BAYESIAN NETWORK CLASSIFIERS FOR DAMAGE DETECTION IN ENGINEERING MATERIALS

By

ADDIN OSMAN MOHAMED ADDIN

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirement for the Degree of Doctor of Philosophy

June 2006

TABLE OF CONTENTS

This page is intentionally left blank.

This page is intentionally left blank.

DEDICATION

The author would like to dedicate this Doctoral dissertation to his father Osman Mohamed Addin and to the soul of his mother Hawa Mohamed Suleiman. The author is deeply grateful for their patience in raising him up, dedication, and willingness for him to explore the world.

ABSTRACT

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Doctor of Philosophy

BAYESIAN NETWORKS CLASSIFIERS FOR DAMAGE DETECTION IN ENGINEERING MATERIALS

BY

ADDIN OSMAN MOHAMED ADDIN

June 2006

Chairman: Associate Prof. Ir. Dr. Mohd. Sapuan Salit, PhD, PEng

Faculty: Institute of Advanced Technology

Bayesian networks have been successfully implemented in many research and industrial areas. Nevertheless, they have not been thoroughly investigated and implemented such as Neural networks for damage detection in engineering materials. This thesis is dedicated to introduce the Bayesian network as a competitive probabilistic graphical model in general and as a classification tool (the Naïve bayes classifier) in particular for damage detection in engineering materials. The Bayesian networks in the thesis have been mostly introduced with the axioms of the damage detection. This is to let the thesis be considered as a reference too for the community of the damage detection and be self-contained. Bayesian networks have two-sided strengths: It is easy for humans to construct and to understand them, and when communicated to a computer, they can easily be compiled. The attractive aspect of the Bayesian networks is that there is a

rational way to build the network based on the causal relationship, informational, and mediating variables. Conversely, the Neural networks are forcing to guess the appropriate structure of the network. The states of the variables in the Bayesian networks can be calculated directly or indirectly through other variables. Thus, changes in a system model should only induce local changes in a Bayesian network, where as system changes might require the design and training of an entirely new Neural network.

The size of the network structures plays a vital role in their accuracies, when used as classifiers. The feature reduction (selection and extraction) represents a very important step in decreasing the sizes of the networks. The state-of-the-art shows that most of the reduction techniques, if not all, that have been implemented for damage detection in engineering materials were devoted for specific types of engineering materials and nondestructive techniques. They have been borrowed and implemented from other fields; no any one found to be proposed based on the waves used for damage detection. In addition, most of the implemented techniques are feature selection not extraction. Feature selection is less flexible than feature extraction in that feature selection is, in fact, a special case of feature extraction (with a coefficient of one for each selected feature and a coefficient of zero for any of the other features). This explains why an optimal feature set obtained by feature selection may or may not yield a good classification results. The feature selection is problematic, when there is a large number of potential features for classification and the best method to use depends on the circumstances. Therefore, it is important to propose a feature extraction algorithm for the damage detection.

The methodology used in the thesis provides a preliminary analysis used in proposing a new feature extraction algorithm (f-FFE: the f-folds feature extraction algorithm) as a general algorithm for all engineering materials and nondestructive testing techniques. The proposed algorithm divides the data into folders, forms new sets of data from these folders, clusters these sets using a clustering algorithm (e.g. k-means algorithm), then extracts the mean, maximum, and minimum values of the clusters to represent the extracted features.

The methodology is developed using two data sets. the first set represents voltage amplitudes of Lamb-waves produced and collected by sensors and actuators mounted on the surface of laminates contain different artificial damages. The laminates composed of *25 cm* × *5 cm* rectangular $[90/\pm45/0]_s$ quasi-isotropic laminates of the AS4/3501-6 graphite/epoxy system. Various types of damages were introduced to the specimens including, holes, fiber fracture, matrix cracking, and delamination. Lamb waves were propagated to the specimens by using *15* and *50 KHz* frequencies. The second set is a vibration data from a type of ball bearing operating under different five fault conditions. The ball bearing is of the type *6204* with a steel cage. The raw measurement data took the form of an acceleration signal recorded on the outer casing for the bearing in five states. The two sets are complete data without any missing values.

The derivation of the f-FFE algorithm is based on an empirical study carried out in the first data set. The empirical study has been shown as graphs of clusters, which formed from the data set after dividing it into folders. To verify the algorithm, the algorithm run on the second data set using the Naïve bayes classifier, which has been implemented on the Weka tool. Weka is an open source code data mining tool. Two software programs have been written in Java programming language so as to implement the parts of the proposed algorithm that were not covered by the Weka. The proposed algorithm implemented on the second set of the data.

The features extracted by the f-FFE algorithm were tested with different number of folders and clusters. The features contain four groups: the first group represents all features, the maximum, mean, and minimum, the second group represents the combination of mean and maximum, the third group represents the mean, and the fourth group represents the maximum values of the clusters. It has been assumed that the maximum values in the clusters represent the peaks of the amplitudes of the waves collected by a nondestructive testing method. The best results obtained when the number of clusters is four, the number of folders is six, and the combination of mean and maximum values has been used. The highest accuracy of the classifier obtained exceeds 95%. It has been shown that the maximum values only (the peaks) have shown the worst classification results in comparison to other cases and the mean values have show good results, which can be compared to the combination of the maximum and mean values. The number of the extracted features is highly decreased to *48,* while the original data contain *2048* amplitudes. Simultaneously, it increased the classification accuracy to more than 95%. The *k*-fold cross validation was used as a model evaluation method. This method divides the data set into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k* - 1 subsets are put together to form a training set. Then the average classification accuracy across all *k* trials is computed.

The studies conducted in this research have shown the Bayesian networks as one of the most successful machine learning classifiers for the damage detection in general and the Naïve bayes classifier in particular. The studies have shown also the effectiveness and efficiency of the proposed algorithm in reducing the number of the input features while increasing the accuracy of the classifier.

ABSTRAK

The translation will be done after the full correction and approval of the thesis by the supervisor and committee members.

ACKNOWLEDGEMENTS

Once, Isaac Newton (English mathematician and physicist, 1642 - 1727) said:

If I have seen further it is by standing on the shoulders of giants.

The writing of a dissertation is a tedious, lonely, and isolating experience, yet it is obviously not possible to successfully be completed, firstly, without the help of ”ALLAH” and secondly, without standing on the shoulders of many giants. First and foremost, the author would like to acknowledge and appreciate the generosity and unlimited help of his supervisor Associate Prof. Ir. Dr. Mohd. Sapuan Salit and his supervisory committee members, Associate Prof. Dr. Mohamed Othman and Dr. Elsadig Mahdi. They have always allowed him freedom to define and determine his own directions in research. They have also offered him valuable comments and suggestions, which played a vital role in successfully completing the thesis.

The author would like to thank Associate Prof. Ir. Dr. Mohd. Sapuan Salit for the financial support he offered him through the Ministry of Science, Technology, and Innovation, Malaysia. The financial support was offered for him as graduate research assistantships in the Faculty of Engineering, Universiti Putra Malaysia.

The author would like to thank professor Czeslaw Cempel and Mr Maciej Tabaszewski for offering the ball bearing data and Dr. Seth Stovack Kessler for offering the quasi-isotropic laminates data used in the thesis.

Many people contributed to this work by providing advice, support, and encouragement. The author would like to thank his father Osman Mohamed Addin for his continuous support and encouragement, his lovely daughter Waad for her great patience to be away from him during his Ph.D. studies, and all his family members. A deep sense of gratitude goes to Prof. Dr. Ibrahim M. Zein and his wife Dr. Nagat Mahmoud together with their daughters and son for their encouragement, wonderful advices, inspiration, and friendship over the dissertation years.

The author would like also to express his grateful thanks to his home mate Anwar Ali Yahya and Wan Mohd. Mozaaliff who contributed with discussions and suggestions in regards to the dissertation.

The author, never forgets the technical assistance provided for him by Ms. Rosnah Nawang, the Science Officer of the Advanced Material Laboratory at the Institute of Advanced Technology, University Putra Malaysia.

Lastly, an immense thank to all other friends who have contributed with valuable suggestions, and good ideas for the sake of the dissertation.

APPROVAL

I certify that an Examination Committee has met on _ _ /_ _ /_ _ _ _ to conduct the final examination of **Addin Osman Mohamed Addin** on his Ph.D. thesis entitled "BAYESIAN NETWORKS FOR DAMAGE DETECTION IN ENGINEERING MATERIALS" in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The Committee recommends that the candidate be awarded Doctor of Philosophy.

