T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

COMPARISON OF DATA REDUCTIONALGORITHMS FOR BIOMEDICAL APPLICATIONS

Thibaut Judicael BAH

MS THESIS

Computer Engineering Department

June-2015

KONYA

TEZ KABUL VE ONAYI

Thibaut Judicael BAH tarafindan hazırlanan “Biyomedikal Uygulama için Veri Azaltma Algoritmaları Karşılaştırılması” adlı tez çalıması 10/06/2015 tarihinde aşağıdaki jüri tarafından oy birliği ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak Kabul edilmiştir.

Jüri Üyeleri **İmza**

Başkan
Doç.Dr. Halife KODAZ ……………………

Danışman
Prof.Dr. Bekir KARLIK ……………………

Üye
Doç.Dr. Halis ALTUN ……………………

Yukarıdaki sonucu onaylarım.

Prof.Dr. Aşır Genç

FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Thibaut Judicael BAH

10/06/2015

ÖZET

YÜKSEK LİSANS TEZİ

BİYOMEDİKAL UYGULAMA İÇİN VERİ AZALTMA ALGORİTMALARI KARŞILAŞTIRILMASI

Thibaut Judicael BAH

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof.Dr. Bekir KARLIK

2015, 83 Sayfa

Jüri
Prof.Dr. Bekir KARLIK
Doç.Dr. Halife KODAZ
Doç.Dr.Halis ALTUN

Tıpta yumuşak hesaplama yöntemi birkaç yıldır büyüyen bir alandır. Biyoinformatik araştırmada ilerlemeye giderek, ve aynı zamanda karmaşık, büyük ve çok boyutlu verisetlerine bakan. Örneğin, yönbağımlı doğrusal olmayan difüzyon ile biyomedikal ve yapısal hücre biyolojisi 3 boyut görüntülerden ilgisiz verilerin ortadan kaldırılması hesaplamada pahalı. ECG Holter kaydedildi ve görevi öğrenmek için 100 binden fazla kalp atışları saklanan, hangi bilgiyi değerlendirecek ve daha sonra nihai bir çalışma veya test için tercih edilecegi hangi kalp atışları belirlenecegi zor bir iştir; bir hesaplama açısından pahalı ve büyük bir bellek alanı gerektirir [1].

Tıbbi görüntülerde hastadan hastaya birçok ortak özellik sunmak, ancak aralarındaki farklılıklar her zaman bazı anormalliklere neden olmayabilir. Bu tür görüntüler için biçimi çeşitli görüntü işleme başarı sınırlayan bir karmaşıklığa yol açar.

Veri azaltma hedefliyor işlenecek konuyu kolay hale getirmek için de orijinal veri kümesinden gereksiz verileri ortadan kaldırmaktır. Veri azaltılması için etkili bir yaklaşımdır. Dahası, etkin biyoinformatik uygulamalarında önemli bir işlemdir.

Anahtar Kelimeler: Biyoinformatik, Özellik seçimi, Veri azaltma,Veri indirgeme, Veri madenciliği, Yumuşak hesaplama.

ABSTRACT

MS THESIS

COMPARISON OF DATA REDUCTIONALGORITHMS FOR BIOMEDICAL APPLICATIONS

Thibaut Judicael BAH

SELCUK UNIVERSITY SCIENCE INSTITUTE
COMPUTER ENGINEERING DEPARTMENT

Advisor: Prof.Dr. Bekir KARLIK

2015, 83 Pages

Jury

Prof.Dr. Bekir KARLIK
Assoc.Prof. Halife KODAZ
Assoc.Prof. Halis ALTUN

The soft computing method in medicine is a growing field for several decades. Bioinformatics research advance increasingly, and facing at the same time complex, complicated, large and multidimensional datasets.For example; removing irrelevant data from 3 dimensions images in biomedicine and structural cellular biology by Anisotropic nonlinear diffusion is computationally expensive.

ECG Holter recorded and stored more than 100 thousand heartbeats for it learning task, which is a difficult work to evaluate the information and then determine which heartbeats are to be choose for an eventual study or test; from a computational perspective it is costly and require a large memory space [1].

Medical images present many common features from patient to patient but the differences between them may not always be due to some abnormality. This variety of format for such images leads to a complexity that restricts the success of image processing.

Data reduction aims is toremove the irrelevant data, reduce the dimensionality, the instances, the redundancy and the complexity of a dataset in order to make it easy to be processed. It is an efficacious approach for data reduction. Moreover, it is a crucial procedure in effective bioinformatics applications.

Keywords:Bioinformatics, Data mining, Data reduction, Feature selection, Instance reduction, Soft computing.

ACKNOWLEDGEMENT

I would like to begin by expressing my deep gratitude to the one who in the past two years has been for me a father, a mentor, an advisor; the Prof.DrBekirKarlik my supervisor who supported me intellectually during the writing of my thesis.

I thank my family, especially my father and my mother for their moral support that has been a great solace throughout these years of hard work.

I would also like to thank all those who make me the honour of their friendship, I could mention among otherZunonCyrille, Augusta Wicaksono, Alaa Hamid, Desire AnnickKouman, Martel Makwemba and BurakDere. I have been blessed by your presence in my life.

Thibaut Judicael BAH
KONYA-2015

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

wij	Weight of node *ij*
y_j	Output of node *j*
o_k	Output of node *k*
d_k	Desired output of node *k*
N_j	Net Input
F_j	Activation function
ε	Learning coefficient
$D(x, f)$	Euclidean Distance
Acc	Accuracy
T	Training Set
S	Subset
V_s	Support Vector

ANN	Artificial Neural Network
BAHSIC	
CLU	Clustering
CNN	Condensed Nearest Neighbor
CTG	Cardiotocography
DROP	Decremental Reduction Optimization Procedure
ENN	Edited Nearest Neighbor
GCM	Generalized-Modified Chang Algorithm
GCNN	Generalized Condensed Nearest Neighbor
GNU	General Public License
HSIC	
ICF	Iterative Case Filtering
K-NN	K-Nearest Neighbour
LVF	
MLP	Multi Layer Perceptron
NSB	Nearest Sub-class Classifier approach
OSC	Object Selection by Clustering
POC-NN	Pair Opposite Class-Nearest Neighbor
POP	Pattern by Ordered Projections
SV-kNNC	Support Vector k-Nearest Neighbor Clustering
SVM	Support Vector Machine
TS	Tabu Search
WEKA	Waikato Environment for Knowledge Analysis
WP	Weighting Prototypes
PSR	Prototype Selection by Relevance

RIF	Resample Instance Filter
SRF	Stratified Remove folds Instance Filter

1. INTRODUCTION

Data Reductionis an approach that is generally useful in bioinformatics, where in a dataset a subset of data are chose for a specific learning task.

The best subset is the one that while havingthe least number of data gives also a better accuracy. This is an essential step of pre-processing and the process by which we can avoid the curse of dimensionality [2]. Dimension reduction has been an important topic of research since 1970's and has proven his effectiveness in taking off redundant and irrelevant data, improving at the same the results of learning tasks, increasing learning accuracy and giving a better understanding of the results [3].

Data reduction is used when the data is tough to be process or when the data mining tool used at that moment is computationally expensive. In the literature the data reduction problems are generally figured out using heuristic search (filter method) or using directly data mining tools (wrapper methods)[4].

Over the last decade,data reduction technique became very important in the field of bioinformatics being one of the important step in preprocessing and an essential condition for model building[5].

This approach diminish the number of data, therefore reduce the cost of recognition and at the same time in some case improve the classification precision due to the few number of datathat make it easy to be processed[6].

Data selection has the ability to make clear and understandable complicated and imprecise data in order for the learning algorithms to learn quick and accurately. Data Reduction can draw interest of various fields of applications in medicine, economics, mathematics, computer science, chemistry and other fields.

The main problem in medical area is the correct and fast diagnosis, because it takes important part in treatment process. Diagnosis some diseases by human has always limitations and human expertise might be the most critical of them. In medicine it is somehow not easy for the doctor to make a correct diagnosis every time. This is due to the fact that the doctor diagnosis is not based on a standard model but on his understanding and interpreting of the patient exam's result, consequently he can make mistakes; hence the importance of machine learning.

In this thesis, different types of data reduction algorithms are presented and compare using different types of datasets and learning algorithms. The purpose of this work is to show the efficiency of data reduction techniques.The study mainly consists of 4 steps:

- Data preprocessing
- Data Reduction
- Implementation
- Comparison of test results

1.1.Organization of Thesis

In the chapter two, data mining concept and definition is presented. It provides a summary of some familiar machine learning algorithms that will be utilized in the application. Also the utility and importance of the software WEKA in this work is explained. Then a literature survey is done to show the importance of the topic.

In the chapter three and four, techniques of data reduction (feature selection and instance reduction) are presented. Moreover the correlation between pattern recognition and data reduction and the different steps of data reduction approach are explained.

In Chapter five, three applications are presented; data reduction methods are applied on the data, and Naive Bayes, K-NN, ANN, C4.5 Decision tree are used for training. Then the test results of the selected data are compared with the results of the original data.

In the Last chapter, the test results of the biomedical data applications arediscussed,and then a conclusion and future worksare given.

1.2.Literature Survey

As many pattern recognition, data mining and statistical techniques were at the beginning not conceived to deal with big quantities of data containing most of the time irrelevant data, it has become important in order to have good learning accuracy to combine them with data reduction[7-9]. Richard L. Bankert and David W. Aha in 1994 proposed a work focused on improving predictive accuracy for a specific task: cloud classification. Properties specific to this task require the use of feature subset selection approaches to ameliorate case-based recognition accuracy[10].In 1994, John et al. made a survey on attributes subset selection Problems. They defined three type of attribute importance in order to make clear theircomprehension of existent algorithms,and to define their purpose–find a relevant subset of attributes that gives good accuracy.[11].Douglas Zongker and Anil Jain (1996) made an evaluation on data reduction algorithm. They illustrated the importance of feature subset selection techniques, especially the branch-and-bound algorithm that most of time gives the best subset of features of a reasonably high dimension dataset[6].Spence and Sajda (1998) presented a pattern recognition program to help the specialists on diagnosis and by the same time they shown the duty of data reduction. They have shown the benefits and disadvantages of attribute selection methods for ameliorate the screening of masses in mammographic ROIs [12].Kudo and Sklansky (2000) have proposed a comparative study on attribute selection algorithms for learning algorithms. In the work, the worth of an attribute subset is defined by the K-Nearest Neighbour classifier and different types of data are utilized. [13].Georges Forman (2003) haveproposed an expansive comparative work of a new data reduction technique for high dimensional field of text classification, using SVM and two class problems. It shown a good performance[14].SaeyInza et al. (2007) havepresenteda work on attributes subset selection methods in bioinformatics.

Different data reduction approaches were compared and for each data reduction technique, they display a set of characteristics to allow the specialists to easily make the choice of a technique based on the intended objectives and the available resources[5].Song, Smola et al. have proposeda backward elimination approach for attributesubset selection with the HSIC. The intend of the creation of this algorithm, BAHSIC, was to find the attributes subset that maximises the correlation between the data and the classes[15].Ong et al. have developed a novel hybrid filter and wrapper data reduction algorithm based on a mimetic framework. The results of the experiments shown that this method is efficient to remove the irrelevant attribute and also able to generate good classification accuracy [16].

2. DATA MINING: CONCEPTS AND DEFINITIONS

2.1.Definition

Data mining also known as knowledge discovery is generally interpreted as the procedure which allows discovering important, valid, understandable, and potentially useful information from source of data (Fig.2.1), for example, texts, the Web, images, databases. Data mining is a domain that gathered many other domains such as visualization, statistics, machine learning, information retrieval, database, and artificial intelligence. Consequently Data mining is a process that makes finding solutions to problems by examining databases [7].

Fig.2.1. Knowledge discovering in databases[7]

The tasks of data mining are many, these are commonly −association rule mining, sequential pattern mining, unsupervised learning (or clustering) and supervised learning (or classification) [17]. During a data mining task or application, the data miners generally start with a good understanding of the application. After that, the data can be performed with data mining;there are generally three important steps:

- ✓ **Pre-processing** –generally the raw data is not appropriate for mining because of many reasons. Before using the data, it is recommended to clean it by removing abnormalities and noises. Sometimes the data is too large or contain many irrelevant data, hence the importance of data reduction by sampling and data reduction.

- ✓ **Data mining –** the obtained data is nowsubmitted to a mining algorithm to bring out knowledge or pattern.

- ✓ **Post-processing**– in practice not all the knowledge or patterns brought to light are meaningful. This step finds the meaningful ones for experiments. To make the decision, many visualization methods and assessment are utilized.

The process of data mining is iterative. Generally it takes several cycles or rounds to finally give a good result that are later used for experiments or operational tasks in the real world.

2.2.Representation ofData

Generally in supervised machine learning application the data correspond to a table of instances; each row representing an instance has an exact number of attributes, along with a class. Commonly attributes are of two types – numeric or nominal. In the Table 2.1 [18] fourteen instances representing different unsuitable and suitable days to play tennis. For each instance there are four features – Humidity, Outlook, Wind and Temperature, with a class label to precise whether or not the day is appropriate to play tennis.

Table 2.1. Tennis dataset

Instance	Attributes				Class
	Outlook	Temperature	Humidity	Wind	
1	sunny	hot	high	False	Don’t play
2	sunny	hot	high	True	Don’t play
3	overcast	hot	high	False	Play
4	rain	mild	high	False	Play
5	rain	cool	normal	False	Play
6	rain	cool	normal	True	Don’t play
7	overcast	cool	normal	True	Play
8	sunny	mild	high	False	Don’t play
9	sunny	cool	normal	False	Play
10	rain	mild	normal	False	Play
11	sunny	mild	normal	True	Play
12	overcast	mild	high	True	Play
13	overcast	cool	normal	False	Play
14	rain	mild	high	True	Don’t play

In a classic machine learning application there are two important data sets such as training sample and testing sample. The training sample is used to learn the concept to the algorithm and the testing set to evaluate the precision of the learning process. During the testing phase, the classes are not presented to the algorithm. The testing set is fed in the algorithm as input, and the algorithm gives as output the class label of each testing instance.

2.3.Learnıng Algorıthms

A learning algorithm is a model that can study and learn or get knowledge from data. Such algorithms proceed by constructing a model based on inputs, and then utilize these inputs to make decisions or predictions, instead of only following expressly programmed instructions.For example, Naive Bayes is a probabilistic summary form of knowledge; C4.5 [19]is a decision tree form of knowledge.

In this thesis, four machine learning algorithms are utilizedfor the comparison of the effects of attribute selectors on the data. These are ANN, naive Bayes, K-NN and C4.5 − each one of them hasa disparate learning method. These learning algorithms are commonly utilized by researchers, because they have shown their efficiency. ANN and C4.5 are the most developed algorithms of the four. The result of C4.5 algorithm is represented by a decision tree and is easy to interpret. Naive Bayes and K-NN are popular in the community because they are easily implementable and could perform as well as the sophisticated algorithms. [20-22]. The four following sections briefly present these algorithms.

