarak Tahmin Edilmesi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 17(2): 87–96.

Guyon, I., and Elisseeff, A., 2003. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3: 1157–1182.

Hamed, E., et al., 2018. An analysis of particle swarm optimization for feature selection on medical data. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017, 227–231.

Jothi, N., Rashid, N.A., and Husain, W., 2015. Data Mining in Healthcare - A Review. Procedia Computer Science, 72: 306–313.

Jović, A., Brkić, K., and Bogunović, N., 2015. A review of feature selection methods with applications. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, 1200–1205.

Kaur, K., and Dhiman, S., 2016. Review of Data Mining with Weka Tool. International Journal of Computer Sciences and Engineering JCSE, 4(8): 41–44.

Keleş, A., and Keleş, A., 2008. ESTDD: Expert system for thyroid diseases diagnosis. Expert Systems with Applications, 34(1): 242–246.

Kennedy, J., and Eberhart, R.C., 1995. Particle Swarm Optimization. IEEE, 1942–1948.

Khan, D.M., Mohamudally, N., and Babajee, D.K.R., 2013. A unified theoretical framework for data mining. Procedia Computer Science, 17: 104–113.

Kılıkçıer, Ç., and Ersen, Y., 2012. İmge Eşiklemede Ayrık İkili PSO Temelli Yeni Bir Y aklaşım A new approach based on Discrete Binary PSO for image thresholding Elektronik Mühendisliği Bölümü. Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu ELECO, Bursa, 590–593.

Kurt, F., et al., 2017. Pathogenesis and treatment of diabetic retinopathy Diyabetik retinopati patogenezi ve tedavisi. Ahi Evran Tıp Dergisi, 1: 1–7.

Lavanya, D., and Rani, K.U., 2011. Analysis of Feature Selection with Classification : Breast Cancer Datasets. Indian Journal of Computer Science and Engineering (IJCSE), 2(5): 756–763.

Leung, K.M., 2007. Naive bayesian classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering.

Li, Y., and Chen, Z., 2018. Performance Evaluation of Machine Learning Methods for Breast Cancer Prediction. Applied and Computational Mathematics., 7(4): 212–216.

Maliha, M., Tareque, A., and Roy, S.S., 2018. Diabetic Retinopathy Detection Using Machine Learning. BRAC University Thesis, Bangladesh, 41 pp.

Mishra, A.K., and Ratha, B.K., 2016. Study of Random Tree and Random Forest Data Mining Algorithms for Microarray Data Analysis. International Journal on Advanced Electrical and Computer Engineering (IJAECE), 3(4): 5–7.

Mohammadian, S., Karsaz, A., and Roshan, Y.M., 2017. A comparative analysis of classification algorithms in diabetic retinopathy screening. In: 2017 7th International Conference on Computer and Knowledge Engineering, ICCKE. pp.84–89.

El Moudden, I., Ouzir, M., and ElBernoussi, S., 2017. Feature selection and extraction for class prediction in dysphonia measures analysis:A case study on Parkinson's disease speech rehabilitation. Technology and Health Care, 25(4): 693–708.

Nentwich, M.M., 2015. Diabetic retinopathy - ocular complications of diabetes mellitus. World Journal of Diabetes, 6(3): 489.

Ok, A., Akar, Ö., and Gungor, O., 2011. Rastgele Orman Sınıflandırma Yöntemi Yardımıyla Tarım Alanlarındaki ürün Çeşitliliğinin Sınıflandırılması. TUFUAB 2011 VI. Teknik Sempozyumu, Antalya, 8pp.

Ordukaya, M., 2010. Asenkron Parçacik SürOptimizasyonu. Ankara Üniversitesi, Ankara, 4s. (yayınlanmamış).

Orhan, U., and Adem, K., 2012. Naive Bayes Yönteminde Olasılık Çarpanlarının Etkileri. Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu ELECO, Bursa, 722–724.

Patil, T.R., and Sherekar, M.., 2013. Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. International Journal Of Computer Science And Applications, 6(2): 256–261.

Paul, J.P., 2005. Feature Selection Methods And Algorithms. British Journal of Plastic Surgery, 31(2): 1787–1797.

Pereira, J., et al., 2018. Using Resistin, glucose, age and BMI to predict the presence of breast cancer. BMC Cancer, 18(1): 1–8.

Ramaswamy, M., and Savitha, B., 2016. Comparative Study of Machine Learning Algorithms for Classification of Datasets using R Programming. International Journal of Advance Engineering and Research Development, 3(2): 157–161.