Members of the Examination Committee are as follows:

Chairman, PhD
Professor
Faculty of Graduate Studies
Universiti Putra Malaysia
(Chairman)

Examiner 1, PhD
Professor
Faculty of Graduate Studies
Universiti Putra Malaysia
(Internal Examiner)

Examiner 2, PhD
Professor
Faculty of Graduate Studies
Universiti Putra Malaysia
(Internal Examiner)

External Examiner, PhD
Professor
Faculty of Graduate Studies
Universiti Putra Malaysia
(External Examiner)

Professor Dr. Hasanah bt Mohd. Ghazali, PhD.
Professor/Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 13 June 2006

This thesis submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Doctor of Philosophy. The members of the Supervisory Committee are as follows:

Chairman of Supervisors Committee, PhD, PEng.
Associate Professor Ir. Dr. Mohd. Sapuan Salit
Department of Mechanical and Manufacturing Engineering
Faculty of Engineering
Universiti Putra Malaysia
(Chairman)

Member, PhD
Associate Professor Ir. Dr. Mohamed Othman
Department of Communication Technology and Network
Faculty of Computer Science
Universiti Putra Malaysia
(Member)

Member, PhD
Dr. Elsadig Mahdi
Department of Mechanical and Manufacturing Engineering
Faculty of Engineering
Universiti Putra Malaysia
(Member)

AINI IDERIS, PhD
Professor/Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 13 June 2006

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UPM or other institutions.

(signed)

ADDIN OSMAN MOHAMED ADDIN

Date: 13 June 2006

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

The list below shows the symbols used in this thesis. The symbols are generally explained when first introduced, and their meaning will normally be apparent from the context. In addition to the symbols in this list, arbitrary variables may appear locally in some part of the thesis to exemplify a description.

Latin Letters

Symbol	Description
a	New ball bearing
A	Variable with n states $(a_1, a_2, \ldots, a_n)$
b	Outer race completely broken
c	Broken cage with one loose element
c	Classifier
C	Number of class labels
d	Damaged cage, four loose elements
D	Entire data
D_n	A data set $(d_1, d_2, \ldots, d_n)$ with n features
e	No evident damage, badly work ball bearing
E	Set of directed ares between variables
G	Network structure
J	Number of Variables
K	Number of classes
m	Number of amplitudes in a folder
k	Number of clusters
V	Set of variables
V_1	Variable number one
V_2	Variable number two
V_m	A data set with m features
V_n	variable number n
$P(A)$	Probability distributions over variable A.
t	Total number of variables in each damage type
TAN	tree augmented Naïve bayes
w_n	An element representing class label
X	A data set with n variables $(x_1, x_2, \ldots, x_n)$
X	Original feature space

X'	Reduced feature space
x_c	Class label
x_i	Probability of A been in state a_i
χ^2-test	Statistical hypothesis test

Greek Letters

Symbol	Description
f	Mapping function
f	Number of folders
ω	Class label that takes a value of 1, 2, ..., C

Abbreviations

Symbol	Description
Age	The age of material
Amps	A set of n amplitudes (amp_1, amp_2, ..., amp_n)
Amplitude	The value of the wave's amplitude
ASH	Aluminum sandwich honeycomb
BN	Bayesian networks
BP	Back propagation
CFRP	Carbon-fiber reinforced polymers
Classify	Function Classify
CMs	Composite materials
DAG	Directed acyclic graph
CPTs	Conditional probability tables
DDF	Digital damage fingerprints
$D(X_i \mid pa(X_i))$	Data involving only X_i and $pa(X_i)$
EM	Expectation-Maximization algorithm
EMs	Engineering materials
f-FFF	f-Folds feature extraction algorithm
FFT	Fast Fourier transformation
$fold(1)$	Folder number one
$fold(2)$	Folder number two
$fold(f)$	Folder number f
LCMs	Laminated composite materials
LW	Lamb waves
MAP	Maximum Posteriori
Maxs	Maximum values of a cluster
MDL	Minimum Description Length
Means	Mean values of a cluster
Mins	Minimum values of a cluster

ML	Machine learning
NewAmp	A data set
NDE	Nondestructive evaluation
NDT	Nondestructive testing
N_{ijk}	The number of samples in D for which $X_i = k$ and $pa(X_i) = j$
NN	Neural network
$P(D\|S)$	Cooper-Herskovits scoring function
PMGs	Probabilistic graphical models
PZT	Piezoelectric Transducer
$Score_a$	Score for the *DAG* S_a after the change
$Score_b$	Score for the *DAG* S_b before the change
SMH	Structural health monitoring
S_{opt}	Bayesian network structure
S	Bayesian network structure
SUN	Selective unrestricted bayesian network classifier
ToolDrop	Tool dropped on the material
Damage	Presence of damage in material

CHAPTER

This page is intentionally included for some technicality related to latex. At the end of the day, this page will not be included when binding the thesis.

CHAPTER 1

BACKGROUND

1.1 Introduction

Recently, there has been a tremendous growth in the usage of engineering materials (*EMs*) in all types of engineering structures (e.g. aerospace, automotive, and sports). *EMs* are used to create a diversity of products, from computer chips and television screens to golf clubs and snow skis. *EMs* include metals, plastics, semiconductors, steel, aluminum sandwich honeycombs (*ASH*), and composite materials (*CMs*). *CMs*, *ASH*, and steel find wide usage in automobile and airplane parts on account of their stiffness and strength. Figure 1.1 shows a super *A*380 airbus plane, which the materials used in its construction are particularly innovative. About 25% of the *A*380 is made up of composites, 22% carbon fibre reinforced plastics (six times stronger and up to 60% lighter than steel) and 3% glare. Glare is a laminate of alternating layers of aluminum and glass fibre reinforced plastic, that is being used in civilian aircraft for the first time. Glare is not only lighter than aluminium, but is also more fire proof and has higher fatigue strength. It also reduces the weight of the *A*380 by 800 *kg* [1].

CMs are fabricated by combining two materials in which one of the materials, called the reinforcing material and the other called the matrix material. The reinforcing material is in form of fibers, sheets, or particles, which is embedded in the reinforced materials. The reinforcing and matrix materials can be metal, ceramic, or polymer.

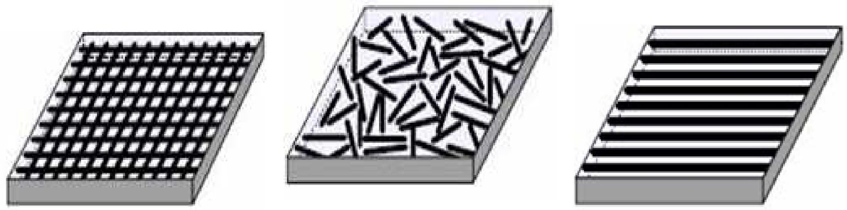
Figure 1.1: A super $A380$ airbus plane constructed from innovative materials [1]

CMs are designed to combine the strength of the reinforcement with the toughness of the matrix to achieve a combination of desirable properties that can not be guaranteed by the constituents when used separately. Examples of some application of *CMs* are the diesel piston, brake-shoes and pads, tires and the Beech-craft aircraft in which 100% of the structural components are composites [2].

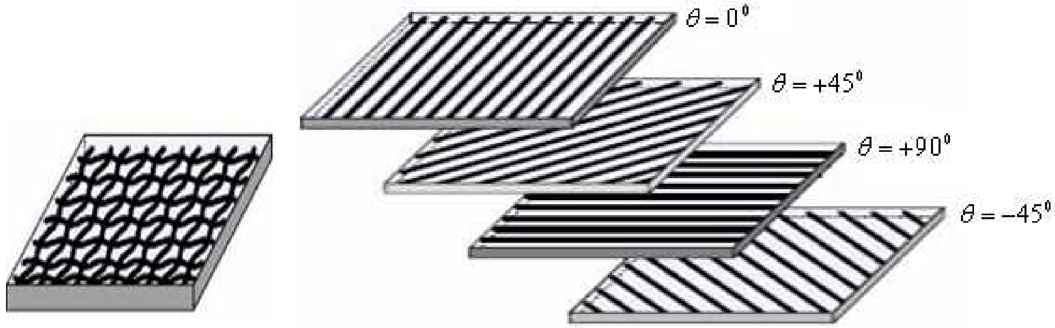
Within the aerospace and marine industries the advantages of *CMs* can be summarized as:

- Weight saving, which can be illustrated in the strength to weight ratio.
- Stronger and stiffer than metals on density basis.
- Highly corrosion resistant, especially in the most corrosive environments.
- Can be folded into many complex shapes during fabrication.
- Complicate the detection of ships and submarine in water using acoustic emission. They reduce transmitted mechanical noise from a vessel to the surrounding water.