2.3.1. Naive bayes

This algorithm is a kind of reduced form of Bayes approach used to evaluatewhether or not a new instance belongs to a class. The attribute values of the instance is utilized to calculate the posterior probability of classes; if the posterior probability of an instance according to a class is the highest then this instance appertain to this class. In Naïve Bayes, from a statistical point of view the attribute are independent according to each class (Equation 2.1).

$$p(C_i|v_1, v_2, \dots, v_n) = \frac{\mathrm{p}(C_i) \prod_{j=1}^{n} p(v_j|C_i)}{\mathrm{p}(v_1, v_2, \dots, v_n)} \quad (2.1)$$

In the equation 2.1, the left side represents the posterior probability of the class label *C*ᵢaccording to the attribute values$< v_1, v_2, \dots, v_n >$ present in the instance. The bottom part of the right side could be excluded because it is a constant and the same for all attributes regarding the class.

In the Table 2.2, the sub-tables a, b, c and d are eventuality tables representing the distribution frequency and the correlation between classes andattributes in the tennis data. These sub-tables are important for the calculation of the necessary probabilities to implement Equation 2.1. Now let make an example, image that someone at a certain moment in the journey want to know whether yes or not the weather is favourable to play tennis. Whereas the outlook is rain, the temperature is cool,

Table 2.2. Correlation between features and classes of the tennis data

	Play	Don’t Play	
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5
	9	5	14

(a) Outlook

	Play	Don’t Play	
hot	2	2	4
mild	4	2	6
cool	3	1	4
	9	5	14

(b) Temperature

	Play	Don’t Play	
High	3	4	7
Normal	6	1	7
	9	5	14

(c) Humidity

	Play	Don’t Play	
true	3	3	6
false	6	2	8
	9	5	14

(d) Wind

the humidity is high and the wind is false (there is no wind). We use the Equation 2.1 for the calculation of the posterior probability of each class, utilizing the information in the sub-tables a, b, c and d of the table 2.2:

$$\begin{aligned} p(Don'tPlay \mid rain, cool, high, true) = {} & p(Don't\ Play) \times p(rain \mid Don't\ Play) \times \\ & p(cool \mid Don't\ Play) \times p(high \mid Don't\ Play) \times \\ & p(true \mid Don't\ Play) \\ = {} & 5/14 \times 2/5 \times 1/5 \times 4/5 \times 3/5 \\ = {} & 0.0137 \end{aligned}$$

$$\begin{aligned} p(Play \mid rain, cool, high, true) = \ & p(Play) \times p(rain \mid Play) \times \ p(cool \mid Play) \times \\ & p(high \mid Play) \times p(true \mid Play) \\ = \ & 9/14 \times 3/9 \times 3/9 \times 3/9 \times 3/9 \\ = \ & 0.0238 \end{aligned}$$

Here the posterior probability of the class “Play” is high than the posterior probability of “Don’t Play”, therefore on this day we could play tennis.

Moreover because in naïve Bayes the attributes values are autonomous in the class, the performance prediction can be negatively influenced by the attendance of redundant attributes in the data especially in the training data. In 1994 Sage and Langley found that the performance of naïve Bayes ameliorates when redundant attributes are removed [22]. But, Pazzani and Domigos discovered that even if strong dependencies between the attributes negatively influenced the performance, when average correlations exist between the features naïve Bayes can still well perform [23].

2.3.2. C4.5 decision tree

The algorithms ID3 [24] and his successor C4.5 [19], represent in a kind of decision tree representing the training result. In practice decision tree algorithm is popular in the community, this is due to the fact that decision tree algorithm is fast in execution, robust and also because it produce a clear concept description, which is easily interpretable by the users. The Figure 2.2 presents a decision tree representing the tennis data’s training result.The features are represented as nodes in the tree, their associated or domains values as branches and the leaves on bottom represent the classes. Therefore to determine the class of a new instance, one first considers the values of the instance's attributes in the tree and then follows the corresponding values of the branches until reaching the leaf that indicate the class of the attribute.

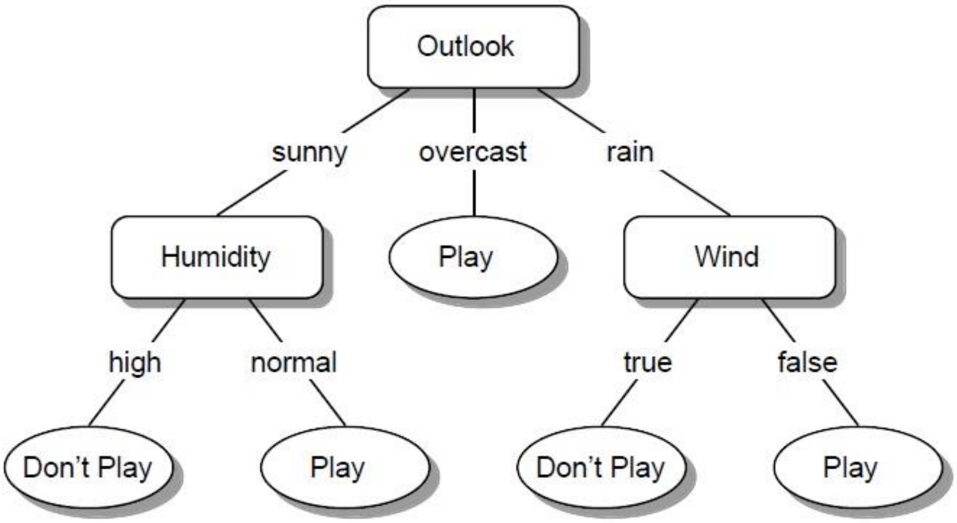

Fig. 2.2. Tennis data decision tree. Nodes correspond to attributes; the branches are the attributes' values and the leaves represent the classes.

In ID3 and C4.5 a greedy approach is used to form a decision tree. To select a feature as the root in ID3, one calculates first the entropy.

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i \qquad (2.2)$$

In the equation 2.2, the left side of the equation represents the *entropy* of the whole dataset *S*, and $p_i$is the portion of *S*belonging to class*i*; the logarithm is in base 2 because entropy is a measure of the expected encoding length measured in bits.Then we calculate the *Gain (S,A)* representing the *information gain* of each feature, defined as:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (2.3)$$

Where $A$is an attribute with a possible set of values, and S_v is a subset of $S$for which the attribute A has the value v.After this the feature which has the best information gain value becomes the root of the tree.To do the same thing (choose the root of the decision tree) C4.5 utilized the criterion of gain ratio[24] to determinate the feature that will be at the root of the decision tree. It chooses among the features with a good information gain,the one that optimizes the result of the division of its gain ratioby its entropy; the algorithm is iteratively repeated to create sub-trees.

In the community C4.5 is used as a benchmark algorithm against which the others learning algorithms performance is compared. C4.5 algorithm is fast, robust, accurate and above all it produces a structural comprehensible decision tree. Moreover, it deals very well with redundant and irrelevant data, That is why the influence of data reduction on its accuracy is little[25]. Even so, the decision tree's size can be reduce after removing redundant and irrelevant data[25, 26].

2.3.3. K-Nearest neighbours

K-NN[8] is an instance based learner but sometimes it is also call a lazy learner because it postpones the learning to the classification moment not before, and its power is in the instances matching plan. In K-NN algorithm, the learning is represented in the form of experiences or specific cases. It is based on effective approximation methods that recover the previous stored cases in order to know the class of a new pattern. K-NN as Naive Bayes generally consists of simple computations [27].

In K-NN to classify a new instance, the closest stored instance to the instance to be classified is determined using the Euclidean distance metric, then the class of this closest one is assigned the new instance.Euclidian distance formula is given as;

$$D(x, y) = \sqrt{\sum_{i=1}^{n} f(x_i, y_i)} \qquad (2.4)$$

This equation determines the Euclidean distance$D$ between$x$and$y$two instances; $x_i$and $y_i$refer to the i^{th} attribute value of pattern$x$ and y and for numeric pattern value $f(x_i, y_i) = (x_i - y_i)^2$.

K-NN can deal with irrelevant data, but to do so it need more training data, as a matter of fact, to maintain or reach a certain accuracy level, it has been demonstrated that the number of training datamust increase exponentially with the irrelevant data's number [2, 28, 29].

For this reason, after removing from the training cases noisy and redundant data it is possible to ameliorate the accuracy of nearest neighbour even if the remaining training data is restricted.

Moreover, because each instance to be classified should be compared successively to each stored training instance, the execution takes a lot of time. But the speed of the algorithm can be improved after reducing the training data's number

2.3.4. Artificial neural networks(ANN)

An ANN is a computer model that combines the human intelligent and the computers processing power; thereby it is able to process a large amount of data simultaneously from experience it has acquired[30]. ANN has several qualities that make them suitable for medical data processing. They are able to extract valuable knowledge from complexes data, something that would be complicated for humans to do [31]. They can also often overcome ambiguous and missing data [32] and provide accurate predications [33]

The most used neural network algorithm is the Multi-Layer Perceptron. A MLP is a set of neurons grouped into different layers these are – input layer, hidden layer(s), and output layer; they form parallel processing units.

The figure 2.3 presents a typical illustration of a MLP, each neuron in a layer is linked to each neuron of the next layer, and the connections are oriented from the input to the output layer. Then on each connection between two neurons of different layers there is a weight (numerical value) which represents the strength of the connection between these neurones − w_{ij}=connection weight between units*i*and *j* [34].

In MLP during the training, the connection weights change at each iteration. During the training, when a pattern is presented to the network, computations are done from the input to the output layer then the obtained result is compare to the desired output which is the class label of this pattern; this action is done until the desired iterations number or the stop criterion is reached. This kind of neural network is called a supervised because a desired output is needed in order for it to learn [35].

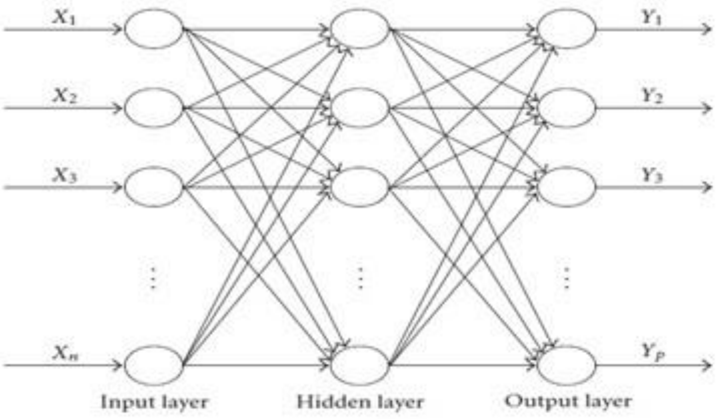

Fig.2.3. General architecture of MLP

The computation step of feedforward backpropagationmodel neural network proceeds like follow:

(1) The input layer neurones are activated when the input patterns are put in, this introduces the feedforward process

(2) The outputs of the first layer's neurones become the inputs of the next layer's neurones, we call it net input,

(a) The net input $N_j$is computes as follows:

$$N_j = \sum_{k=1}^{P} w_{jk}\, o_k \qquad (2.5)$$

Where o_k= output from previous units going on the next unit $j$as input, then P= number of inputson unit j.

(b) The value of their activation function is calculating with their net input:

$$a_j = F_j(N_j) \qquad (2.6)$$

The activation function$F_j$is generally a sigmoid function:

$$F_j = \frac{1}{1 + e^{-(N_j - \theta_j)}} \qquad (2.7)$$

(3) Again the units' outputs of this layer become the net inputs for the next layer. This process continues until it reaches the output layer, then the activation values of the output layer are called the actual output of the neural network computation.

Like explained by Rumelhart[36],the adjustment of the weights connections in the generalized delta rule is performin a given training case through the gradient descent on the total error:

$$\Delta w_{ij} = \eta \delta_j o_j \qquad (2.8)$$

In this formula, η refers to the learning rate which is a constant; δ_j= the gradient error of the net input at unit$j$. $\delta_j$is found by the subtraction of the computer activations a_j (also called actual outputs)from the expected activations t_j (also called desired outputs):

$$\delta_j = (t_j - a_j)F'(N_j) \tag{2.9}$$

where*F'* refers to the activation function's derivative. At the hidden layer, the desired outputs are not known. The next equationrefers to the gradient error givesδ_j *formula*for the hidden layers:

$$\delta_j = \left(\sum_{k=1}^{P} \delta_k w_{jk}\right) F'(N_j) \tag{2.10}$$

In the equation (2.10), a layer, the error rating to a hidden unit relies on the error of the units that affect it. Furthermore, the connection's weight between the hidden unit and the units that affect it influence the error's amount that coming from these units. The disadvantage of this algorithm is that it does not guarantee convergence toward a local minimum.

2.4.Performance Assessment

In a learning task, one of the most important steps is the performance assessment of the learning algorithms. Moreover, it is not just crucial for comparison of different algorithm, but it is an entire part learning algorithm.Although many others criterion of machine learning algorithms performances evaluation have been proposed; the testing set classification precision is the most used criterion[37, 38].

In this work, testing data classification precision is the main assessment criterion for all experiments; different data reduction methods and machine learning algorithms are utilized. A Data reduction algorithm is effective when the data amount is reduced and in addition the learning algorithm accuracy remains the same or improves. The classification precision is determined as the percentage of the training set elements properly classified by the algorithm. The error rate is therefore defined by − one minus the testing set accuracy. Utilizing the test set accuracy to measure the precision of the algorithm is better than utilizing the training instances because they have already been utilized to induce or create concept description. However sometimes the data is limited, in this case it is important to resample the data by partitioning it into two sets like usual – training and test sets. Then the machine is trained and tested with each set and the final accuracy is the average of both (training and testing) sets accuracies.

2.5.Weka Toolbox

In data mining, experiences have demonstrated that no single learning algorithm is suitable for all cases in data mining. In the real world, datasets vary, and for a machine learning algorithm fits with a dataset and gives and accurate model, the bias of this machine learning algorithm must accommodate the domain structure of the data. Therefore the universal learning algorithm is an utopia[7].

The workbench of Weka is a data processing tools and machine learning algorithms collection. It is shape so that we could easily experiment or test on a new dataset existing data mining methods in flexible ways.It affords almost all the tools for the whole experimental process of data mining, encompassing input data preparation, statistical evaluation of learning models and the visualization of the input samples and the learning result. It also provides a large variety of preprocessing tools. All those detailed and complete toolkit is available on one interface so that the users can easily compare different methods then choose among them the suitable one or the most accurate for the problem he want to solve.

Weka stands for *Waikato Environment for Knowledge Analysis.* It was developed in New Zealand at the University of Waikato. It has been written with the java programing language and published under the terms of the GNU General Public Licence. Furthermore, Weka can be used on practically any platform. It provides the same interface for most of the learning algorithms, together with techniques for pre-processing and post-processing and for the evaluation of learning algorithms on any given dataset.The Wekaworkspace consist of methods for almost all the standard data mining issues − clustering, regression, association rule mining, data reduction, and classification. The data is represented in a relational table, the formats which can be read are varied these are: ARFF, XLS, CSV, XLSX, etc.Weka provides implementations of learning algorithms that you can easily apply to your dataset. It also includes a variety of tools for transforming datasets, such as the algorithms for discretization. You can pre-process a dataset, feed it into a learning scheme, and analyse the resulting classifier and its performance–all without writing any program code at all.