Sheena, Kumar, K., and Kumar, G., 2016. Analysis of Feature Selection Techniques: A Data Mining Approach. International Conference on Engineering & Technology, 4: 17–21.

Shi, Y., and Eberhart, R.C., 2002. A modified particle swarm optimizer. IEEE, 69–73.

Tamer, S., and Karakuzu, C., 2006. Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri. Kocaeli Üniversitesi, Müh. Fak., Elektronik ve Haberleşme Mühendisliği Bölümü, İzmit, 5pp.

Tekin, A., 2014. Early Prediction of Students' Grade Point Averages at Graduation: A Data Mining Approach. Eurasian Journal of Educational Research, 14(54): 207–226.

Tokan, F., Türker, N., and Yıldırım, T., 2005. Ekokardiyogram Verilerinin Yapay Sinir Ağları İle Değerlendirilmesi. Biyomedical Mühendisliği Ulusal Toplantısı, İstanbul, Turkey, 218–221.

Tran, B., Xue, B., and Zhang, M., 2014a. Overview of Particle Swarm Optimisation. Simulated Evolution and Learning, 605–617.

Tran, B., Xue, B., and Zhang, M., 2014b. Overview of Particle Swarm Optimisation for Feature Selection in Classification. Simulated Evolution and Learning, 268 pp.

Vaishali, R., et al., 2017. Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset. Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI, 1–5.

World Health Organisation, 2018. Cancer burden rises to 18.1 million new cases and 9.6 million cancer deaths in 2018. International Agency for Research on cancer, 13–15.

Yaman, I., 2014. Portföy Optimizasyonunda Değiştirilmiş Parçacik Sürü Optimizasyonu Yaklaşimi. Hacettepe Üniversitesi, Yüksek Lisans Tezi, Ankara, 60 pp.

Yan, P., and Jiao, M.H., 2016. An improved particle swarm optimization for global optimization. Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016, 8: 2181–2185.

Yang, C. S., Chuang, L.Y., Ke, C. H., and Yang, C.-H., 2008. Booean Binary Particle Swarm Optimization for Feature Selection. IEEE, 2093–2098.

Yao, Y.Y., 2003. A Step Towards the Foundations of Data Mining. Data Mining and Knowledge Discovery: Theory, Tools, and Technology V, 5098: 254–263.

Zarchi, M.S., Fatemi Bushehri, S.M.M., and Dehghanizadeh, M., 2018. SCADI: A standard dataset for self-care problems classification of children with physical and motor disability. International Journal of Medical Informatics, 114: 81–87.

Zunic, B., and Peter, S., 2018. Iris Recognition System Using Support Vector Machines. Biometric Systems, Design and Applications, 267–322.

http://bilgisayarkavramlari.sadievrenseker.com/

CURRICULUM VITAE

Ferda SUNA DÖKME was born in Diyarbakır in 1983. She studied at Şair Sırrı Hanım Primary School. Then she studied secondary and high school Diyarbakır Anatolian High School. She graduated from Mersin University Computer Engineering Department. She is currently working at the Information Technologies Research and Application Center at Mersin University.

APPENDIX

Appendix 1: Comparison of our study with the literature

Study	Datasets used	Classifiers applied	Feature Selection Method used
Pereira *et al.* study (2018)	BCCD	SVM,RF,LR	ROC analysis and ini Coefficient
Li and Chen study (2018)	BCCD	DT,SVMRF,LR,NN	X
Aslan *et al.* study(2018)	BCCD	ANN,ELM,K-NN,SVM	Hyperparameter Optimization Technique
Antal and Hajdu study (2014)	DRDD	RF,NB,SL,IBK,DT,IMC,J48,RT,Bagging	X
Ramaswawy and Savita study (2016)	DRDD	RF,SVM,NB,C5.0	X
Maliha, Tareque and Roy study (2018)	DRDD	RF,SVM,K-NN,NNET	X
Mohammadian, Karsaz and Roshan (2017)	DRDD	RF,SVM,NB,DT,K-NN,MLP,AB	X
Buarque *et al.* (2017)	DRDD	NB,DT,K-NN	Proposed Method (PM) and Principal Component Analysis (PCA)
Zarchi, Fatemi Bushehri, and Dehghanizadeh (2018)	SCADI	ANN,DT	X
El Moudden, Ouzir, and ElBernoussi (2017)	LSVT	LR,C5.0	Principal Component Analysis (PCA)
This Study (2019)	BCCD,DRDD, SCADI,LSVT	SVM,NB,RF	PSO+NB