CMs can be classified in many different ways, e.g. they can be classified based on the nature of the constituent materials. A more traditional classification than this one is derived from their forms. Figure 1.2 shows examples of *CMs* with different forms. The last form of *CMs* is so-called laminated composite materials (*LCMs*). *LCMs* are fabricated by bonding thin plates or plies of *CMs* with fibers laid in different angles [3].



(a) Bidirectional (b) Discontinuous fiber (c) Unidirectional

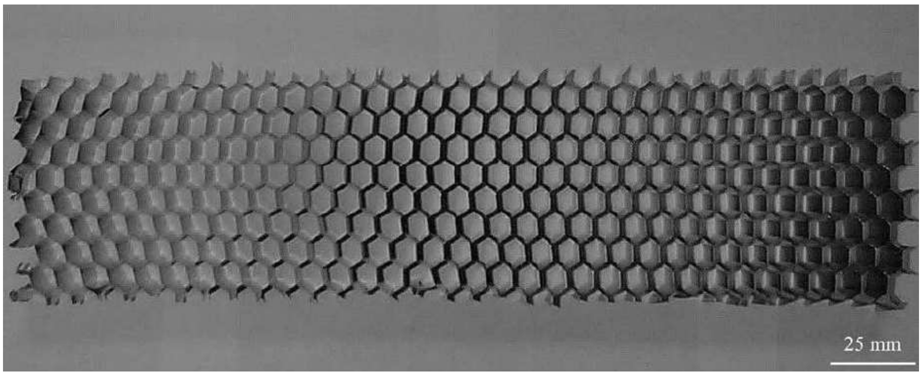


(d) Woven (e) laminated

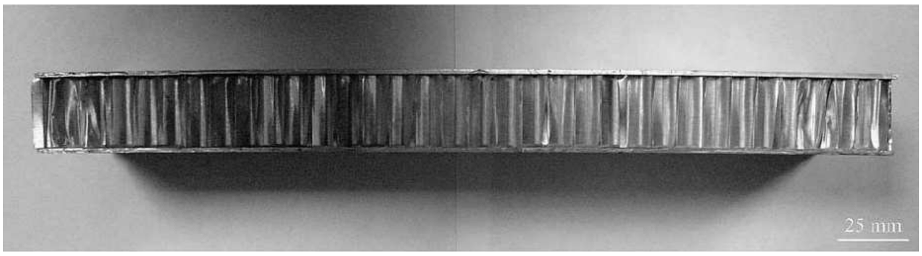
Figure 1.2: Different forms of composite materials [3]

Aluminum honeycomb sandwich (*AHS*) structures are tremendously used in a variety of engineering applications where there is a demand for strong, lightweight structural materials. Aerospace and aircraft applications primarily use the materials elastic properties (e.g. cores of sandwich panels) where stiffness and buckling are primary concerns. *AHS* plates are omnipresent on light-weight aerospace structures, such as trailing edges, spoilers and flaps. An *AHS* plate is almost four times lighter than an aluminum single plate with the same stiffness. *AHSs* have excellent properties such as high-energy absorption, fire proof, and weight saving [4] Figure 1.4 shows a top and side view of a honeycomb specimen.

Steel is widely used for roller and ball bearings. Rolling element bearings are critical components in rotating machinery, e.g. turbine engines and helicopter transmissions. Their function is to connect two machine members that move relative to one another



(a) A top view of a honeycomb specimen without face plate

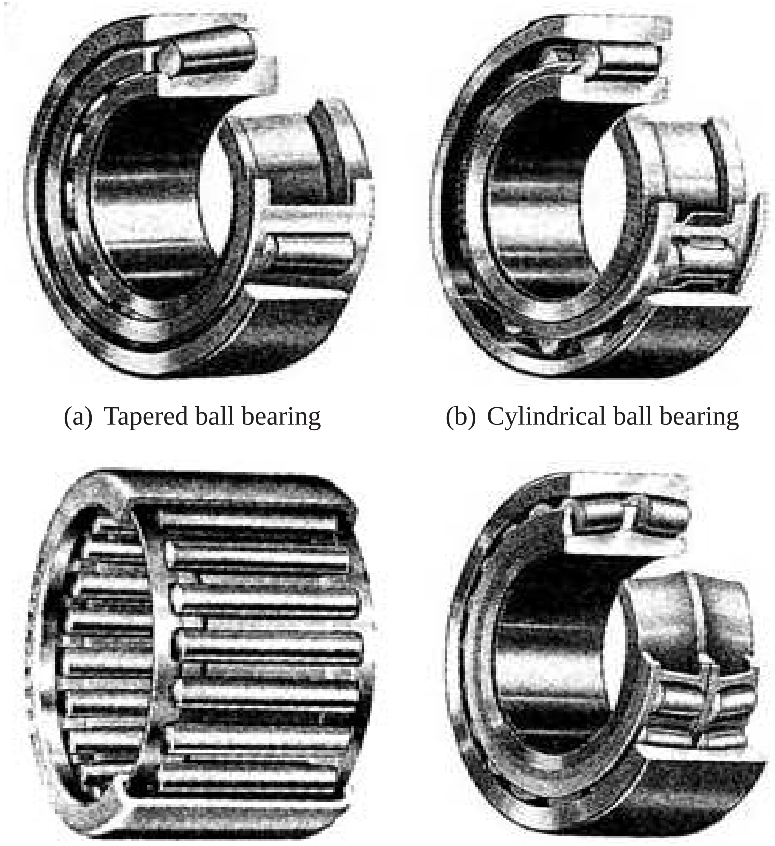


(b) A side view of a honeycomb specimen with face plate

Figure 1.3: A top and side honeycomb views without and with face plates [5]

so that the frictional resistance to motion is minimal.

In practical situations, material failure or damage may occur during manufacturing processes or in-service. The manufacturing related damages are like foreign object inclusion, porosity, and resin rich areas. In-service damages can happen in the case of aeronautical materials because a tool is dropped during maintenance, there is a bird or hail strike in plain flight, perhaps runway debris striking the aircraft during takeoff or landing. The damages have the potential of growing and leading to catastrophic loss of human life, and decrease in economy. Examples of real-life damages can be shown as airline crashes, space shuttle explosions, and building and bridge collapses. The early detection and characterization of *in situ* damages in *EMs* are very significant to ensure their structural health and integrity, prevent them from catastrophic failures, and prolong their service life [7, 8].



(a) Tapered ball bearing (b) Cylindrical ball bearing

(c) Needle ball bearing (d) Spherical ball bearing

Figure 1.4: Different types of steel ball bearings [6]

The damage process of *LCMs* is quite complex, involving both intralamina damage mechanisms (e.g. matrix cracking and fiber fracture) and interlamina damage (e.g. delamination between plies and debonding between fibers and matrix). For example, in a fiber reinforced plastic laminate, a delamination may occur between plies and propagate, eventually, leading to catastrophic failure of the structure.

In today's turbine engines and helicopter transmissions, the damage of bearing is detected by the properties of the debris found in the lubrication line when damage begins to occur. Vibration data is also used to indicate the damage of the bearing by monitoring the fundamental defect frequencies of the rolling element bearings such as the fundamental cage frequency, ball pass frequencies of the inner and outer race, and the ball spin frequency. The sensitive measurement method for early detection of the damages in ball bearing and the diagnostic techniques for evaluating the abnormal details are necessary [9, 10].

One of the potential solutions used for damage detection in *EMs* is the structural health monitoring (*SHM*). The literature defines the *SHM* as the acquisition, validation, and analysis of technical data to facilitate the life-cycle management decisions [11]. *Kessler et al.* [7] stated that *SHM* denotes a reliable system with the ability to detect and interpret adverse changes in a structure due to damage or normal operation. The intent of *SHM* system is to detect and locate damage in *EMs* and to provide this information in a form easily understood by the operators systems. Aerospace structures have one of the highest payoffs for *SHM* applications since damage can lead to catastrophic and expensive failures, and the vehicles involved undergo regular costly inspections.

There are several components required to design a successful and robust *SHM* systems for damage detection. It essentially involves implementation of a nondestructive evaluation (*NDE*) or nondestructive testing (*NDT*) technique (e.g. ultrasonic, eddy-current, acoustic emission, and radiography) to a structure to acquire data for the damage detection. Some of these techniques use actuators to propagate waves into the material. The waveforms reflected by the damages and the surfaces of the material are captured and digitized by the sensors, then will be compared to waveforms captured during the calibration of the damage detection system. In conventional *NDT*, the reflected wave shapes are visually inspected and analyzed, which is not accurate in assessing and determining the damages. The system should be able to automatically detect, locate, and assess the damage within the structure. The need for quantitative damage detection methods that can replace the visual inspection and be applied to complex structures has encouraged the *SHM* community to borrow and implement some techniques from machine learning (*ML*). The results of the waves detected by the sensors are fed into the *ML* techniques to quantitatively specify the characteristics of the damages found on the material.