One important way of utilizing Weka is to apply on a dataset a learning algorithm and analyse the output result to learn something about the data. Moreover it could be beneficial to use different learning algorithms to process a dataset them compare the results and choose the best one for prediction. In Weka the learning methods are called *classifiers* and tools for preprocessing are called *filters.*

3. FEATURE SUBSET SELECTION

To have successful machine learning task, it is important to take into consideration many factors and among them, the most significant is the quality of the dataset. In Theory, having many features should result in a best discriminability, yet, practically it has not always been the case; sometimes, good discrimination (classification) is achieved with limited dataset.

Because of this, the estimation of several probabilistic parameters is not easy. Therefore in order to prevent the training samples overfitting the bias of Occam's Razor [39]is utilized to construct a simple model that is able to achieve good performance with training sample. This bias sometimes encourage algorithm to favour data with small amount of features than the large ones, and if utilised properly can be fully accurate with the class label; but if the data contains noisy, irrelevant, unreliable or irrelevant data, it becomes difficult to learn throughout the training.

Feature Selection is a process that consists of identifying redundant and irrelevant data and then removes them; this process helps reduce the dimensionality and at the same time allow a fast and effective machine learning task. Moreover, in some cases, the future test performance can be better; in other words, the outcome is more compact and easily interpretable.

Many researches have shown that ordinary machine learning algorithms are negatively influenced by redundant and irrelevant data. Event the K-Nearest Neighbour learning algorithm is sensible to redundant and irrelevant features; its data complexity increases exponentially with the amount of irrelevant features [2, 22, 28]. For decision tree also in some cases such as parity concept, the data complexity can increase exponentially as well. In decision tree algorithm such as C4.5, the training samples can overfit often, having as a result a large tree[26].Therefore, by removing noisy data, in many cases the result can be better resulting in small tree easy to interpret. In Naïve Bayes algorithm, due to the fact that its features are independent in the class, it is also sensitive to irrelevant attributes [22].

In this chapter, we begin in section 3.1 and section 3.2 by reviewing common approaches to attribute subset selection (filters and wrappers) for machine learning present in literature. In sections 3.3 and 3.4, major aspects of feature subsetselection algorithms and some familiar searching methods (heuristic search)are presented.

3.1. Pattern Recognıtıon And Feature Selectıon

For the last decades, many researches in pattern recognition have been focus on feature selection techniques [40]. Just as for pattern recognition, feature selection is important for machine learning, because they share the same task of classification. In fact the feature subset selection have been developed to facilitate the knowledge extraction from big amount of data, and also to improve its comprehensibility [26]. For example in pattern recognition and machine learning, attribute selection techniques can help economise time in data acquisition, ameliorate precision of classification and ease the perplexity of the classifier [9]. Matter of fact, machine learning is based on both statistics and pattern recognition[4].

3.2.Feature Selectıon Algorıthms

In feature selection, the search is done through a feature space, therefore in order to perform well; it should follow four important steps that positively influence the search[41]:

1. Start step. There are two different way to start the search. The first way is to start with no feature in the space then successively add features. This way of search is called forward search process. Inversely, the search can start with all attributes in the search space and then successively remove the attributes until the best subset remains; this kind of search is commonly called backward feature search. Another way consists to start somewhere in the middle and remove useless attributes from this point.

2. Second step. It is about the search organisation; a complete search of the feature space is not recommended. Because for N initial number of features,

there are 2^N possible feature subsets. That is why a heuristic search is better and more conceivable than complete one by one search. Moreover heuristic search can produce good feature subsets, although it cannot every time give or guarantee the optimal subset.

3. Third step. It is about the strategy of evaluation; the only thing that differentiates the feature subset selection algorithm is the way the subsets are evaluated by each algorithm for machine learning. One model called the filter [6, 26] works independently of any machine learning algorithms—before the learning starts, irrelevant attributes are removed from the data. These algorithms are based on heuristics search to decide the quality of attribute subsets using the characteristics or properties of the data. However, some researchers think that the bias of a given learning algorithmshould be taken into consideration for the feature selection. this model is called the wrapper [6, 26], using learning algorithm along with cross validation to approximate the precision of the subsets of feature. An illustration of both models wrapper and filter is shown in Fig.3.1

4. The Fourth step is about stopping criterion; it is crucial for the feature selection algorithm to determine when to end the searching in the feature subsets space. According to the assessment strategy, a feature selection algorithm has to stop removing or adding attributes when none of the remaining attributes ameliorates the worth of the existent subset of feature. Otherwise, the feature selector could continue to correct the subset as the quality of the subset does not decrease.

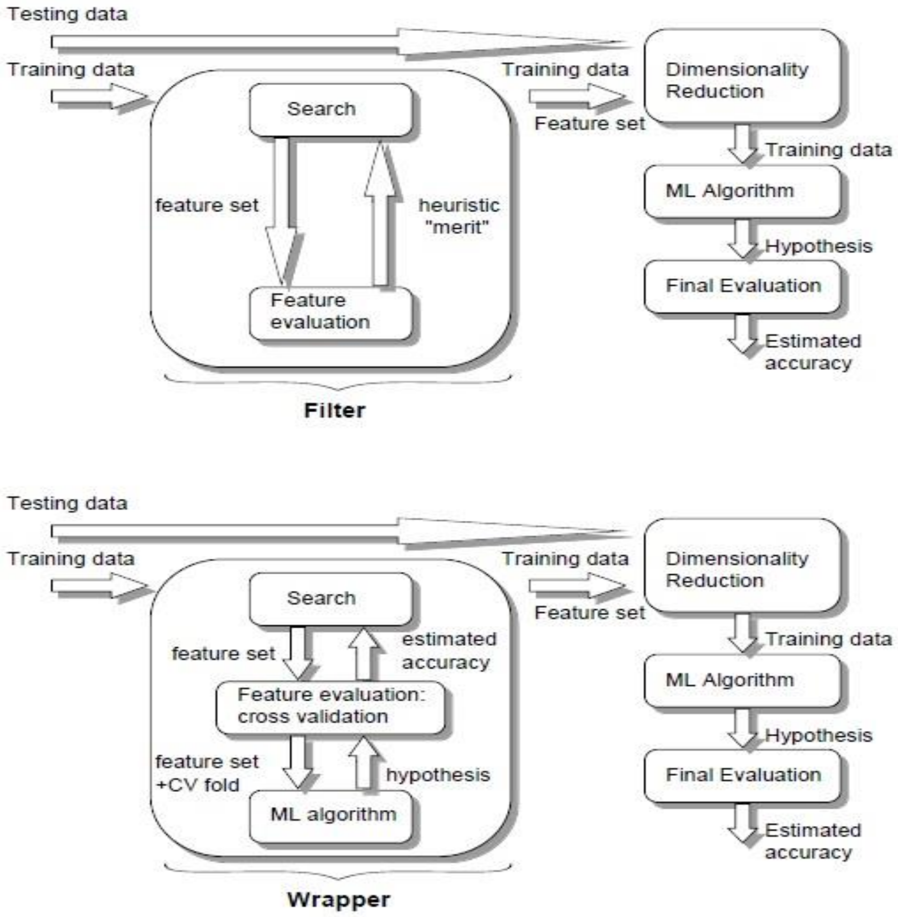

Fig.3.1. Wrapper and Filter algorithms [4]

3.3.Heurıstıc Search

When a feature selector is dealing with a large amount of features to extract the best feature subset from a feature subsets space and we want it to be done in an acceptable time, it is important to define constraints. For example, the greedy hill climbing, an ordinary search method provides local adjustment to the current subset of feature. Frequently, the local adjustment is merely the deletion or the addition of a single attribute to the subset.

In a feature selection algorithm, when only deletions from the feature subset is considered it is called a *backward elimination*; but when it only considers additions it is called *forward selection*[3]. Alternatively a method called *stepwise bi-directional* search, utilizes deletion and addition. within each techniques, in order to get the subset of features, in the algorithm pay attention to every variation in the feature subset then opt for the best, or in some case may merely select the feature subset that first ameliorate the worth of the current subset.In both case, when a change is admitted, it is never reconsidered. Fig.3.2 presents the feature subset space for tennis dataset. If swept from the bottom to the top the figure presents all possible local deletion; if scanned from the top to the bottom, it presents all the addition to each node [4].

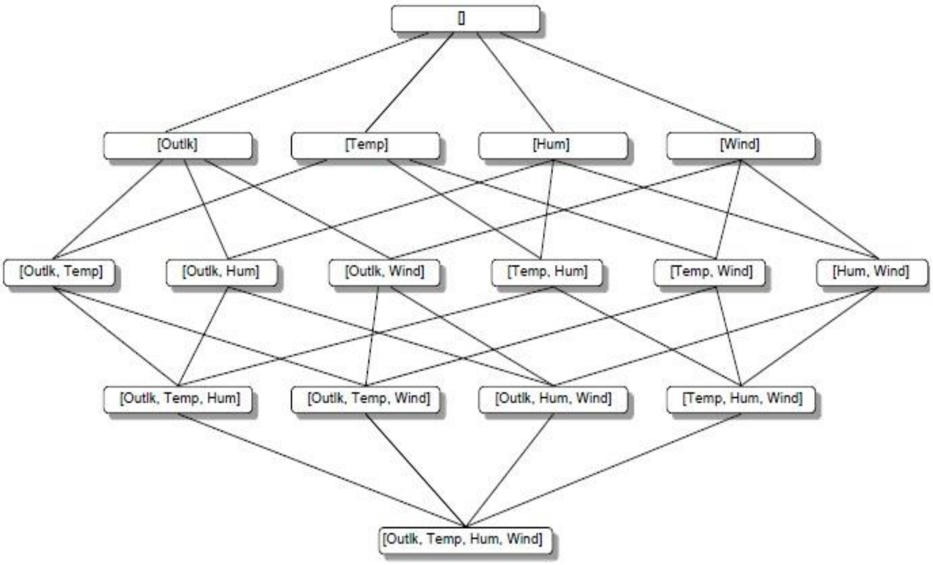

Fig.3.2. Space of feature subset for the "tennis" dataset [4]

The Table 3.1 presents *the greedy hill climbing search algorithm*.The Best first search [42] is an Artificial Intelligence method that permits backtracking on the search way. The best first goes through the feature subset search space just like greedy hill climbing algorithm by making local modification to the current subset of feature.

Table.3.1. Algorithm of Greedy hill climbing search [73]

1. Let $s \leftarrow$ start state.
2. Expands by making each possible local change.
3. Evaluate each child t of s.
4. Let $s' \leftarrow$ child t with highest evaluation $e(t)$.
5. If $(s') \geq e(s)$ then $s \leftarrow s'$, goto 2.
6. Return s.

Yet, differently to hill climbing algorithm, if the explored path performance begins to decrease, the best first algorithm may backtrack to another encouraging previous subset and from there continue the search. In the best first search, it is important to define a stopping criterion otherwise the search will go on until the exploration of entire space and this can take enough time. Table3.2 describes the algorithm of the best first search.

Table 3.2. The Best first search algorithm [72]

1. Begin with the OPEN list containing the start state, the CLOSED list empty. And Best ← start state.
2. Let s = arg max $e(x)$ (get the state from OPEN with the highest evaluation).
3. Remove s from OPEN and add to CLOSED.
4. If $e(s) \geq e$(BEST), then BEST ← s.
5. For each child t of s that is not in the OPEN or CLOSED list, evaluate and add to OPEN.
6. If BEST changed in the last set of expansion, goto 2.
7. Return BEST

Genetic algorithms are adaptive search methods founded on the criterion of natural biological selection [12]. They utilize many rival solutions—changed over time—to move toward a best solution.

In fact, to help keeping off local optima the space of solution is searched in parallel. For attribute selection generally, a result is a fixed (determined) binary length of string corresponding to a subset of feature—each value location in the string corresponds to the absence or presence of an individual feature. The algorithm is an iterative process in which each consecutive generation is created by using genetic operators such as mutation and crossover to the current generation members. Crossover put together different features from a couple of subsets in a new subset.While mutation transforms some of the values (thereby deleting or adding features) in a subset arbitrarily. The genetic operators' utilization on population members is defined by their fitness (quality of a subset of feature compared to an evaluation strategy); then through mutation and crossover, the better features subsets have more chance to be selected to become a new subset. A simple genetic search strategy is shows in table3.3. Crossover combines different features from a pair of subsets into a new subset. The application of genetic operators to population members is determined by their fitness (how good a feature subset is with respect to an evaluation strategy). Better feature subsets have a greater chance of being selected to form a new subset through crossover or mutation. In this manner, good subsets are “evolved” over time. Table 3.3 definesalgorithms of a simple genetic search strategy step by step.

Table 3.3. Simple genetic search strategy

```
1. Begin by randomly generating an initial population $P$.
2. Calculate $e(x)$ for each member $x \in P$.
3. Define a probability distribution $p$ over the members of $P$ where $p(x)\ \alpha\ e(x)$.
4. Select two population members x and y with respect to p.
5. Apply crossover to $x$ and $y$ to produce new population members $x'$ and $y'$.
6. Apply mutation to $x'$ and $y'$.
7. Insert $x'$ and $y'$ into $P'$ (the next generation).
8. If $|P'| < |P|$, goto 4.
9. Let $P \leftarrow P'$.
10. If there are more generations to process, goto 2.
11. Return $x \in P$ for which $e(x)$ is highest.
```

3.4.Wrapper Methods For Feature Selectıon

Wrapper methods for feature subset selection are methods that use learning algorithms to justify the quality of subsets of features. The principle of Wrapper approaches is that the induction or learning algorithm which is going to use the subset of feature must provide the best accuracy (Lan94). Often wrappers methods results are better than filter methods. This is due to the fact that in wrapper approaches, a specific learning algorithm and a training data are tuned together, then different subsets of the training data are tested until the best subset according to the induction algorithm is obtained. However compared to filters, wrappers are very slow because they frequently call the learning algorithm and when a different learning algorithm is used, they rerun. This section presents works centred on wrapper methods and techniques to decrease its computational cost.

3.4.1. Wrapper using decision trees algorithms

In 1994, Pfleger and Kohavi[43]have been the first researchers to propose the Wrapper as common technique for feature subset selection; they presented two precise characteristics of attribute relevance and affirmed that wrapper could find relevant attributes in a training set. According to them, given a full feature set, an attribute X_i is hardily relevant if the performance or the accuracy of the class values distribution decrease when it is removed. An attribute X_i is considered as slightly relevant when it is not hardily relevant and the accuracy of the class values distribution given a feature subset S of the full subset decrease when it is removed. However when an attribute is not hardily or slightly relevant it is irrelevant. To demonstrate this, experiments were done on different data utilizing ID3 and C4.5 as learning algorithm. The results demonstrate that feature selection did not importantly improve the accuracy of C4.5 or ID3. The main benefit was the reduction of the tree size.