1.2 Problem Statement

Bayesian network has emerged as one of the successful machine learning techniques and a generalizing graph-based framework for creating statistical models of domains with uncertainty. It has attracted a great deal of attention in research institutions as well as in industry as a good modeling tool. Nevertheless, it has not been demonstrated for damage detection in engineering materials. It is very important to demonstrate the potential of Bayesian network as a modeling tool for damage detection in engineering materials that qualify it to compete with other machine learning techniques. In addition, it is very important to show its simplicity and effectiveness to offer attractive features that are difficult to be achieved by other techniques.

1.3 Objectives

The first objective of this thesis is to introduce the Bayesian network as a probabilistic graphical model to the community of damage detection in engineering materials. This can be done by introducing the Bayesian network, describing its graphical structure, and its quantitative part in terms of damage detection factors. This will make it easier for the community to understand the Bayeisan network concept and implement it.

The second objective of the thesis is to propose a new feature extraction method based on the waves propagated to the materials using traditional nondestructive testing methods, e.g. ultrasonic. These features can be used as variables in the Bayesian network models developed for damage detection in engineering materials, which will play vital role in increasing the accuracy of damage prediction and expand its criterion.

The third objective of the thesis is to implement the Bayesian network in general and Naïve-bayes in particular as a classifier for damage detection in engineering materials. The features extracted from the second objectives will be used as an input to the classifier.

The technology of the Bayesian networks is a marriage between probability and graph theory. It is mainly developed within the machine learning community. Currently, the technology is implemented in many fields (e.g. image and voice recognition, medical diagnostic systems, and weather forecasting) and it is available in inexpensive and free software systems. The system seldom implemented for damage detection in engineering materials. Therefore, it is time to be implement the technology in this field. Thus the main contribution of the thesis is a desirable technology transfer.

1.4 Scope of the Study

The concentration of the study involves the introduction of the Bayesian networks to the damage detection community in engineering materials. The Bayesian network is introduced in the axioms of the damage detection. Since the classification is the main issue in the damage detection, the Bayesian networks is introduced as a classifier, in particular the Naïve bayes classifier. A feature extraction algorithm is proposed and tested to reduce the dimensionality of the data by decreasing the number of features and to improve the classification accuracy. The limitation of the time and scarcity of data related to the damage detection forced the research to be limited only to two sets of data. The first set of data used were *25 cm* $\times$ *5 cm* rectangular $[90/\pm45/0]_s$ quasi-isotropic laminates of the AS4/3501-6 graphite/epoxy system. Various types of damages were introduced to the specimens including, holes, fiber fracture, matrix cracking, and delamination. The thesis limits the damages to their types only and does not include damage locations, sizes, etc. Lamb waves were propagated to the specimens by using *15* and *50 KHz* frequencies. The second set of the data considered to test the Nïave bayes classifier were vibration data from a type of ball bearing operating under different fault conditions. The raw measurement data took the form of an acceleration signal recorded on the outer casing for five bearing states.

1.5 Significance of the Study

The early detection and characterization of *in situ* damages in engineering materials are very significant to ensure their structural health and integrity, prevent them from catastrophic failures, and prolong their service life. The damages have the potential of growing and leading to catastrophic loss of human life, and decrease in economy. Examples of real-life catastrophic accidents happened as a result of damages in engineering materials can be shown as airline crashes, space shuttle explosions, and building and bridge collapses.

There are many artificial intelligence techniques (e.g. neural networks and genetic programming) that have been implemented for damage detection in engineering materials. Nevertheless, the implementation of these techniques is very preliminary, limited, and does not lend itself for complex damage detection, for example, the characterization of damage types (e.g. delamination, crack, and hole in laminated composite materials). This may be due to the complex nature of the used techniques. Recently, Bayesian network is a probabilistic graphical model that has evolved as one of the most successful machine learning techniques, which has been successfully implemented in many areas. Nonetheless, it is seldom introduced for damage detection in engineering materials.

The primary goal of implementing these techniques is to be able to replace current inspection cycles with a continuously monitoring system. This would reduce the downtime of the vehicle, and increase the probability of damage detection prior to catastrophic failure. Several of artificial intelligent techniques have been implemented and tested successfully, however much work remains before these systems can be implemented reliably in an operational vehicle. The present research attempts to fill some of the gaps remaining in using these techniques.

1.6 Thesis Organization

The ultimate goal in carrying out this thesis is to introduce Bayesian networks as a classifier to the community of damage detection in engineering materials. The steps taken on the way to this are:

Chapter *2* shows the literature review. Most of the reviews are based on the works that used Neural networks as modeling techniques and natural frequency, electrical conductivity, and lamb waves as nondestructive techniques. Brief summaries of these works together with some critics are shown.

Chapter *3* presents the theory upon which the thesis is based on so as to make it as self-contained as possible. The first part of Chapter three provides an overview of the practical issues that need to be considered for the implementation and usage of structural health monitoring systems. The issues are discussed in detail with the help of an example, piezoelectric based structural health monitoring system. Issues and solutions for integration of sensors and sensors networks have been presented along with some examples. Finally, the intelligent signal processing has been shown as a key element, which builds the bridge between the sensor signal and the structural integrity interpretation. The second part of this chapter shows the Bayesian networks. It shows the theory behind the networks, different types of Bayesian network classifiers in particular the Naïve bayes classifier, and learning the Bayesian networks.

Chapter *4* illustrates the steps of the methodology used in the thesis. The steps include collecting data, discussion of a proposed feature extraction algorithm (f -FFE: f-folds feature extraction), selecting a suitable tool for the classifier, and implementing and evaluating the extracted features in the classifier. A preliminary study has been shown, which specifies the base upon which the algorithm has been developed.

Chapter *5* is intended to demonstrate the implementation of Naïve bayes classifier for damage detection in engineering materials using the features extracted by the f-FFE algorithm. The algorithm extracted features from a set of vibration data from a type of ball-bearing data operating under different fault conditions. The Naïve bayes classifier used in this study was implemented in the open-source machine learning package Weka. Two java programs have been described which implement some part of the algorithm that were not implemented in Weka.

Chapter *6* evaluates the results of Naïve bayes classifier and the f-FFE algorithm, when implemented using different number of folders and clusters. The first purpose of the evaluation is to compare the classification accuracies based on folders for all number of clusters considered in this thesis and to specify the number of clusters that give the best results. The second purpose of the evaluation is to compare the classification accuracies based on clusters for all number of folders considered and to specify the number of folders that give the best classification accuracies. The third purpose is to determine the features that give the best results.

Chapter *7* shows an overall conclusion of the thesis as well as some future recommendations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Traditional *NDT* techniques had been widely implemented for damage detection in *EMs* as qualitative diagnostic systems to assess the health and integrity of materials. As mentioned in Section 1.1, several *NDT* techniques (e.g. ultrasonic inspection, eddy current testing, *X*-radiography and acoustic emission testing) were developed and successfully implemented for a variety of applications. However, traditional *NDT* methods were not able to satisfy the increasing demands of continuous assessment of materials health and integrity while in service. Recently, there has been a tremendous growth and advancement in the technology of sensors and machine learning techniques (e.g. *NN*, Fuzzy logic, and Genetic algorithms), which can be implemented for damage detection. *NNs* were widely implemented for damage detection in engineering materials using different *NDT* techniques. Therefore, most of the review done in this thesis are based on the works that used *NNs* as modeling techniques and natural frequency, electrical conductivity, and lamb waves as *NDT* techniques.

Lamb wave (*LW*) methods have re-emerged as one of the most reliable techniques that are capable of propagating relatively long distances in some *EM* plates (e.g. *LCMs*) [14, 15]. Alternatively, electric conductivity is also widely implemented for the same purposes. This method has a long history in geology and biomedical applications in which an electric current is applied and the electric potential is recorded at monitoring electrodes around the area of study. The successful *NDT* techniques for small laboratory specimens, such as radiographic detection and *C*-scanning, are impractical for large components. Natural frequency methods are simple to implement on structure of any size. Structures can be excited by external shakers or embedded actuators, and embedded strain gauges or accelerometers can be used to monitor the structural dynamic responses [16].

2.2 Natural Frequencies

The presence of damage in some *EMs* (e.g. *LCMs*) causes changes in the physical properties of the material, which does not affect the mass distribution but reduces the stiffness of the structure and leads to changes in modal parameters (notably frequencies, mode shapes, and modal damping factors). It has also been shown that natural frequencies are sensitive to the size, location, and shape of the damage such as delaminations in structural components [17–20]. Therefore, natural frequencies can be used as indicative parameters of internal damages. Modal analysis may be used to quantify internal defects through shifts in the natural frequencies of a structure [17, 21–23]. *NN* simulations can accurately and robustly respond to dynamic characteristics of *EMs* structures and they can be used to predict the damages of *EMs*. The *NN* uses natural frequencies as input and the corresponding damage information (location, size, and shape) as an output to the network [24–27].

Smart instrumentation has been extensively tested to specify damages in *EMs* using permanent sensors as monitoring or field evaluation systems. Fiber optic sensors are attractive candidates for smart composite applications. They may also be incorporated into a composite element since their temperature tolerances and small sizes are compatible with composite processing and structure. Optical sensor data is commonly processed with *NNs* [28]. Watkin *et al.* [29] used *BP NN* and fiber optic vibration sensors to predict different sizes and locations of delaminations in composite beams. The fiber optic sensors measured the first five modal frequencies for healthy (undelaminated) and delaminated cantilever beams made of eight-ply glass/epoxy composite laminates. The delamination size and location prediction resulting from the network simulation had an average error of *5.9%* and *4.7%* respectively. Table 2.1 shows the experimental sizes, the predicted sizes, and the percentage differences between them. The results might be improved by using training data from more accurate analysis. Further studies are needed so as to obtain an efficient health monitoring capability in composite structures with integral fiber optic sensors and *NN*. The fiber optic outputs may also be fed directly into a *NN* to provide accurate information for complex structures.

Table 2.1: The prediction accuracy of delamination sizes using *NN* [29].

True Sizes (*cm*)	*NN* Sizes (*cm*)	Differences (*%*)
1.27	1.29	1.2
2.54	2.81	10.6
3.81	4.22	10.7
5.08	5.40	6.3
6.35	6.41	0.9

Chakraborty [12] introduces an approach that predicts the presence of embedded delamination (in terms of location, shape, and size) in fiber reinforced plastic composite laminates by using *BP NN* with *3* layers (input, hidden, and output). The network has been tested to predict the presence of delamination along with its size, shape, and location. It has been observed that the network can learn effectively the size, shape, and location of a delamination embedded in the laminate and can predict reasonably well when tested with unknown data set. Simulated data has been used for training and testing the network, but the approach has not been tested by using real life data sets so as to specify its actual efficiency.

Crispin and Gerard [30] proposed an approach that combines a simple but sensitive optical fiber vibration sensor, a fast Fourier transformation (*FFT*) pre-processing stage, and *BP* multi-layer perceptron *NNs* to detect damage in carbon-fiber reinforced polymers (*CFRP*). In this study two *NNs* were used, which were receiving data from four sensors fixed in the composite plates and using these information to specify the location of the damages on the plates. One network was responsible for specifying the location of the damages from the *FFTs* of strain and the second one for finding their magnitudes. The system detected the damages with an average error of *7.08%*, when data sets with simulated damages were used. In the later work, the composite panel was fitted with a number of ribs and stringers to simulate a real load-bearing *CFRP* skin structure. This made the task of loading impacts harder but a *92%* success was achieved. The system was trained successfully to differentiate between test transient signals from *CRFP* plates with four levels of damages and with three degrees of simulated impact damage.

Lew [31] introduced a novel study of optimal controller design for structural damage detection. The study was based on a neural network approach that uses the correlation of the identified natural frequency change of open-loop and closed-loop systems. In the optimal control designs, passive controllers and low-order controllers are used. The results show that the use of optimal controllers can significantly enhance the correlation difference between the damaged element and the undamaged element. This can dramatically improve the performance of damage detection. The example of low-order controllers demonstrates that the controller can be designed for both the performance of structural damage detection and also the specified damping performance. The performance of damage detection is very sensitive to sensor/actuator location.

Sahin and Shenoi [32] presented an experimentally validated damage detection algorithm using features extracted from vibration-based analysis data as input for *NNs*, for location and severity prediction of damage in beam-like structures was presented. In this work, different damage scenarios have been created by reducing the local thickness of the selected elements at different locations, and simulated vibration responses have been introduced to *NNs* with and without artificial noise during training. Sensitivity analysis has also been performed on extracted features by using different vibration modes considering the effect of damage location and severity before introducing them to *NNs*.

Yan *et al.* [33] had evaluated the ability of detecting crack damage in a honeycomb sandwich plate using natural frequency and dynamic response, and the feasibility of detecting small crack using method proposed. It has been found that using structural natural frequency may not be suitable for detecting crack damage less than 10%, even up to 20% of the total size of a plate-like structure. Besides, it is very difficult to determine the location and category of crack damage with such a dimension. However, energy spectrum of wavelet transform signals of structural dynamic response has higher sensitivity to crack damage, it can exhibit structural damage status for a crack length close to 2% of the dimension of a plate-like structure. It has been also found that structural damage information is often contained in some high order modes of a structure, and more vibration modes should be included in structural dynamic model for detection of small damage.

2.3 Electrical Conductivity

The implementation of natural frequencies as indicative parameters for damage detection in *EMs* has the disadvantage that sometimes, the measurement of the frequencies is very difficult due to some limitations like the connectivity associated with the sensors (e.g. space and bandwidth restrictions) and external noises. Another approach to identify the damages is embedding fiber-optic strains into the materials so as to measure the strain distribution [35, 36]. Unfortunately, this may reduce the static and fatigue strengths, and increase the total weight of the material. In addition, the optical fiber sensors and the sensing systems are very expensive. These guide to another form of smart technology to identify damages in *EMS*.

Some of the materials used in the *LCMs* are electrical conductors, e.g. carbon and graphite fibers. Therefore, the measurement of the electrical resistance appears to be a valuable technique for the detection of different types of damages in *CFRP* laminates, which does not cause reduction of static strength or fatigue strength. Moreover, the electric-potential method does not cause increase in weight. This method has been adopted by many researchers, e.g. Irving and Thiagarajan [37], and Abry *et al.* [38]. In the case of *CFRP*, the carbon fibers are not only used as a reinforcement material, but also as sensors of damage detection [39]. Dae-Cheol and Jung-Ju [39] investigated this kind of damage detection by mounting electrodes on the surface of the *CFRP* structures. They showed that the measured stiffness change had a similar trend as the electrical resistance change during fatigue tests. The electrical resistance showed gradual increase while the stiffness was decreasing and showed an unexpected change when the final fatigue stiffness changed suddenly. They used *NN* to investigate the relation between the electrical resistance damage parameter, fatigue life, and stiffness reduction, which showed good relationship (Figure 2.1).

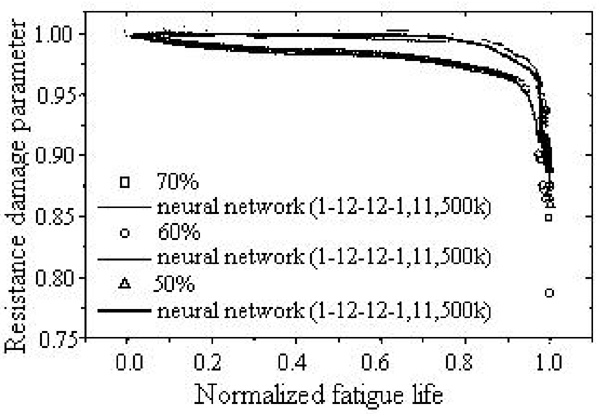


Figure 2.1: Comparison of fatigue life and resistance damage parameter [39].

Figure 2.1 shows the relationship between electrical resistance and fatigue life using *NN*. The input node of the *NN* was the electrical resistance damage parameter and the output node was the fatigue life of stiffness reduction. Three stress levels, 70%, 60%, and 50% of the average static ultimate strength were selected. The error convergence of the network relies on the structure of the hidden layer. In this case, it showed better results with two hidden layers than with one hidden layer. About *11* to *18* numbers of experimental data were used as learning input data. After the learning step, a graph very similar to the experimental results was acquired. Thus, it was possible to predict specimen damage by monitoring electrical resistance using *NNs*.