During the 90's, many researchers tried to improve wrappers methods, among them Shavlik and Cherkauer[16]who in order to ameliorate wrapper on Decision tree algorithms utilized the genetic search approach. This approach successively ameliorates the accuracy of ID3 on a classification task. More precisely, to achieve this, they proposed an algorithm called SET-Gen whose purpose was to improve the accuracy as well as the easy comprehension of decision tree. This algorithm utilizes a fitness function:

$$Fitness(X) = \frac{3}{4}A + \frac{1}{4}\left(1 - \frac{S+F}{2}\right) \qquad (3.1)$$

where$X$ represents subset of feature, A is C4.5 cross validation accuracy average, S average of the tree size generates by C4.5 and F the features number in a subset.

3.4.2. Wrapper using naïve bayes classifier

Sage and Langley[22]due to the fact that in Naive Bayes classifier the distribution probability of a given feature is independent from the others, claim that the accuracy of Naive Bayes could be ameliorate if the irrelevant attributes are removed from the training set. In order to choose features that are going to be used with Naive Bayes, a strategy of forward search is utilized, unlike to decision tree learners that generally are used with backward strategy. The reason why forward search strategy is used is because it instantly discerns dependencies when irrelevant features are added. To test this assumption many experiment have been done and the selected features improved the performance during the classification task.

In the concern to ameliorate the precision of Naive Bayes Classifier, Pazzani[14]associates in a wrapper framework a simple constructive induction and feature selection. Then a comparison of backward and forward hill climbing search is done. The experiments results show that both methods ameliorate the Naive Bayes accuracy. Yet the forward search strategy is advantageous than backward search in removing irrelevant features because it begins with all the set of features and takes in consideration all the possible pair of attributes. The backward search strategy is efficient in determining interaction between features.

3.4.3. Wrapper improvement techniques

The computational expense of wrappers techniques is the basis of most of the blame on wrappers. With wrappers, each potential subset of feature is tested in a k-fold cross validation manytimes using a learning algorithm; therefore on a dataset with large amount of feature, the wrapper can be extremely slow. This handicap has pushed many researchers to do some research to find a way to reduce the computational cost of wrapper approaches. In 1994, a system that stores decision tree has been conceived by Caruanna and Freitag[15]; this to decrease the trees number generate over wrapper feature subset selection and free larger space for searches. In order to reduce the computation time of best first search and backward strategies, John and Kohavi[44]have presented the concept of *Compound Operators*. In a search of feature subset, the first Compound Operator is constructed after the full backward or forward search evaluation of a given set of features; the creation of the Compound Operator combines the two best subsets. Then this operator is utilized on the feature set to create a new feature subset, and if this subset ameliorates the performance, another operator is created and this time combining the best three subsets, and so on. The Compound Operator is very useful in the search of the best subset of feature. To verify the effectiveness of this technique, the compound operators were used with forward best first search to find a subset of feature, then this subset was trained and test with Naive Bayes and ID3. The results presented no important improvement of the precision for Naive Bayes and ID3. However, combining with backward search, the operators ameliorated the precision of C4.5 but insignificantly degraded the accuracy of ID3. The good result with C4.5 is due to pruning in C4.5 algorithm process which permits the best first search to surmount local minima, which is not the case for ID3.

A technique to compare concurrent feature subset selectors has been introduced in 1994 by Moore and Lee[10]. It is a forward selection algorithm. In this algorithm, a subset is removed from the competition of the best feature subset if over the leave-one-out cross validation this subset is considered to be improbable to have the smallest error rate; moreover the indistinguishable subsets are blocked; only one remains in the competition. This technique has the advantage to decrease the number of subsets that are going to be used over the training; therefore it reduces the computational time of the full evaluation. The competition cease as soon as only one subset of feature remains.

3.5.Filter Methods For Feature Selection

In machine learning, the first approaches concerning the feature selection algorithms were filter techniques. Those techniques are based on heuristic search rather than learning algorithms to assess the worth of feature subsets. As a matter of fact, filter techniques are faster than wrapper techniques; moreover they are simple, more practical and most of the time more effective on high dimensionality data.

3.5.1. Filters through consistency subset

In 1991 Almuallim and Dieterich[11] present the *FOCUS* algorithm which was originally made for Boolean domains; this algorithm completely searches the feature subset space until it reaches the minimum combination of attribute that shares the training set into pure classes where each combination of attribute values belong to a single class. This is called "min-features bias". After that, the ID3 [24] is used to build the decision tree of the final feature subset. As Freitag and Caruanna [15] have figured out, there are two great issues with FOCUS algorithm. The first one is that sometimes in FOCUS a complete search may be impossible if many attribute are needed to reach consistency. Secondly, an acute bias for consistency may be unwarranted statistically and can drive to an overfitting of the training set; because to solve only one inconsistency, the algorithm will keep adding features.In 1994 Almuallim and Dieterich[45]deal with the first of these issues. Three algorithms were designed to make FOCUS algorithm able to deal computationally with many features. These algorithms were forward selection search combined with heuristic to approximate the "min-feature bias". Thus using the following formula of information theoretic, the first algorithm assesses features:

$$Entropy(Q) = -\sum_{i=0}^{2^{|Q|-1}} \frac{p_i + n_i}{|Sample|}\left[\frac{p_i}{p_i + N_i} log_2 \frac{p_i}{p_i + N_i} + \frac{n_i}{p_i + n_i} log_2 \frac{n_i}{p_i + n_i}\right] \quad (3.2)$$

where, Q represents a given subset of feature, and there are 2^Q possible truth value assignments to the features. The training set instances in a given set of feature Q are divided with equal truth value assignments in Q. In each group, the Equation above calculates the general entropy of the class values; n^i and p^irespectively represent the negative and positive number of examples in the i^{th} group. At each step, the attribute which minimises the equation is put in the current subset of feature.

In the second algorithm, at each step of the search, the feature that presents the most discriminating characteristics is chosen and added to the current feature subset. A feature is discriminating if for two given examples negative and positive, the value of this feature is different for each one of them. At each step, the chosen feature is the one that discriminates the largest number of negative-positive couples of examples—which have not yet been discriminated by any existent feature in the subset.

The third algorithm looks like the second algorithm excepting the fact that a weight is incremented to the count of each feature that discriminate a negative-positive example pair. This increment relies on the number of feature that differentiates or discriminate the pair.

In 1996 Setiono and Lui[13] present the LVF algorithm comparable to FOCUS algorithm. It is consistency driven but differently to FOCUS, it can deal with irrelevant data if the irrelevant data level is approximately known. The LVF during the successive iteration randomly produces a subset S. İf the feature number of the S is fewer than the feature number of the current best subset, then the inconsistency rate of S and the inconsistency rate of the current best subset are compared and if S is at least as consistent as the best current subset, the best current subset is replaced by S.Setiono and Liu made some experiments with LVF; they used dataset with big amount of features and instances. They have shown that LVF was capable to reduce features number by more than half.

3.5.2. Feature selection by discretization

According to Liu and Setiono [17] it is possible to select feature using discretization methods. Combining Chi2 algorithm and discretization it is possible to create a feature selector. Initially the numerical features are sorted by positioning each attribute value in its interval. Then each attribute is discretized with χ^2 test to define when adjacent intervals should be merged. Then, to control the merging operation's extent, they used a χ^2 threshold which has been set automatically. This threshold is defined by trying to keep the structure of the original data. The process is ensured by inconsistency which is measured like in LVF algorithm.

Three reports have been done by the authors on data containing both numeric and nominal data utilizing C4.5 [24, 46] before and after discretization. They came to the conclusion that Chi2 is efficient at eliminating some features and improving C4.5 accuracy. However we really don't know whether it is the removing of some features or the discretization that is to the basis of the C4.5 performance improvement.

3.5.3. Feature filter using information theory

In 1996, Sahami and Koller[47]have developed a new feature selection algorithm based on probabilistic reasoning and information Theory[18]. Thereasons behind this feature subset selection method are that as the purpose of machine learning or pattern recognition algorithms is to evaluate the probability distributions for a class value. So the selected feature subset should remain as close as possible to the original distributions. For example consider a set of classes *C*, a set of features *V*, a subset *X* of *V*, *v* a set of values (v_1,...,v_n) assigned to each features, and v_x the projection of the values in f onto variables in *X*. The purpose of feature selection algorithm is to define *X* so that $Pr(C|X=v_n)$ remains as close as possible to $Pr(C|V=v)$. To do so the algorithm starts with the original features and at each step or stage, using the backward elimination search, it removes the feature that generates between the two distributions a change. To approximate the difference between two distributions, the cross entropy is utilized, also the number of features to be removed by the algorithm must be specified. Given two different features, the cross validation of the class distribution is given as:

$$D\big(\Pr\big(C\big|V_i = v_i, V_j = v_j\big), \Pr\big(C\big|V_j = v_j\big)\big) = \sum_{c \in C} p\big(c|V_i = v_i, V_j = v_j\big) log_2 \frac{p(c|V_i = v_i, V_j = v_j)}{p(c|V_j = v_j)} \quad (3.3)$$

From the remaining features, a set M_i composed of K attributes is found by the algorithm for each feature$i$, That is supposed to contain information that feature i has about class values. The features present in M_i have been taken from the remaining features for which the value of the Equation 3.3 is smallest. For each feature i, the cross entropy is calculated between the class distribution given only M_i and the class distribution given M_i, V_i. Then after the cross entropy performed for each feature i is done, the feature with the minimal quantity is deleted from the set. This process is executed until the number of features specified by the user is removed from the original dataset.

Experiments have been done on different dataset from different domains using Naive Bayes and C4.5 as learning algorithms. The experiments showed that the results of the feature selection algorithm are good when the size K of the conditioning Set M is set to 2. Also the algorithm is capable to reduce by more than half the features number in two domains having more than 1000 attributes, moreover it ameliorate the performance by about one or two per cent.

The problem of this algorithm is that to be encoded in binary the feature must have more than two values in order to avoid the bias that entropic measures have toward features with many values.

3.5.3. Feature filter using instance based approach

In 1992, Rendell and Kira [19] presented an algorithm so called RELIEF which gives to each feature a weight by utilizing instance based learning. The weight of each feature represents it ability to discriminate the classes. Furthermore, this weight is used to rank the features and the features whose weight is higher than the threshold specified by the user are selected to create a final subset of feature. The algorithm operates by randomly sampling the training data instances; and for each sampled instance, the nearest neighbour of opposite class called "nearest miss" and the nearest neighbour of same class called "nearest hit" is detected. The Updating of the weight of an attribute is made based on how its values identify the sampled instances from their nearest miss and nearest hit. If a feature distinguishes between instances of different classes without ambiguity, it weight will be high. The formula of the weight updating utilized by RELIEF is:

$$W_x = W_x - \frac{diff(X,R,H)^2}{m} + \frac{diff(X.R,M)^2}{m} \tag{3.4}$$

where, $W_x$represents the weight of feature X, M is the nearest miss, H the nearest Hit, R a randomly sampled instance and m represents the number of randomly sampled instances. For a given feature, the function diff computes the difference between two instances. For the continuous features, the difference is a real number normalized between 0 and 1, while the difference between nominal attribute is either 0 when the values are the same or 1 when the values are different. Then to ensure that all values are between [-1, 1] it is divided by m.

4. INSTANCE REDUCTION

In todays' databases, there are big amount of data; in order to extract useful knowledge from them with data mining applications, these databases need to be prepared. Thus, discarding some instances from the original dataset could help us to prevent unreasonable storage, excessive learning time and ameliorate classification accuracy.

In machine learning one of the important tasks is the automatic classification of instances, which is possible through a task of supervised learning and classification. To classify a new instance, an earlier evaluated set called the training set T is used as a classifier. Most of the time, this training set T contains irrelevant, superfluous, redundant and/or noisy data; in order to obtain good accuracy during the training task, these useless information must be discarded (see Fig.4.1).

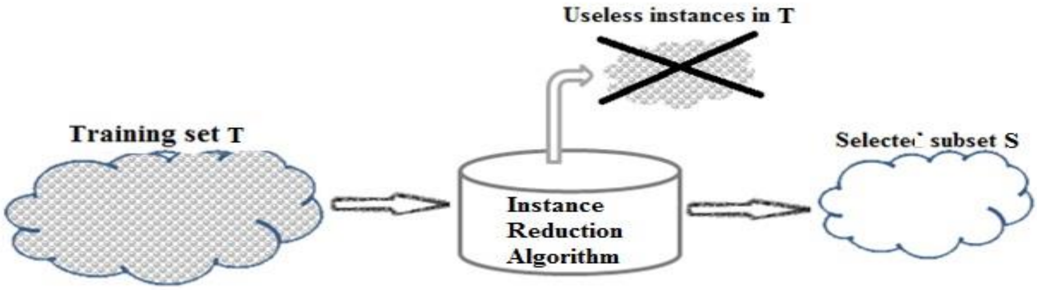

Fig.4.1: Instance reduction process

Considering *T* a training set, The aim of instance reduction techniques is to find a subset of instance $S \subset T$, with S without irrelevant instances and $Acc(S) \geq Acc(T)$, where *Acc(Y)* represent the classification accuracy of Y. A method of instance reduction can start with an empty instance subset space *(S=Ø)* these are called *forward or incremental* methods; again it can start with *S=T*, these methods are called *backward or decremental*. The difference between these methods is that in the forward methods the instances are iteratively added in the instance subset space, while in the backward methods the instances are iteratively removed from the subset space.

Through this process (deletion of irrelevant instances), the training set is reduced, and the training time could therefore decreased, especially in instance based classification task where to classify one instance the whole training set is utilized. As in feature subset selection there are two method groups, in instance reduction methods also we have two groups: wrapper and filter.

In *wrapper methods* for instance selection are methods that use learning algorithms to justify the quality of subsets of instance. While in *filter methods*, the selection is not based on learning algorithms but on heuristic search.

The objective of this section is to talk about instance selection techniques and their principal characteristics.

4.1. Wrapper Methods For Instances Reductıon

4.1.1. Wrapper methods based on the concept of nearest neighbors

In the literature, the wrapper methods are most of the time based on K-Nearest Neighbor learning algorithms[8].

The *CNN* (*Condensed Nearest Neighbor*) is one of the first wrapper methods for instance reduction[48]. It is a forward technique that initially put in the subset S one instance of each class to start.

The next step consist of classify each element of T in accordance with each first element of each class presents in S, Then when an instance p is misclassified according to his class it is put in S so that a the new instances comparable to p will be well classified. In this method, irrelevant instances could be kept due to the criterion because they are most of the time misclassified.

A recent version of *CNN* is *GCNN* (*Generalized Condensed Nearest Neighbor*) [49]. This method is similar to *CNN* but the only difference is that *GCNN* includes into the instance subset S the elements that fulfil a criterion of absorption in accordance with a threshold. The absorption is calculated for each instance according to the nearest enemies (instances that belong to a disparate class) and the nearest neighbors. Considering that an instance $p \in T$; we say that S absorbed p if:

$|p - x| - |p - w| > \delta$(4.1)

Where x and w belong to S (x, $w \in$ S) and are respectively nearest neighbor and nearest enemy of p; in such a situation p is not include in S.