Graphite fibers in graphite/expoxy laminated composites are also very good electric conductors and the epoxy matrix is an insulator. Generally, electric conductivity is very high in the direction of the fibers and much lower in the transverse direction of the fibers or may vanish under normal conditions [40]. When a delamination grows between plies in a graphite/expoxy composite, the electrical resistance increases in the composite. Therefore, delaminations can be detected by calculating the variation of electric resistance in this kind of composites. Todoroki and *et al.* [41–44] showed that electrical resistance change method using response surfaces was very effective in identifying delaminations in laminated composite materials both experimentally and analytically. They proposed a schematic representation of a delamination monitoring system (Figure 2.2).

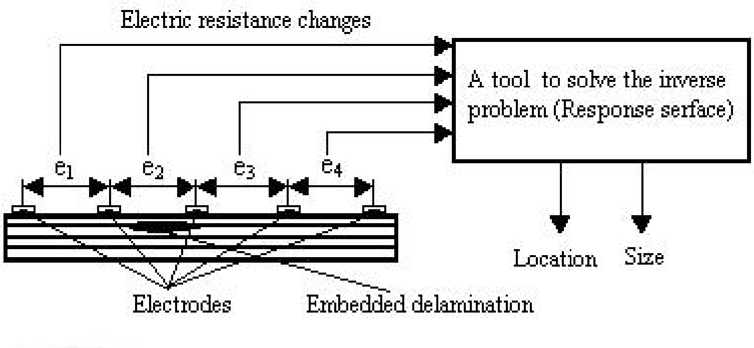


Figure 2.2: Delamination identification using electric resistance [41].

In Figure 2.2, multiple electrodes were mounted on the surface of a specimen with equal spaces from each other. All of these electrodes were placed on a single side of a specimen. Usually it is impossible to place electrodes and lead wires outside of the aircraft structures. Mounting of electrodes on the singe side surface represents modeling of electrode attachment in thin aircraft shell type structures. Electrical resistance change of each segment between electrodes was measured for various cases of location and size of delaminations. Using the measured data, the relationships between electrical resistance changes and delaminations (location and length) were obtained using the response surfaces. The response surface methodology comprises regression curve fitting to obtain approximate responses, design of experiments to obtain minimum variances of responses, and optimizations using approximated responses [43, 44]. The main draw back identified for this method was the high number of experiments that must be performed to obtain sufficient number of sets of electrical resistance changes. The response surface was similar to *NNs* and it was a widely adopted tool for quality engineering fields.

2.4 Lamb Waves

Sir Horace Lamb was the first to introduce the Lamb wave (*LW*) in *1917*. *LW* is one of the widely used techniques in *NDT* for damage detection in *EMs*. *LWs* are acoustic waves that can be launched into relatively thin solid plate with free parallel surfaces and are also known as plate waves [46]. There are different kinds of techniques used to propagate and receive *LWs*. These techniques have been implemented in a variety of configurations, including the use of single purpose devices (e.g. transducers) that use separate actuators (sources or transmitters) and sensors (receivers) to propagate and monitor the propagated waves and/or reflected waves individually, and multipurpose transducers in which, a single transducer is used to actuate and sense the waves simultaneously. The simplest methods of the multipurpose transducers use piezoelectric transducers. The single purpose transducers are like laser transmitters and optical fiber sensors. Each of these techniques has its own unique properties and different analytical complexity in detecting and predicting specific types of damages in *EMs*.

The *LWs* generated by a transmitter propagate through the material and reflected by damages and the surfaces of the material back to the sensors. The signals reflected to the sensors contain some information (e.g. size, location, and orientation) about the damages and they can be used to test the structural integrity of the material. *LWs* excite the whole volume of the structure along the line between the transmitter and receiver. They can propagate over long distances. However, their dispersive nature and the existence of many modes simultaneously can complicate the interpretation of the acquired signal [46].

Worlton was the first to introduce the implementation of *LWs* for damage detection in *1960*. He investigated the dispersion curves of aluminum and zirconium to describe analytically the characteristics of the various modes that would pertain to nondestructive testing applications. During the late *1980* and *1990*s, work began on the application of *LW* to composite materials [49].

A sensor using metallic multi-electrodes deposited on a piezoelectric substrate has been especially designed and developed bye Rguiti *et al.* [47] in order to detect Lamb waves generated by *PZT* transducers in aluminum plates. The process used for this device fabrication is the tape casting technique, which is adapted to manufacture large and thin piezoelectric sheets. The resonance method was used to characterize the *PZT* material. This study demonstrated some significant improvements when using this sensor compared to the multi-element array [48]. First of all, the integration of this sensor becomes easier, since only one component has been realized and stuck on the structure to be monitored. Secondly, the use of only one piezoelectric substrate allows to get an equivalent electric response on each electrode, which eases the conditioning of the signal to be processed. Thirdly, the problem of possible conversions and reflections on each element that can be observed when using the multi-element array is now solved.

Finally, the sensor sensitivity to Lamb wave variations in presence of damage has been demonstrated

Many researchers have adopted the *LWs* together with *NNs* as a technique for damage detection in *EMs*. Su and Ye [13] demonstrated a *LW* propagation-based quantitative identification scheme for delamination in *CFRP* composite structures by using a multi-layer *BP NN*. An Intelligent signal processing and pattern recognition package was developed to perform the identification, where a *BP* was trained using spectrographic characteristics extracted from acquired *LW* signals. Excellent quantitative diagnosis results for damage parameters in terms of presence, location, geometry, and orientation were achieved. Although a certain amount of time is inevitably spent on the preliminary off-line development of the *NN*, the researchers did not test the developed *NN* and the structural health monitoring system to diagnose an actual damage performed instantly online.

Yuan *et al.* [49] introduced a damage signature based on wide-band *LW* for on-line delamination and impact detection monitoring system applied to honeycomb sandwich and *CFRP* structures. The damage signature was introduced together with a Kohonen *NN* to determine the presence and extent of damage in the composites, while diminishing the influence of different distances between the transmitters and sensors. They showed the efficiency and the reliability of the proposed method for the different types of the materials used, which suffer various levels of damage.

Su and L.Ye [50] developed an approach to locate structural damage (e.g. delamination or through-hole damage) in CF/EP composite laminates, using digital damage fingerprints (DDF) extracted from raw Lamb wave signals. A multi-layer feed-forward *NN* was designed and trained under the supervision of an error-backpropagation algorithm. Assisted by an active online structural health monitoring)system, the methodology was validated by locating the actual delamination and through-hole damage in

CF/EP (T650/F584) quasi-isotropic composite laminates. The results showed that such an approach is able to predict the damage location much more quickly. Exponential decreases in time consumption for *NN* training and in requirements for computational tools have been achieved, without any decrease in prediction precision. It was proposed that this approach is more practical and economical for low cost engineering applications.

2.5 Feature Selections

The identification of features receives great attention in the damage detection. Feature selection is the minimum portion of the data that allow one to distinguish between different types of the damaged and undamaged components or systems. The best features for damage detection are typically application specific. A variety of methods are employed to identify features for damage detection. This is illustrated in more detail in Chapter 3.

Lindh *et al.* [52] introduced a new automatic analysis method for the detection of cyclic bearing faults. The method uses a multivariate statistical fault classification and fuzzy logic. Features are extracted from an envelope spectrum of the frame acceleration of a motor frame. The features are created from the coefficients of the envelope spectra calculated from the motor frame acceleration signal. The expected bearing pass frequencies (cited in [52]) for different fault types are calculated and the peaks near to these calculated values are selected as features. A bearing fault feature vector consists of 16 components of a signal envelope at selected frequencies. The selected components represent the characteristic fault frequencies and their three nearest harmonic frequencies for all the following faults: outer race fault, inner race fault, rolling element fault, and cage fault. The classification results are presented in Tables 2.2 and 2.3.

Table 2.2 presents the classification results using a *16*- dimensional feature space, cov-

Table 2.2: Classification results using *16*- dimensional feature space [52].

healthy	outer race	inner race	ball	spin	cage
healthy	0.6981	1.3607	2.7905	1.6052	2.3892
outer race	5.3817	1.2018	7.0394	8.7226	7.5465
inner race	2.9616	3.2331	1.0653	2.7675	2.7684
ball spin	6.5451	5.3318	8.2085	1.0579	6.7352
cage	0.7758	1.4425	0.8656	1.7417	0.1666

Table 2.3: Classification results using four dimensional feature space [52].

test data	distance to	outer race	inner race	ball spin	cage
healthy	healthy	0.1767	0.148	0.1107	0.138
	broken	0.781	4.235	1.2339	1.3203
outer race	healthy	3.9542	0.2946	0.2454	0.2164
	broken	0.0394	6.5133	1.4492	1.4779
inner race	healthy	0.556	2.4608	0.441	2.0332
	broken	1.385	0.3182	1.6345	2.656
ball spin	healthy	0.2327	0.3109	4.4202	0.1176
	broken	0.9328	4.1784	0.2219	1.7265
cage	healthy	0.2827	0.3915	0.2275	4.0238
	broken	1.0563	5.0148	1.7709	0.073

ering all four fault types. Table 2.3 presents the classification results using four dimensional feature space. Both indicate only correct classification results. The statistical distance between healthy and broken cases is bigger when four dimensional feature space is used. On the other hand, there is a bigger risk of misclassification if the shape of the test feature vector changes. It is important to bear in mind, firstly, that the correct classification was obtained without any tuning of the prototype vectors, and secondly, that there can be many other fault modes that were not taken into account and much more research work should be done with various fault types and motors before jumping into conclusions that generalize the result obtained with these tests. The results obtained in this study clearly demonstrated the advantages of introducing both quantitative and qualitative features in the calculation as well as combining the statistical classification and fuzzy logic.

Reddy and Ganguli [53] presented a *NN* approach for the detection of structural damage in a helicopter rotor blade using rotating frequencies of the flap (transverse bending), lag (in-plane bending), elastic torsion and axial modes. A finite element method was used for modeling the helicopter blade. Several combinations of modes are investigated for training and testing the *NN*. Using the first *10* modes of the rotor blade for damage detection yields accurate results for the soft in-plane hingeless rotor considered in this study. Using a parametric study of the blade rotating frequency in conjunction with the *NN*, it was found that a reduced measurement set consisting of five modes (the first two torsion modes, the second lag mode and the third and fourth flap modes) also gave good results for damage detection. Furthermore, taking only the first four flap modes also resulted in good damage detection accuracy.

2.6 Summary

All above mentioned researches lend valuable insights to the problems association with the damage detection in engineering materials. There are different gaps need to be filled. The first gap is concerning the implementation of Bayesian networks. Different machine learning techniques have been implemented for damage detection in engineering materials. The Neural networks have greatly attracted the attention of the researchers more than the other techniques and have been widely implemented for the damage detection. However, one of the main drawbacks of Neural networks is the number of hidden layers, which must be specified in advance. Nonetheless, some of the researchers think that one or two hidden layers is enough to simulate any damage detection system. Most of the studies carried out do not lend themselves for complex damage detection systems. They detect the damage, but do not tell exactly what is there (e.g. the type of the damage, the size, etc.). Generally, the implementation of all machine learning techniques in the damage detection were very primitive, this might be due to the complex nature of these techniques.

The second gap is concerning the method used for feature extraction and selection. Most of the feature selection and extraction methods employed for damage detection have been borrowed from other fields and it is seldom to find any technique, which is developed based on the damage detection background. They been have borrowed and implemented for specific types of materials and nondestructive testing techniques, which made them difficult to be generalized for other materials and nondestructive techniques. The features extracted and selected by some of these techniques have limitations in representing the whole input data. For example, the peaks of input waves used for damage detection can be guaranteed to represent the whole waves, since more than on groups of waves can give the same or similar peaks. Therefore, it is important to develop an new feature extraction techniques for the damage detection.

CHAPTER 3

Theory Background

The objective of this chapter is to present the theory upon which the this research is based so as to make the thesis as self-contained as possible. All relevant theories are explained to the level at which it is used in the subsequent chapters.

Structural health monitoring systems and Bayesian networks play important role in this thesis. In the first part of this chapter, the definition of the structural health monitoring systems is presented and some of their required components are presented, including sensor and actuator elements, vibration and analysis methods, and intelligent signal processing. In the second part the Bayesian networks are presented in terms of damage axioms, the representation of the networks, Bayesian networks as classifiers, and learning the networks are presented.

3.1 STRUCTURAL HEALTH MONITORING SYSTEMS

The damage in *EMs* can be described with a number of different terms, such as health and monitoring of structures. Intuitively, health is the ability to perform and maintain the structural integrity throughout the entire lifetime of the material, monitoring is the process of diagnosis and prognosis, and damage is the failure of material. In this context, the meaning of the damage detection is the same as that of the *SHM* and the *SHM* is a safety issue. *SHMs* can be considered as a real-time (continuous) and discontinuous system. A real-time *SHM* system is one that continually monitors a structure during operation, and produces data that can be directly utilized at any point by either an operator or ground control station (see Figure 3.1). A discontinuous *SHM* system is one that can only be accessed post-operation and could contain either a stored record of operational health data or might involve performing an integral inspection upon demand [56].

Figure 3.1: A real-time system to monitor a plane structure

There are a number of different approaches to specify the *SHM*. One of these approaches is based on direct visual observations and various physical phenomena. This approach is limited to single-point measurements but allows surface scanning if a complete structure is considered to be analyzed. Another approach, indirectly relates various parameters or symptoms to possible structural conditions. This approach monitors for damage globally and does not require single-point measurements. Examples include the usage of impact damage detection in *CMs*. Both approaches utilize the relationship between the symptom of damage and the structural damage condition. Different physical models and system identification procedures are used to establish this functional relationship. More recently artificial intelligent methods have been used to solve the problem (e.g. neural networks, fuzzy-logic, etc.). In this context, the group of methods based on the symptom and damage relationship is commonly known as *SHM* [54].

There are several components required to design a complete *SHM* system, including sensor and actuator elements, processing and communication chips, a power supply, and some form of packaging to integrate and protect these components. The *SHM* determines the condition of the monitored structure by examining the output of sensors attached to or embedded in the material to form an integral part of it. This may involve measuring strain values or vibration characteristics at different points in the material. The establishment of relevant parameters used to monitor or detect damage is one of the major problems in this area as well as the prediction of subsequent damages. All aircraft structural health and usage monitoring approaches consider stress as symptom used for damage monitoring. Levels of stress can be estimated relatively easy from load models, strain, or flight parameters [54].

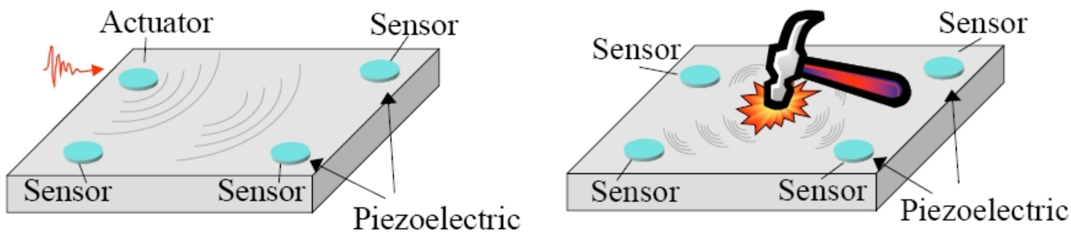
The requirements placed on *SHM* are strict if they are to replace current inspection methods performed by skilled labors. Considering the demands placed on the sensors, they must be sufficiently sensitive to detect indicators of damage. They must be stable, durable and reliable for the lifetime of the structure they are monitoring. Reliability is crucial. If automatic systems are to replace human inspections we must have absolute confidence in them. They should not produce too many false positives and certainly should not miss damage features they are supposed to detect. Many initial installations of integrated *SHM* will have to prove their worth alongside conventional inspection techniques. They must also be cost-effective. That is, the cost of installing the system must be less than an equivalent inspection regime (unless the safety or performance benefits outweigh the cost) and the sticker price should not deter potential users. Conventional measurement systems simply cannot satisfy all these requirements simultaneously.

3.1.1 Vibration and Modal Analysis

Damage can be often considered as a modification of physical parameters such as mass, stiffness or damping. A number of vibration-based parameters have been used for structural health monitoring. The application of modal analysis is one of the most popular approaches since the classical work on the use of natural frequencies for damage detection in structures. Previous studies show that modal shapes and damping can also be used to detect damage. Other applications in this area involve modal energy, curvatures, and transfer functions. Vibration-based data have been employed with some success to detect aircraft structural damage. However, the major problem in this area is related to damage sensitivity. Modal and vibration based techniques are in fact global methods. A number of studies have been performed on beams and plates where cracks originated from the specimen's surface perpendicular to the applied normal stress. However, very long cracks or delaminations are required to affect the structural physical and/or modal parameters in the case these cracks and delaminations are parallel to the loading direction. Despite different reports on successful crack detection, the ability of vibrational/modal techniques for damage inspection in aerospace structures becomes somewhat questionable and leads the ongoing discussion on global and local monitoring. Experimental results show that the size of damage (e.g. delamination in *CM*) must be at least 10% of the area monitored to be reliably detectable [54].