Another earlier method of instance selection is the ENN (*Edited Nearest Neighbor*)[50]based on the deletion of irrelevant instances from the training set. In this method, when an instance p has a different class from his k nearest neighbors (with k=3) it is discarded from T.

Another version of the *Edited Nearest Neighbor* (ENN) method is the *All K-NN* method[51] which operates like this: for *i = 1* to *k*, flag all instances that are misclassified by their k-nearest neighbors. Then after the loop, all the instances that were flagged are deleted from T.

In 1991 Aha et al.[28] proposed instance based methods *IB2* and *IB3*. These methods are *forward* (incremental), the IB2 works like CNN; it picks out the misclassified instances by *1-NN*. IB3 is an improved version of IB2; it utilized the previous recorded classification in order to define the instances to be deleted, so that the discarding of some instances does not affect the classification performance.

4.1.2. Wrapper methods based on the concept of associate

In 2000 five instance reduction methods based on *the concept of associate* were presented by Wilson and Martinez [52]; These are *Decremental Reduction Optimization Procedure* (*DROP1 to DROP5*). Given an instance *p*∈T, the associates of *p* are his Nearest Neighbors. In the *DROP1* method, an instance *p* is removed from T if its associates are well classified without it; but if the associates of the irrelevant instance are first discarded, the irrelevant instance will not be deleted. In order to find a solution to this problem, *DROP2* which is pretty similar to *DROP1* search in T all the associates of the noisy instance *p*, and then if the associates instances of *p* in T are correctly classified without p, p is discarded. *DROP3* and *DROP4* meanwhile delete first the irrelevant instance utilizing a filter corresponding to *ENN* and afterward implement *DROP2*. *DROP5* is a *DROP2* based method which starts by removing the nearest instances with separate class; so that the decision boundaries will be clear.

There is also a recent method related to the concept on associate named *Iterative Case Filtering* algorithm (ICF) presented by Brighton and Mellish in 2002

[53]; this method is based on *Coverage(p)* and the *Reachable(p)* sets which are the associate and neighbor sets respectively. In ICF, p is removed from T when |*Coverage(p)*|< |*Reachable(p)*| meaning that similar instances to *p* can be correctly classified without p in T. Initially, ICF implement ENN. In the concept of *Coverage(p)* only the associates having the same class with p are considered so that only elements in the same class will be removed. But before deleting any instance, this method first defines whether the element is critical, noisy or superfluous. So an element is critical if its discarding change the classification of some others. In fact, this technique deletes either superfluous or noisy elements but leaves the critical instances. An instance p is considered as noisy when |*Coverage(p)*| < |*Reachable(p)*|; while it is superfluous if well classified by *Reachable(p).*

4.1.3. Wrapper method based on Support Vector Machine

The *Support Vector Machine algorithm* (SVM) [54] which is a learning algorithm can be used as a instance selector due to the fact that in this algorithm only the *support vectors* (Vs) are utilized to distinguish the classes; Here *Vs* is considered as S (S=Vs).

Yuangui et al. in 2005 [55] presented a SVM based wrapper method which made a double selection to reduce instances; The first selection extract the Vs after applying SVM, then the second step consists of implementing DROP2 on *Vs*.

Another instance reduction method based on *SVM* was introduced *Srisawat et al.*[56]; This method is *the Support Vector k-Nearest Neighbor Clustering* (SV-kNNC) which implements SVM over T to extract the Vs (*Support Vectors*); then after applies k-NN to cluster Vs, and it only retained the instances that belonging to the same cluster and having the same class (*homogeneous cluster*).However for the *non-homogeneous* cluster, the instances that not belong to the class of the majority are discarded, therefore only instances belonging to the majority having the same class are retained.

4.1.4. Wrapper methods based on Tabu and Sequential Search

Introduced in 1986 by Glover the Tabu Search (TS) [57] was implemented for instances selection[58, 59]. First of all an initial set called Solution (S_i) which is included in T ($S_i \subset T$) is used to implement TS. This is done to discriminate two kinds of solutions: the *Tabu Solution* and the *Non Tabu Solution*. The *Tabu Solution* is a solution that should not be changed; but the non Tabu Solution are assessed using a learning algorithm so that we could pick out the best one. To choose the best solution from the *Non Tabu Solution* ($\mathrm{S_i}$), the neighboring subsets are assessed. This is done iteratively and the iterations number is determined but a parameter called *Tabu Tenure*; when a subset S giving better accuracy than S_i is found, it replaces S_i.

4.2. Filter Methods For Instances Reduction

Unlike wrapper methods, the filter methods are not based on a classifier to determine the instances to be discarded from the training set.

Among the instances belonging to the same class in a dataset, two groups can be discriminated: interior and border instances. Given an instance $p_i \in T$ belonging to the class C_i; p_i is a border instance for C_i if p_i has as nearest neighbor an instance p_k belonging to the class $C_k$with$C_k \neq C_i$; therefore if p_i is not a border instances for C_i, it is an interior instance. However many filter techniques are focusing on border instances because they afford important information to preserve the class discrimination regions[52, 53].

4.2.1. Filter methods based on border instances

Another interesting attribute filter method: *Pattern by Ordered Projections* (POP) was proposed in 2003 by Riquelme et al.[60]it is based on *border instances*. This method is focused on the concept of *weakness(p)* which determines with regard to p features values how many times p is not *border* in a class. The filter rule deletes noisy instances that are, in accordance with this technique element that respect *weakness(p) = m*, where m is the attributes number of *p*. The *weakness* of a given instance *p* is calculated by augmenting the *weakness* of its features that are far from others instances with different class (that means this instance is not a border instance). To define the boundaries of an instance at most four elements are needed.

Another instance reducer that selects *border instance* is the *Pair Opposite Class-Nearest Neighbor* (POC-NN) [61]; in this method, for each class the mean of the instances is calculated. Considering an instance p_1 belonging to the class C_1; to define p_1 as a border instance, POC-NN calculates the instances' mean m_2 of the opposite class, then if the nearest instances to m_2 belonging to C_1 is p_1, therefore p_1 is a border instance for the class C_1. The process is the same for the class C_2.

4.2.2. Filter methods using clustering

Many researchers have initiated the hypothesis of instance reduction through *clustering*, among them: Bezdek and Kuncheva, Liu and Motoda, Spillmannetc[62-64]; the main idea is to define some instances as *centers* of *clusters* after *splitting T* in *m clusters*.

In 2002 Mollineda et al. presented the *Generalized-Modified Chang Algorithm* (GCM)[65] which put together the nearest clusters belonging to the same class in order to create *new clusters* and then for these new clusters finds new *centers*.

Also the *Nearest Sub-class Classifier approach* (NSB) [66] principal idea is to select many *centers* from the same class (in all the classes) using the *Maximum Variance Cluster* approach [67].

Lumini and Nanni in 2006 [68] introduced a technique of instance selection named *CLU* (Clustering) which was based on signature recognition. Moreover, the *Object Selection by Clustering approach* (OSC) [69] defines *S* by selecting some interiors instances; this method (OSC) splits the training set *T* in *n* clusters; afterward the interior instances are searched in the *homogeneous* (elements belonging to the same class) clusters, while borders instances in the *non-homogeneous* (elements that not belong to the same class) clusters. More clearly, in order to specify the border, The *OSC* method considers an instance *p* as a border when in a *non-homogeneous* cluster *p* is the nearest element to another element which belongs to an opposite class. In this technique the instance subset is formed with some elements of the *homogeneous* clusters and the centers of the clusters are always retained in order to keep the most representative elements.

4.2.3. Filter methods based on weights assigning

Some filter approaches in the literature consist of allocating to the instances a weight, and then those having a good weight in accordance with a *threshold* are selected. In 2000, Paredes and Vidal [69] utilized *gradient descent* in a method called *Weighting Prototypes* (WP) to calculate the weight to be assigned to each instances in terms of *nearest enemies* and *nearest neighbors*; then the elements with a weight larger than the *threshold* are discarded.

Again more recently in 2008, Olvera-López et al. [70] proposed another approach using the weights assigned to instances to define their relevance in *T*; This method is named *Prototype Selection by Relevance* (PSR). The main idea in this approach is that the relevance of instances is defined according to a parameter called *Average Similarity*; for example in a class, the most similar instances are the most relevant. In *PSR* the percentage of relevant instances to be selected in each class is specified by the user.

4.2.4. Filter methods based on sampling

Sampling is an approach that extracts a sample S from T by a random process where all the samples have the same chance to be chosen. Some constantly utilized sampling methods in the community are *Resample Instance Filter* (RIF) and *Stratified Remove folds Instance Filter* (SRF).

In the *Resample Instance Filter* the subset are produced randomly utilizing either sampling *without replacement* or sampling *with replacement*; the parameters used in this approach are: *sampleSizePercent, noReplacement, invertSelection, randomSeedand biasToUniformClass.*

Stratified Remove folds Instance Filter is an approach of sampling which before producing samples divides the dataset into subsets of instances called strata. Stratification is a procedure which consists of dividing elements of a group into homogeneous subset (strata) before sampling, each element is assigned to only one*strata*, and then in each *strata* a simple random sampling is implemented. This process minimizes the *sampling error* and ameliorates the representativeness of the sample.

5. APPLICATIONS

In this chapter, we present different medical dataset applications in order to show the importance and advantages of implementing data reduction technique before training and testing a data. We present 4 features selection and 2 instance reduction techniques, filters and wrappers; and use them in each proposed application; moreover we train and test the reduced dataset using the machine learning algorithms. Then we compare their results (data reduction techniques) with the original dataset resultsin the light of the test accuracies. The data reduction technique is a success if it happens to reduce the data by removing the noisy data and improving or not changing the accuracy of the original data; and is a failure if the accuracy of the reduced data is lower than the original data accuracy.

The titles of those 3 different applications are:

- Data selection techniques application on cardiotocography dataset using machine learning algorithms.
- Application of data reduction techniques on the Parkinson disease dataset using machine learning algorithms.
- A study of data reduction techniques using machine learning algorithms and Cardiac Arrhythmia Database.

For all experiments, 80% of the available data is used for training and 20% for testing. Furthermore, different data selection algorithms and four machine learning algorithms (Naïve Bayes, C4.5, K-NN and ANN) are used for testing and training.

5.1. Data Reduction Techniques Applicatıons On Cardiotocography Dataset Using Machine Learning Algorithms

Cardiotocography (CTG) is a test usually done in the third trimester of pregnancy. It is done to see if a baby's heart beats has a normal rate and variability. Normally, a baby's heart beats rate is anywhere between 110 and 160 beats per minute and increases when the baby moves. Checking that your baby's heart rate responds to his movements is an indirect way of knowing if he gets enough oxygen from the placenta. The test will also see how the baby's heart rate is affected by his mother's contractions. This study presents a data reduction based learning to diagnose whether a baby's heart beats are *normal, suspect or pathologic*.

5.1.1. Materials and methods

5.1.1.1. Dataset

The dataset used in this study has been taken from UCI database. 2126 foetal *cardiotocograms* (CTGs) were automatically processed and the respective diagnostic features measured. The CTGs were also classified by three expert obstetricians and a consensus classification label assigned to each of them. Classification was both with respect to a morphologic pattern and to a foetal state (Normal, Suspect, and Pathologic). Therefore the dataset can be used for 3-class experiments. Each patient is represented in the data set by 22 attributes.

5.1.1.2. Data reduction algorithms

The data reduction algorithms used for this study are supervised and have been taken from Weka (section 2.5.1). As*feature selection approaches* we chose respectively – *ConsistencySubsetEval*[71]which selects the feature subsets by the consistency level in the class values. *CorrelationAttributeEval*[4] selects an attribute subset by defining the correlation between the classes and the subset. *InfoGainAttributeEval*, this one shows the relevance of a feature by measuring the information gain with respect to the class, and *WrapperSubsetEval*[26] evaluates the worth of a subset of features by using the learning algorithms that it will utilize for training and testing; And for *the instance reduction methods* we chose *Resample*a filter method and *RemoveMisclassified* a wrapper method.

5.1.1.3. Learning algorithms

The purpose of this study is to show the efficiency of the data reductions techniques on medicine datasets; especially on a cardiotocography dataset; to describe whether the heart beats of a baby is normal, suspect or pathologic. For this reason, after the reduction of the data, we need a learning algorithm to train and test the data; so we use:

- Naïve Bayes (section 2.3.1)
- C4.5 decision tree (section 2.3.2)
- K-Nearest Neighbour (section 2.3.3)
- Artificial Neural Network (section 2.3.4). The Artificial Neural Network model utilized in all experiments is supervised. For the training stage, the data are fed into the network through the input layer along with the desired output; then after training, the network is tested with the testing data.

5.1.2. Application

In this study, we conducted several experiments with the original and the reduced dataset by using four different data reduction algorithms. We also used different percentage split for the training and testing data by using Naïve Bayes, C4.5, K-NN and ANN. For the training data, we used 80% of the available data, and 20% for testing. In the ANN, we utilized *1000 iterations* for each test except for the data reduced with *correlationAttributeEval* algorithm where we applied **500 iterations* because it gives the best results; backpropagation is used as learning algorithm, it updates weight and bias values according to optimization method.

5.1.2.1. Experiments and results

The samples in this work are collected from *2126 foetal cardiotocography*, and each instance is represented in the data by *22 features* including the class label. We have 2 important steps in this part: *the phase of data reduction* and *the phase of training and testing*. For the first step, we applied each *data reduction algorithm* to the *original dataset* in order to remove the *irrelevant, redundant, and noisy data* to reduce the data. In the second step, we used *four learning algorithms* (Naïve Bayes, C4.5, K-NN an ANN) for the training and testing of both *original and reduced samples*. Furthermore for the *ANN*, we used a *backpropagation* algorithm with one hidden layer. The neurons in the input hidden layer are equal to the number of features, idem for the hidden layer; however for the output layer we have three neurons representing class distribution of *cardiotocography dataset*. The *activation function* is *sigmoid* and the *learning rate* is set to *0.3*.

In the tables 5.1 to 5.4 four different feature selectors: *consistencySubsetEval, correlationAttributeEval, infoGainAttributeEval* and *wrapperSubsetEval*; and two instance reducers: *Resample* and *RemoveMisclassified* are used to reduce a cardiotocography dataset. Then four learning algorithms: *Naïve Bayes, C4.5 decision tree, Artificial neural network,* and *k- nearest neighbour* were utilized to train and test dataset before and after reduction. For all the experiments *1701 instances* representing *80 percent* of the sample for training and *425 instances* representing *20 percent* of the sample for testing.In all the tables the first experiment contains the test result of the original dataset; it is needed in order to be compared with the reduced data accuracies so that we could say whether yes or no *data reduction algorithms* influence the training and testing.In the column "*Features*" and "*Instances*" a sub column of "*experiment dataset*", we have different number of attribute and instance in some columns; these are their number after reduction. In fact, the selected data are the best according to the algorithm used.In the table 5.1 to 5.4 all the accuracies of the reduced data are better than the accuracy of the original data, except for the 5th experiment in the table 5.4 where we used the *wrapperSubsetEval* a feature selector.

Moreover, all the instance reducers performed better than the original data. In the first parts of all the tables wherewe applied

Table 5.1. Cardiotocography tests results of original and reduced data using Naïve Bayes

Exp	Learning Algorithm	Feature Selector	Experimental dataset			Test accuracy (%)	Correctly classified instances	Misclassified instances
			Feature	Instance	Class			
1	Naïve Bayes	None	22	2126	3	81.41	346	79
2	-	consistencySubsetEval	13	2126	3	82.35	350	75
3	-	correlationAttributeEval	13	2126	3	84.47	359	66
4	-	InfoGainAttributeEval	14	2126	3	83.06	353	72
5	-	wrapperSubsetEval	8	2126	3	**87.06**	370	55
-	-	**Instance Reducer**	-	-	-	-	-	-
6	-	Resample	22	1594	3	83.38	266	53
7	-	RemoveMisclassified	22	1748	3	**97.14**	340	10

Table 5.2. Cardiotocography tests results of original and reduced data using C4.5 decision tree

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correctly classified instances	Misclassified instances
			Feature	Instance	Class			
1	C4.5	None	22	2126	3	92.94	395	30
2	-	consistencySubsetEval	13	2126	3	93.18	396	29
3	-	correlationAttributeEval	13	2126	3	**94.59**	402	23
4	-	InfoGainAttributeEval	14	2126	3	94.35	401	24
5	-	wrapperSubsetEval	8	2126	3	93.18	396	29
-	-	**Instance Reducer**	-	-	-	-	-	-
6	-	Resample	22	1594	3	94.36	301	18
7	-	RemoveMisclassified	22	2059	3	**97.09**	400	12

Table 5.3. Cardiotocography tests results of original and reduced data using K-NN with K=3

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy(%)	Correctly classified instances	Misclassified instances
			Feature	Instance	Class			

1	K-NN	None	22	2126	3	90.82	386	39
2	-	consistencySubsetEval	13	2126	3	**92.94**	395	30
3	-	correlationAttributeEval	13	2126	3	92.47	393	32
4	-	InfoGainAttributeEval	14	2126	3	91.53	389	36
5	-	wrapperSubsetEval	8	2126	3	92.23	392	33
		Instance Reducer						
6	-	Resample	22	1594	3	92.48	295	24
7	-	RemoveMisclassified	22	2026	3	**95.55**	387	18

Table 5.4.Cardiotocography tests results of original and reduced data algorithms using ANN-MLP

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correctly classified instances	Misclassified instance
			Feature	Instance	Class			
1	ANN-MLP	None	22	2126	3	90.82	386	39
2		consistencySubsetEval	13	2126	3	90.82	386	39
3		correlationAttributeEval	13	2126	3	91.53	389	36
4	-	InfoGainAttributeEval	14	2126	3	**91.77**	390	35
5	-	WrapperSubsetEval	8	2126	3	90.12	383	42
-	-	**Instance Reducer**	-	-	-	-	-	-
6	-	Resample	22	1594	3	94.67	302	17
7	-	RemoveMisclassified	22	2007	3	**96.76**	388	13

feature selectors, in each table a distinct feature reduction algorithm gave the best result; to put it differently, none of the feature selector algorithms gave the best accuracy in all the cases, which means that all the data reduction algorithms used in this study are effective to decrease the noise and increase the accuracy.

Moreover for a better view of the results, a graphical representation is presented below (see Fig.5.1, 5.2, 5.3 and 5.4);they represent the accuracies of the reduced data. The *red*, *blue* and *orange* lines represent respectively the *original, feature selected and instance reduced data*.We can notice that the reduced data accuracies (blue and orange bars) are better than the accuracy of the *original* *data*(red line); again in each figure the best accuracy for the *feature selector*(blue line) is given by a different algorithm, we can deduce that the performance of the *feature selectors* depends on the structure of the data and the learning algorithm.

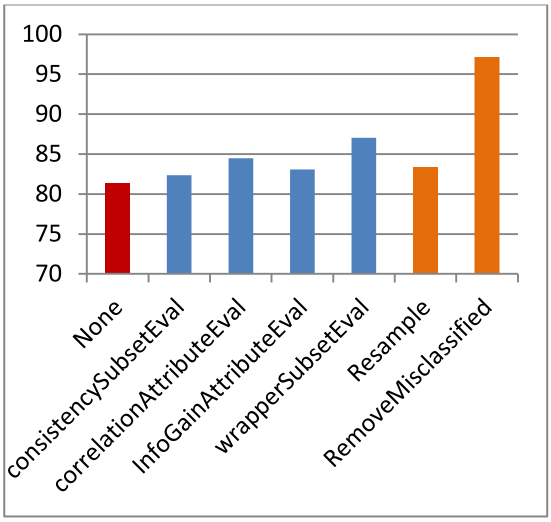

Fig.5.1. Cardiotocography test results of original and reduced data using Naïve Bayes as learning algorithm

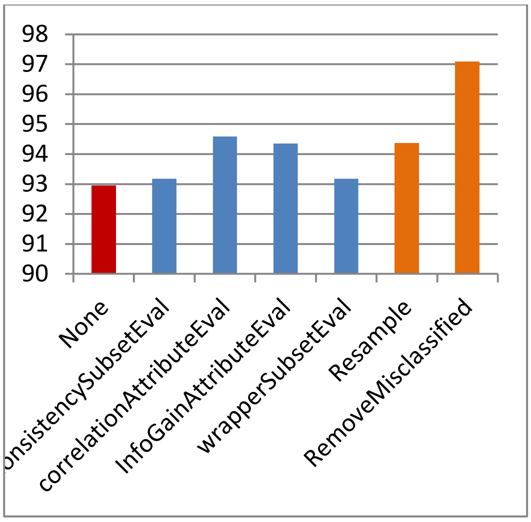

Fig.5.2. Cardiotocography test results of original and reduced data using C4.5 as learning algorithm

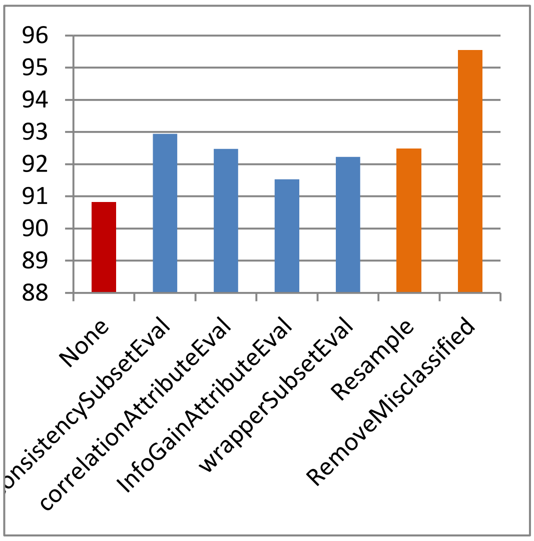

Fig.5.3. Cardiotocography test results of original and reduced data using K-NN as learning algorithm

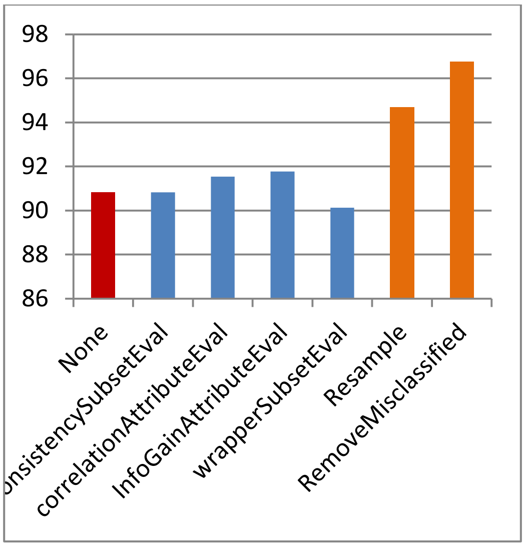

Fig.5.4. Cardiotocography test results of original and reduced data using ANN-MLP as learning algorithm

5.2. Applicatıon of Data Reduction Algorithms on Parkinson Disease Dataset Using Machine Learning Algorithms

Parkinson's disease is a progressive disorder of the nervous system that affects the movement. It develops gradually, sometimes starting with a barely noticeable tremor in just one hand. But while a tremor may be the most well-known sign of Parkinson's disease, the disorder also commonly causes stiffness or slowing of movement. In the early stages of Parkinson's disease, the face may show little or no expression or the arms may not swing when you walk. The speech may become soft or slurred. *Parkinson's disease* symptoms worsen as the condition progresses over time. Although Parkinson's disease can't be cured, medications may markedly improve the symptoms. In occasional cases, the doctor may suggest surgery to regulate certain regions of your brain and improve your symptoms. This study presents a data reduction based learning to diagnose whether a patient is affected or not by the *Parkinson's disease* using data reduction and machine learning algorithms.

5.2.1. Materials and methods

5.2.1.1. Dataset

The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. This dataset is composed of range of biomedical voice measurements from *31 people*, *23* with *Parkinson's disease* (PD). Each column in the table is a particular voice measure, and each row corresponds one of *195 voice recording* from one patient. The main aim of the data is to discriminate healthy people from those with PD, according to "class" column which is set to *0* for healthy and *1* for PD.

5.2.1.2. Data reduction algorithms

The data reduction algorithms used for this study are supervised and have been taken from Weka (section 2.5.1). As *feature selection approaches* we chose respectively – *ConsistencySubsetEval*[71]which selects the feature subsets by the consistency level in the class values. *CorrelationAttributeEval*[4] selects an attribute subset by defining the correlation between the classes and the subset. *InfoGainAttributeEval*, this one shows the relevance of a feature by measuring the *information gain* with respect to the class, and *WrapperSubsetEval* [26] evaluates the worth of a subset of features by using the *learning algorithms* that it will utilize for training and testing; And for *the instance reduction methods* we chose *Resample* a filter method and *RemoveMisclassified* a wrapper method.

5.2.1.3. Learning algorithms

The purpose of this study is to show the efficiency of the data reductions techniques on medicine datasets; especially on a Parkinson's disease dataset; to describe whether a patient is healthy or sick. For this reason, after the data reduction, learning algorithms are utilized to train and test the data (Original and reduced data); so we used:

- *Naïve Bayes* (section 2.3.1)
- *C4.5 decision tree* (section 2.3.2)
- *K-Nearest Neighbour* (section 2.3.3)
- *Artificial Neural Network* (section 2.3.4). The Artificial Neural Network model utilized in all experiments is supervised. For the training stage, the data are fed into the network through the input layer along with the desired output; then after training, the network is tested with the testing data.

5.2.2. Application

In this study, we conducted several experiments with the original and the reduced dataset by using four different data reduction algorithms. We also used different percentage split for the training and testing data by using *Naïve Bayes*, *C4.5*, *K-NN* and *ANN*. For the training data, we used *80%* of the available data, and *20%* for testing. In the ANN, for all the experiments we used 1000 iterations except for the original data where we applied 500 iterations because it gives better accuracy; and back-propagation is utilized as learning algorithm.

5.2.2.1. Experiments and results

The samples in this work are collected from *195 voice recording*, and each instance is represented in the data by *23 features* including the class. We have two important steps in this part: *the phase of data reduction* and *the phase of training and testing*. For the first step, we applied each *data reduction algorithm* to the *original dataset* in order to remove the *irrelevant, redundant, and noisy data* to reduce the data. In the second step, we used *four learning algorithms* (Naïve Bayes, C4.5, K-NN an ANN) for training and testing both *original and reduced samples*. Furthermore for the ANN, we used a backpropagation algorithm with one hidden layer. The neurons in the input hidden layer are equal to the number of features, idem for the hidden layer; however for the output layer we have *two neurons* which is class distribution of Parkinson's disease dataset. *The activation function* is *sigmoid* and *the learning rate* is set to *0.3*.

The tables 5.5 to 5.8 present the testing results of the original and reduced data using different data mining algorithms (Naïve Bayes, C4.5, ANN-MLP and K-NN). In all the tables, the first experiment represents the test result of the original data; from experiment 2 to 5, we used feature selectors to reduce the data and in the experiments 6 and 7 we utilized instance reducers. In the tables 5.5 and 5.6, the test results of the reduced data are all better than the test results of the original data, except for the second experiment in the table 5.6 where the accuracy is the same with the original data. In the table 5.7, only the third experiment gave a bad result acording to the original data; besides this all the reduced data performed better than the original data. Again in the table 5.8 where we used K-NN as learning algorithm, we recorded two bad result from the reduced data accuracies according to the original data, but the others reduced data accuracies are better than the original data.

Table 5.5. Parkinson Disease tests results for original and reduced data algorithmsusing Naïve Bayes

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correcty classified instances	Misclassified instances
			Feature	Instance	Class			
1	Naïve Bayes	None	23	195	2	66.67	26	13
2	-	consistencySubsetEval	10	195	2	69.23	27	12
3	-	correlationAttributeEval	10	195	2	74.36	29	10
4	-	InfoGainAttributeEval	10	195	2	74.36	29	10
5	-	wrapperSubsetEval	6	195	2	**89.74**	35	4
		Instance Reducer						
6	-	Resample	23	155	2	74.42	24	7
7	-	RemoveMisclassified	23	131	2	**100**	26	0

Table 5.6. Parkinson Disease test results for original and reduced data algorithms using C4.5

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correcty classified instancing	Misclassified instances
			Feature	Instance	Class			
1	C4.5	None	23	195	2	89.74	35	4
2		consistencySubsetEval	10	195	2	89.74	35	4
3		correlationAttributeEval	10	195	2	**94.87**	37	2
4		InfoGainAttributeEval	10	195	2	**94.87**	37	2
5		wrapperSubsetEval	6	195	2	92.31	36	3
		Instance Reducer						
6	-	Resample	23	155	2	90.32	28	3
7	-	RemoveMisclassified	23	173	2	**100**	35	0

Table 5.7. Parkinson Disease test results for original and reduced data algorithms using ANN-MLP

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correcty classified instancing	Misclassified instances
			Feature	Instance	Class			
1	ANN-MLP	None	23	195	2	89.74	35	4
2		consistencySubsetEval	10	195	2	**97.44**	38	1
3		correlationAttributeEval	10	195	2	87.18	34	5
4		InfoGainAttributeEval	10	195	2	94.87	37	2
5		wrapperSubsetEval	6	195	2	94.87	37	2
		Instance Reducer						
6	-	Resample	23	155	2	**90.32**	28	3
7	-	RemoveMisclassified	23	194	2	89.74	35	4

Table 5.8: Parkinson Disease test results for original and reduced data algorithms using K-NN

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correcty classified instance	Misclassified instances
			Feature	Instance	Class			
1	K-NN	None	23	195	2	94.87	37	2
2		consistencySubsetEval	10	195	2	**97.44**	38	1
3		correlationAttributeEval	10	195	2	92.31	36	3
4		InfoGainAttributeEval	10	195	2	92.31	36	3
5		wrapperSubsetEval	6	195	2	94.87	37	2
		Instance Reducer						
6	-	Resample	23	155	2	**100**	31	0
7	-	RemoveMisclassified	23	191	2	97.37	37	1

For a better view of the results, the diagrams (see Fig.5.5, Fig.5.6, Fig.5.7, Fig.5.8) below are presented. The *red line* represents the accuracy of the *original data*, the *blue lines* show the *feature selector results* and the *orange lines* the accuracies of the *instance reducers.*These figures conclude that selected data performed better than original data. Nevertheless, in the last figure the original data representing by the red line performed better than two of the selected data (blue line); it is due to the fact that the K-NN learning algorithm can sometimes handle the noisy and irrelevant data.

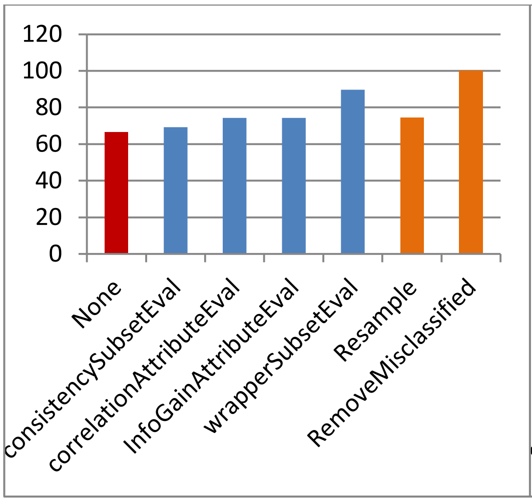

Fig.5.5. Parkinson Disease test results of original and reduced data using Naïve Bayes as learning algorithm

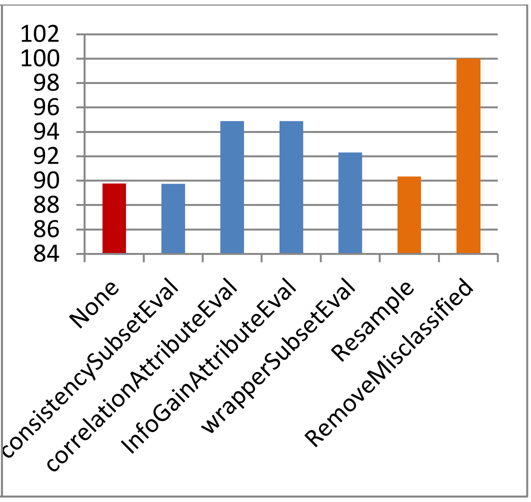

Fig.5.6. Parkinson Disease test results of original and reduced data using C4.5 as learning algorithm

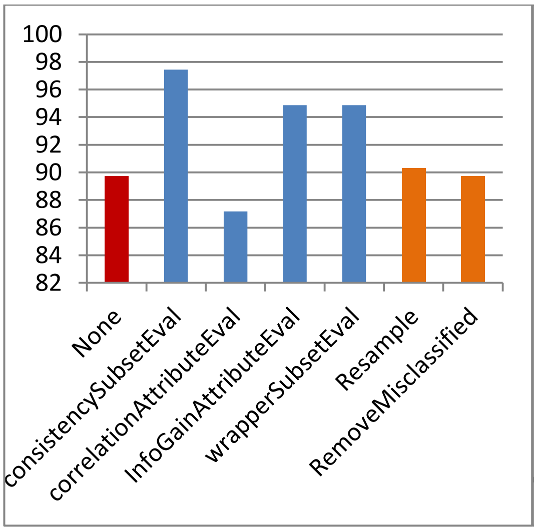

Fig.5.7. Parkinson Disease test results of original and reduced data using ANN-MLP as learning algorithm

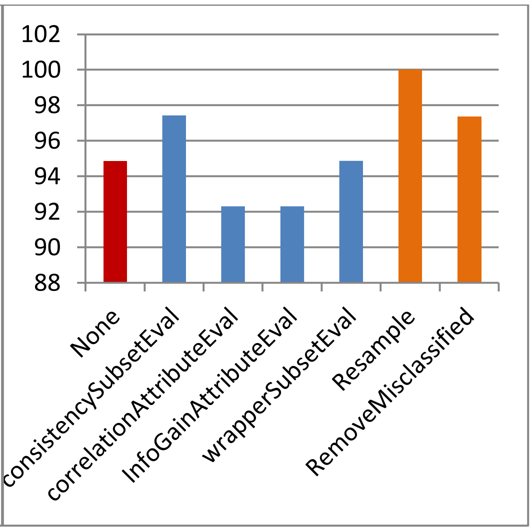

Fig.5.8. Parkinson Disease test results of original and reduced data using K-NN as learning algorithm

5.3. A Study of Data Reduction Techniques Using Machine Learning Algorithms And Cardiac Arrhythmia Dataset

The heart is a very important organ and more vital than most of the organs in the Human body, its dysfunction could be fatal for a patient. Sometimes heartbeat may be slow, too fast, too early and sometimes irregular. This type of heart malfunction is called *arrhythmia*.

Arrhythmia is a heart rhythm problem; it usually happens when the electrical impulses of the heart beat that coordinates it are not working as they should, making the heart beat too slow, too fast, incoherent or unstable. Indeed, anyone can experiment arrhythmia because many arrhythmias are innocuous because they are often due to stress, anxiety, shock... However some are potentially dangerous and even fatal, especially if they arise from sick hearts, or when the irregularity of the beats is very far from normal. When the heart beats is slower than normal, this arrhythmia is called *bradycardias*. When the beats are faster than normal, it is a *tachycardia* and when the heart beats are irregular, it is called *fibrillation*. Fibrillation can be *atrial* or *ventricular*. And premature contraction is when a single beat occurs earlier than expected. Arrhythmia is a strange disease because it is often not detected in some patients having symptoms, while often it is detected in patient presenting no symptoms.

5.3.1. Materials and methods

5.3.1.1. Dataset

The dataset was taken from UCI repository. Its goal is to define the absence or presence of cardiac arrhythmia and for the classification of a given patient data in one of the *13 classes*. The used dataset consists of *141 attributes* and *452 instances*. This dataset was pre-processed before training and testing because containing attributes with missing values and classes without instances. Initially this dataset had 279 attributes and 16 classes, and after removing the attributes with missing values and classes without any instances. 141 attributes and 452 instances and 13 classes remained.

5.3.1.2. Data reduction algorithms

The data reduction algorithms used for this study are supervised and have been taken from Weka (section 2.5.1). As *feature selection approaches* we chose respectively: *CfsSubsetEval*[4], this technique defines the relevance of attributes by estimating the prediction ability of each attribute and the redundancy rate between them. And the attributes that have low intercorrelation and high level of correlation with the class are selected;*FilteredAttributeEval*, it selects the attributes by using an arbitrary feature evaluator that has been approved by an arbitrary filter. Identically to the filter, the evaluator structure is defined by the training data; *FilteredSubsetEval*, which selects the subset by using an arbitrary feature evaluator that has been approved by an arbitrary filter;and *WrapperSubsetEval*[26] evaluates the worth of a subset of features by using the learning algorithms that it will utilize for training and testing. Then for *the instance reduction methods* we chose *Resample*a filter method and *RemoveMisclassified* a wrapper method.

5.3.1.3. Learning algorithms

The purpose of this study is to show the efficiency of the data reductions techniques on medicine datasets; especially on a Parkinson's disease dataset; to describe whether a patient is healthy or sick. For this reason, after the reduction of the data, we need a learning algorithm to train and test the data; so we use:

- Naïve Bayes (section 2.3.1)
- C4.5 decision tree (section 2.3.2)
- K-Nearest Neighbour (section 2.3.3)
- Artificial Neural Network (section 2.3.4). The Artificial Neural Network model utilized in all experiments is supervised. For the training stage, the

data are fed into the network through the input layer along with the desired output; then after training, the network is tested with the testing data.

5.3.2. Application

In this study, we conducted several experiments with the original and the reduced dataset by using four different data selection algorithms. We also used different percentage split for the training and testing data by using Naïve Bayes, C4.5, K-NN and ANN. For the training data, we used 80% of the available data, and 20% for testing. In the ANN, for all the experiments we used 500 iterations; and back-propagation is utilized as learning algorithm.

5.3.2.1. Experiments and results

This dataset is a collection of 452 instances, and each instance is represented by 141 features including the class. We have two important steps in this part: the phase of data reduction and the phase of training and testing.

In the tables 5.9 to 5.12 four *feature selectors* and two *instance reducers* are used to reduce a cardiotocography dataset; then four learning algorithms – *Naïve Bayes, C4.5 decision tree, Artificial neural network, and k- nearest neighbour* with *k=3* were utilized to train and test the original and reduced data. In all experiments *362 instances* representing *80%* of the available data are used for training and *90 instances* representing *20%* of the sample for testing.In all tables, the first experiment represents the test accuracy of the original data, the four next experiments show the test accuracies of the *feature selectors* and the two last experiments represent the *instance reducers* test accuracies.

In the tables 5.10 and 5.11, the test accuracies of the reduced data are all far better than the test accuracy of the original data. Also in the tables 5.9 and 5.12 the test accuracies of the reduced data are better than the test accuracy of the original data except for the third experiment of the table 5.9 and the second and fourth experiments of the tables 5.12 where the test accuracies are the same with the accuracy of the original data.

Table 5.9: Cardiac Arrhythmia test results for original and reduced data algorithms using C4.5

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correcty classified instance	Misclassified instances
			Feature	Instance	Class			
1	C4.5	None	141	452	13	55.55	50	40
2		CfsSubsetEval	27	452	13	**68.89**	62	28
3		FilteredAttributeEval	25	452	13	55.55	50	40
4		FilteredSubsetEval	27	452	13	**68.89**	62	28
5		WrapperSubsetEval	12	452	13	67.78	61	29
		Instance Reducer						
6	-	Resample	141	333	13	**74.63**	50	17
7	-	RemoveMisclassified	141	349	13	72.85	51	19

Table 5.10: Cardiac Arrhythmia test results for original and reduced data algorithms usingNaïve Bayes

Exp	Learning Algorithm	Feature Selector	Experiment dataset			Test accuracy (%)	Correcty classified instance	Misclassified instances
			Feature	Instance	Class			
1	Naïve Bayes	None	141	452	13	62.22	56	34
2		CfsSubsetEval	27	452	13	74.44	67	23
3		FilteredAttributeEval	25	452	13	68.89	62	28
4		FilteredSubsetEval	27	452	13	74.44	67	23
5		WrapperSubsetEval	18	452	13	**76.67**	69	21
		Instance Reducer						
6	-	Resample	141	333	13	70.15	47	20
7	-	RemoveMisclassified	141	297	13	**72.88**	43	16

Table 5.11: Cardiac Arrhythmia test results for original and reduced data algorithms using K-NN

Exp	**Learning Algorithm**	**Feature Selector**	**Experiment dataset**			**Test accuracy(%)**	**Correcty classified instance**	**Misclassified instances**
			Feature	Instance	Class			
1	K-NN	None	141	452	13	66.67	60	30
2		CfsSubsetEval	27	452	13	68.89	62	28
3		FilteredAttributeEval	25	452	13	**71.11**	64	26
4		FilteredSubsetEval	27	452	13	68.89	62	28
5		WrapperSubsetEval	12	452	13	**71.11**	64	26
		Instance Reducer						
6	-	Resample	141	333	13	82.09	55	12
7	-	RemoveMisclassified	141	288	13	**89.65**	52	6

Table 5.12: Cardiac Arrhythmia test results for original and reduced data algorithms using ANN-MLP

Exp	**Learning Algorithm**	**Feature Selector**	**Experiment dataset**			**Test accuracy (%)**	**Correcty classified instance**	**Misclassified instances**
			Feature	Instance	Class			
1	ANN-MLP	None	141	452	13	68.89	62	28
2		CfsSubsetEval	27	452	13	68.89	62	28
3		FilteredAttributeEval	25	452	13	75.56	68	22
4		FilteredSubsetEval	27	452	13	68.89	62	28
5		WrapperSubsetEval	18	452	13	**76.67**	69	21
		Instance Reducer						
6	-	Resample	141	333	13	**77.61**	52	15
7	-	RemoveMisclassified	141	403	13	71.60	58	23

For a better view of the results, graphical representations are presented; the diagrams (see Fig.5.9, 5.10, 5.11 and 5.12) below represent the test accuracies of the original (red line) and the reduced data (feature selection in blue lines and ınstance reduction in orange lines) accuracies. We can clearly deduce that the test accuracies of the reduced data are better than the test accuracies of the original data.

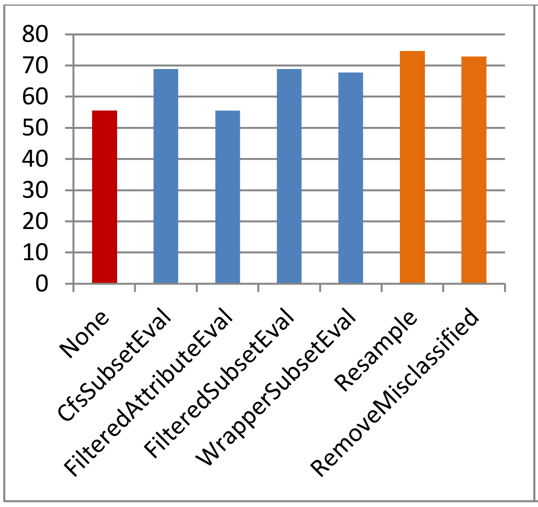

Fig.5.9. Cardiac Arrhythmia test results of original and reduced data using C4.5 as learning algorithm

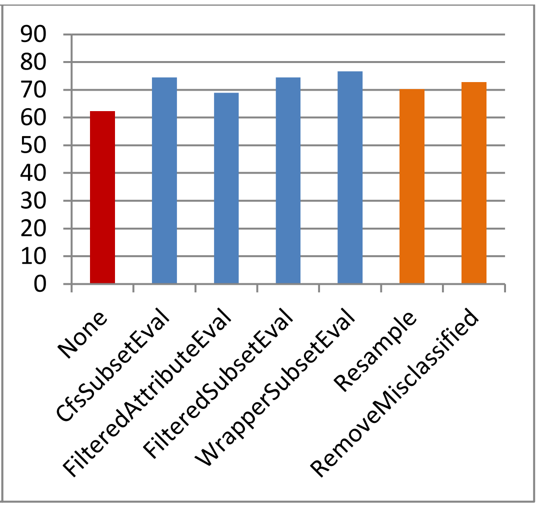

Fig.5.10. Cardiac Arrhythmia test results of original and reduced data using Naïve Bayes as learning algorithm

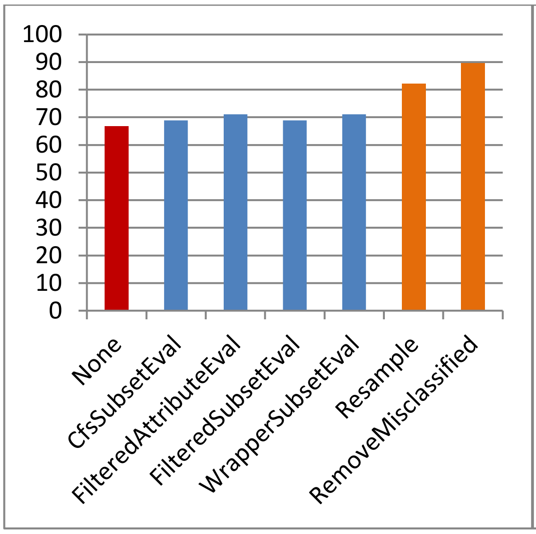

Fig.5.11. Cardiac Arrhythmia test results of original and reduced data using K-NN as learning algorithm

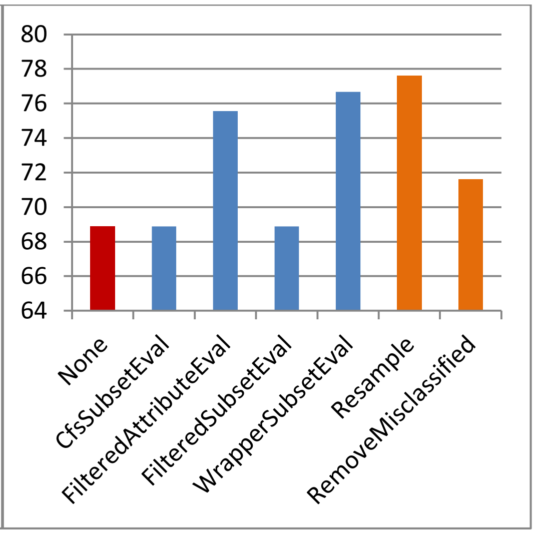

Fig.5.12. Cardiac Arrhythmia test results of original and reduced data using ANN-MLP as learning algorithm

6. CONCLUSIONS

6.1. Conclusıon

The work of this thesis aims to make a certain contributions to research on the data reduction in learning tasks. A work on the advantages of data reduction techniques for biomedical applications has been presented; several experiments were carried out with the utilization of different feature selection and instance reduction techniques, medical databases and machine learning algorithms. Then a comparison between the original data and reduced data results was made.

Supervised data reduction methodsand learning algorithmswere applied to the original data. All the data reduction techniques used are components of the WEKA workbench.

The obtained results demonstrated that the reduced data test accuracies are better than the original data accuracy. Therefore for better results of experiments in data mining field or the computer-aided diagnosis programs, it is crucial to use data reduction methods.

6.2. Future Work

In this thesis to demonstrate the effectiveness of data reduction methods on learning tasks, we used small databases in term of quantity of features and instances, and the results were satisfactory. However, nowadays the databases in most of the fields such as medicine, marketing, banks, web, trading… are more and more bigger. The next time we will try to make a research on the efficiency of data reduction methods in learning tasks using big databases.

BIBLIOGRAPHY

1. Delgado, E., et al. *Recognition of cardiac Arrhythmias by means of Beat Clustering on ECG-Holter Records*. in *Computers in Cardiology, 2007*. 2007. IEEE.

2. Langley, P. and S. Sage, *Scaling to domains with irrelevant features.* Computational learning theory and natural learning systems, 1997. **4**: p. 51-63.

3. Miller, A.J., *Selection of subsets of regression variables.* Journal of the Royal Statistical Society. Series A (General), 1984: p. 389-425.

4. Hall, M.A., *Correlation-based feature selection for machine learning*. 1999, The University of Waikato.

5. Blum, A.L. and P. Langley, *Selection of relevant features and examples in machine learning.* Artificial intelligence, 1997. **97**(1): p. 245-271.

6. Kohavi, R., *Wrappers for performance enhancement and oblivious decision graphs*. 1995, Citeseer.

7. Witten, I.H. and E. Frank, *Data Mining: Practical machine learning tools and techniques*. 2005: Morgan Kaufmann.

8. Cover, T. and P. Hart, *Nearest neighbor pattern classification.* Information Theory, IEEE Transactions on, 1967. **13**(1): p. 21-27.

9. Kittler, J. and P.A. Devijver, *Statistical properties of error estimators in performance assessment of recognition systems.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1982(2): p. 215-220.

10. Lee, M.S. and A. Moore. *Efficient algorithms for minimizing cross validation error*. in *Machine Learning Proceedings 1994: Proceedings of the Eighth International Conference*. 2014. Morgan Kaufmann.

11. Almuallim, H. and T.G. Dietterich. *Learning with Many Irrelevant Features*. in *AAAI*. 1991. Citeseer.

12. Holland, J.H., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. 1975: U Michigan Press.

13. Liu, H. and R. Setiono. *A probabilistic approach to feature selection-a filter solution*. in *ICML*. 1996. Citeseer.

14. Pazzani, M.J., *Searching for dependencies in Bayesian classifiers*, in *Learning from Data*. 1996, Springer. p. 239-248.

15. Caruana, R. and D. Freitag. *Greedy Attribute Selection*. in *ICML*. 1994. Citeseer.

16. Cherkauer, K.J. and J.W. Shavlik. *Growing Simpler Decision Trees to Facilitate Knowledge Discovery*. in *KDD*. 1996. Citeseer.

17. Liu, H. and R. Setiono. *Chi2: Feature selection and discretization of numeric attributes*. in *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*. 1995. IEEE Computer Society.

18. Pearl, J., *Probabilistic reasoning in intelligent systems: networks of plausible inference*. 2014: Morgan Kaufmann.

19. Kira, K. and L.A. Rendell. *A practical approach to feature selection*. in *Proceedings of the ninth international workshop on Machine learning*. 1992.

20. Clark, P. and T. Niblett, *The CN2 induction algorithm.* Machine Learning, 1989. **3**(4): p. 261-283.

21. Cost, S. and S. Salzberg, *A weighted nearest neighbor algorithm for learning with symbolic features.* Machine Learning, 1993. **10**(1): p. 57-78.

22. Langley, P. and S. Sage. *Induction of selective Bayesian classifiers*. in *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*. 1994. Morgan Kaufmann Publishers Inc.

23. Domingos, P. and M. Pazzani. *Beyond independence: Conditions for the optimality of the simple bayesian classifier*. in *Proc. 13th Intl. Conf. Machine Learning*. 1996.

24. Quinlan, J.R., *Induction of decision trees.* Machine learning, 1986. **1**(1): p. 81-106.

25. John, G.H., R. Kohavi, and K. Pfleger. *Irrelevant Features and the Subset Selection Problem*. in *ICML*. 1994.

26. Kohavi, R. and G.H. John, *Wrappers for feature subset selection.* Artificial intelligence, 1997. **97**(1): p. 273-324.

27. Cunningham, S.J., J. Littin, and I.H. Witten, *Applications of machine learning in information retrieval.* 1997.

28. Aha, D.W., D. Kibler, and M.K. Albert, *Instance-based learning algorithms.* Machine learning, 1991. **6**(1): p. 37-66.

29. Langley, P. and S. Sage, *Oblivious decision trees and abstract cases*. 1994: Defense Technical Information Center.

30. Ayer, T., et al., *Comparison of Logistic Regression and Artificial Neural Network Models in Breast Cancer Risk Estimation 1.* Radiographics, 2010. **30**(1): p. 13-22.

31. Dayhoff, J.E. and J.M. DeLeo, *Artificial neural networks.* Cancer, 2001. **91**(S8): p. 1615-1635.

32. Markey, M.K., et al., *Impact of missing data in evaluating artificial neural networks trained on complete data.* Computers in Biology and Medicine, 2006. **36**(5): p. 516-525.

33. Shamseldin, A.Y., *Application of a neural network technique to rainfall-runoff modelling.* Journal of Hydrology, 1997. **199**(3): p. 272-294.

34. Lee, Y.I., et al. *Design rules of multilayer perceptrons*. in *Aerospace Sensing*. 1992. International Society for Optics and Photonics.

35. Özbay, Y., R. Ceylan, and B. Karlik, *A fuzzy clustering neural network architecture for classification of ECG arrhythmias.* Computers in Biology and Medicine, 2006. **36**(4): p. 376-388.

36. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning internal representations by error propagation*. 1985, DTIC Document.

37. Kononenko, I. and I. Bratko, *Information-based evaluation criterion for classifier's performance.* Machine Learning, 1991. **6**(1): p. 67-80.

38. Cleary, J.G., S. Legg, and I.H. Witten, *An MDL estimate of the significance of rules.* 1996.

39. Gamberger, D. and N. Lavrač, *Conditions for Occam's razor applicability and noise elimination*. 1997: Springer.

40. Miller, A., *Subset selection in regression*. 2002: CRC Press.

41. Langley, P., *Selection of relevant features in machine learning*. 1994: Defense Technical Information Center.

42. Korf, R.E., *Linear-space best-first search.* Artificial Intelligence, 1993. **62**(1): p. 41-78.

43. John, G.H., R. Kohavi, and K. Pfleger. *Irrelevant features and the subset selection problem*. in *Machine Learning: Proceedings of the Eleventh International Conference*. 1994.

44. Kohavi, R. and D. Sommerfield. *Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology*. in *KDD*. 1995.

45. Almuallim, H. and T.G. Dietterich, *Learning boolean concepts in the presence of many irrelevant features.* Artificial Intelligence, 1994. **69**(1): p. 279-305.

46. Quinlan, J.R. *Bagging, boosting, and C4. 5*. in *AAAI/IAAI, Vol. 1*. 1996.
47. Koller, D. and M. Sahami, *Toward optimal feature selection.* 1996.

48. Hart, P., *The condensed nearest neighbor rule (Corresp.).* Information Theory, IEEE Transactions on, 1968. **14**(3): p. 515-516.

49. Chou, C.-H., B.-H. Kuo, and F. Chang. *The generalized condensed nearest neighbor rule as a data reduction method*. in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. 2006. IEEE.

50. Wilson, D.L., *Asymptotic properties of nearest neighbor rules using edited data.* Systems, Man and Cybernetics, IEEE Transactions on, 1972(3): p. 408-421.

51. Tomek, I., *An experiment with the edited nearest-neighbor rule.* IEEE Transactions on Systems, Man, and Cybernetics, 1976(6): p. 448-452.

52. Wilson, D.R. and T.R. Martinez, *Reduction techniques for instance-based learning algorithms.* Machine learning, 2000. **38**(3): p. 257-286.

53. Brighton, H. and C. Mellish, *Advances in instance selection for instance-based learning algorithms.* Data mining and knowledge discovery, 2002. **6**(2): p. 153-172.

54. Vapnik, V., *The nature of statistical learning theory*. 2000: Springer Science & Business Media.

55. Li, Y., et al., *Support vector based prototype selection method for nearest neighbor rules*, in *Advances in Natural Computation*. 2005, Springer. p. 528-535.

56. Srisawat, A., T. Phienthrakul, and B. Kijsirikul, *SV-kNNC: an algorithm for improving the efficiency of k-nearest neighbor*, in *PRICAI 2006: Trends in Artificial Intelligence*. 2006, Springer. p. 975-979.

57. Glover, F. and C. McMillan, *The general employee scheduling problem. An integration of MS and AI.* Computers & operations research, 1986. **13**(5): p. 563-573.

58. Cerveron, V. and F.J. Ferri, *Another move toward the minimum consistent subset: a tabu search approach to the condensed nearest neighbor rule.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2001. **31**(3): p. 408-413.

59. Zhang, H. and G. Sun, *Optimal reference subset selection for nearest neighbor classification by tabu search.* Pattern Recognition, 2002. **35**(7): p. 1481-1490.

60. Riquelme, J.C., J.S. Aguilar-Ruiz, and M. Toro, *Finding representative patterns with ordered projections.* Pattern Recognition, 2003. **36**(4): p. 1009-1018.

61. Raicharoen, T. and C. Lursinsap, *A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm.* Pattern recognition letters, 2005. **26**(10): p. 1554-1567.

62. Bezdek, J.C. and L.I. Kuncheva, *Nearest prototype classifier designs: An experimental study.* International Journal of Intelligent Systems, 2001. **16**(12): p. 1445-1473.

63. Liu, H. and H. Motoda, *On issues of instance selection.* Data Mining and Knowledge Discovery, 2002. **6**(2): p. 115-130.

64. Spillmann, B., et al., *Transforming strings to vector spaces using prototype selection*, in *Structural, Syntactic, and Statistical Pattern Recognition*. 2006, Springer. p. 287-296.

65. Mollineda, R.A., F.J. Ferri, and E. Vidal, *An efficient prototype merging strategy for the condensed 1-NN rule through class-conditional hierarchical clustering.* Pattern Recognition, 2002. **35**(12): p. 2771-2782.

66. Veenman, C.J. and M.J. Reinders, *The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005. **27**(9): p. 1417-1429.

67. Veenman, C.J., M.J.T. Reinders, and E. Backer, *A maximum variance cluster algorithm.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002. **24**(9): p. 1273-1280.

68. Lumini, A. and L. Nanni, *A clustering method for automatic biometric template selection.* Pattern Recognition, 2006. **39**(3): p. 495-497.

69. Olvera-López, J.A., J.A. Carrasco-Ochoa, and J.F. Martínez-Trinidad, *Object selection based on clustering and border objects*, in *Computer Recognition Systems 2*. 2007, Springer. p. 27-34.

70. Olvera-López, J.A., J.A. Carrasco-Ochoa, and J.F. Martínez-Trinidad, *Prototype selection via prototype relevance*, in *Progress in Pattern Recognition, Image Analysis and Applications*. 2008, Springer. p. 153-160.

71. Liu, H. and R. Setiono. *A probabilistic approach to feature selection-a filter solution*. 1996. Citeseer.

CURRICULUM VITAE
(ÖZGEÇMİŞ)

KİŞİSEL BİLGİLER

AdıSoyadı	:	Thibaut Judicael BAH
Uyruğu	:	FildişiSahili
DoğumYeriveTarihi	:	14/07/1987 Niakaramandougou'da
Telefon	:	5436250428
e-mail	:	cooljudi007@yahoo.fr

EĞİTİM

Derece	**Adı, İlçe, İl**	**BitirmeYılı**
Lise :		
Modern High School	San-Pedro	2006
Üniversite :		
Superior Institute of Technology Loko	Abidjan	2008
Superior Institute HEC La Roche	Abidjan	2011
Necmettin Erbakan University	Konya	2013
YüksekLisans :		
Selçuk University	Konya	June 2015

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
April 2012-july 2012	Superior Institute HEC La Roche	Trainee teacher.
20 September - 30 November 2010	ELITEL Technology	Traineeship
January 2012 - March 2012	Onyx Technology	Traineeship

UZMANLIK ALANI

Data Reduction Techniques
Data Mining
Machine Learning
Artificial Neural Network

YABANCI DİLLER

French, English, Turkish

YAYINLAR

1. Thibaut Judicael BAH and BekirKarlik, *The role of data reduction for diagnosis of pathologies of the vertebral column by using supervised learning algorithms.*XVIII International Conference on Soft Computing and Measurements, IEEE, 2015. Vol2(2): pp 61-65

2. Bah T.J., Karlik B., *Application of data selection techniques on cardiotocography data*. International Conference on Intellectual Systems for Decision Making and Problems of Computational Intelligence: Conference Proceedings – Kherson: KNTU, 2015. pp 337-338