3.1.2 Monitoring Techniques and Sensor Technology

Sensors are used to record variables such as strain, acceleration, sound waves, electrical or magnetic impedance, pressure or temperature. Marantidis *et al.* [57] have estimated that a *SHM* system for an aerospace vehicle would require between *100* and *1000* sensors, depending on its size and desired coverage area. Sensing systems can generally be divided into two classes: passive or active sampling (see Figure 3.2).



(a) Active sensing (b) Passive sensing

Figure 3.2: Active and passive sensing systems used by piezoelectric materials [55]

Active sampling systems 3.2(a) are those that require externally supplied energy in the form of a stress or electromagnetic wave to properly function, they provided their own source of energy. A few strain-based examples of active systems include electrical and magnetic impedance measurements, eddy currents and optical fibers which require a laser light source. Passive techniques tend to be simpler to implement and operate within a *SHM* system and provide useful global damage detection capabilities, however generally active methods are more accurate in providing localized information about a damaged area. Passive sampling systems 3.2(b) are those that operate by detecting responses due to perturbations of ambient conditions without any artificially introduced energy (e.g. strain measurement by piezoelectric wafers), they rely on energy emitted from other sources[56].

3.1.2.1 Smart Structures and Materials

Materials and structures, which are able to sense and perhaps respond to a change in their environment are commonly known as smart. Smart structures and materials have opened new opportunities for damage monitoring. In general damage monitoring systems, which utilize smart structures and material technologies are concerned with a design philosophy directed to the integration of actuators, sensors, and signal processors. The attractive potential of such technologies arise from the added value in terms of more reliable damage monitoring systems, reduced inspection monitoring cost and improved safety. The last ten years have seen an enormous mount of research in this area. This includes new materials (piezopolymers, piezoceramics), sensors, and actuators (*MEMS*, Micro-surface acoustic waves - *MSAW* devices) and intelligent data processing (pattern recognition, data fusion, neural networks, combinatorial optimization based biological and physical systems, and much more).

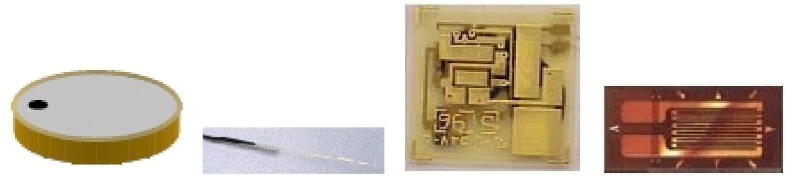
3.1.2.2 Damage Detection Techniques

In principle all the *NDT* techniques mentioned before can be considered as implemented onto or into a component to be monitored, which in the end is already some initial type of smart structure. With regards to simplicity and availability of sensing and possibly also actuation elements, piezoelectric elements have turned out to be one of the types being highly viable. The acousto-ultrasonic technique therefore looks to be one of the very promising techniques to start with. It is based on stress waves introduced to a structure by a probe at one point and sensed by another probe at a different position. The frequency of these waves can go up to *MHz*. Various types of signal are used as input excitation including impulse, sine burst, sine sweep, and Gussian white noise signal. Damage in a structure can be identified by a change of the output signal. Often attenuation is sufficient to detect defects.

Lamb wave inspection is based on the theory of guided waves propagating in palates. In general, the principles of acousto-ultrasonic and Lamb wave inspections are similar. Also, signal processing used for damage detection in similar and is often based on wave attenuation and/or wave dispersion. The factors, which determine the Lamb wave inspection, are related to properties of the structure under inspection and transducer schemes. Other important elements, which form the monitoring strategy include various aspects related to transducer coupling methods, types of excitation signals, optimal sensor location, sensor validation, and intelligent signal processing.

3.1.2.3 Sensor Technologies

Various sensor technologies are currently available which can be either adapted onto or integrated into the structure to be monitored. These include, piezoelectric, optical fiber, micro-electro-mechanical systems (*MEMS*), and strain-gages sensors, etc. (see Figure 3.3). The maturity and networking capability of the various sensor types depends on the conditions of usage and structural application.



(a) Piezoelectric (b) Fiber-optic (c) HEMS (d) Strain-gages

Figure 3.3: Different types of sensors used for structural health monitoring [55]

There are several issues involved in the practical usage and implementation of structural health monitoring systems using the sensors defined above. These include [55]:

- Sensor integration
- Calibration
- Reliability
- Effect of environmental conditions

These practical issues are discussed by using an example of piezoelectric materials. Piezoelectric sensors and actuators are made of piezoelectric materials (piezo-crystals, ceramics, and polymers). Materials that have a piezoelectric effect convert mechanical force to electrical charge, and vice versa. Hence, piezoelectric materials can be used as both sensors and actuators. As sensors, they produce an electrical signal when they are physically deformed (strained). As actuators, they physically deform (expand, contract, or shear) when an electrical charge is applied. Using this property, piezoelectric materials can be used to measure stress and strain and can also be used to mechanically excite the structure to propagate stress waves and induce internal vibrations. Inputting a time-varying electrical signal to any of the actuators/sensors causes a propagating stress wave or propagating mechanical deformation to emanate from the sensor/actuator and travel through the material for detection by a plurality of neighboring sensors/actuators.

A lot of development work has been done in the area of optical fiber sensors. The major advantage of these sensors is their immunity to electromagnetic fields and their compatibility with data transmission systems. However, more work needs to be done in this area regarding material integration and reparability procedures. Optical fiber sensors have been used for monitoring the curing process and/or damage induced by impact and overloads in *CMs*. Optical fiber sensors are also increasingly used for strain and temperature measurements. Recent development in this area shows applications of Bragg-Grating sensors for acousto-ultrasonic monitoring. It is quite feasible that multi-functional optical fibre sensors will be soon available for both strain and damage monitoring.

Piezoelectric materials have been used for years for actuating and sensing stress waves. However, only recently these materials have become available in the form of ceramic sensors are also available on Kapton layers in the form of so called smart layers, which can be embedded or bonded on structural components and here specifically in areas prone to damage such as notches. A variety of sizes and shapes for these sensor layers can be made available and basically tailored according to customer needs.

Actuating and sensing for active damage detection can be accomplished using other new technologies such as interdigital transducers, phase array transducers, piezoelectric paints, and *MEMS*.

As mentioned above, the *PZT* can be used in dual sensing modes, passive and active. In the passive sensing mode, the structural health monitoring system:

1. Finds location of impacts

2. Records date/time of occurrence

3. Determines impact force/energy (to predict structural damage)

No calibration is required for impact location. However, in order to determine the impact force/energy, calibration with known impact forces is required. This can be typically done with the use of an instrumented hammer as shown in Figure 3.4.

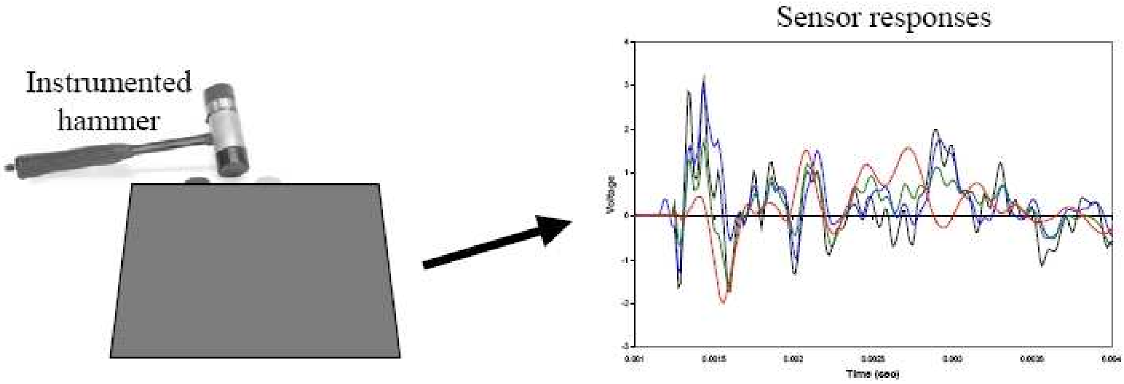


Figure 3.4: Calibration of impact force with instrumented hammer [55]

3.1.2.4 Intelligent Signal Processing

Intelligent signal processing is the key element, which builds the bridge between the sensor signal and the structural integrity interpretation. Various methods have been developed in recent years. This includes: data pre-processing techniques, outlier analysis, feature extraction and selection (e.g. signature analysis, time-frequency analysis, wavelets) and pattern recognition (e.g. neural networks, novelty detection). All these methods lead to signal features, which are sensitive to damage but insensitive to boundary, load or environmental conditions.

Sensors are usually deployed in arrays. Multi-sensor architectures not only improve the signal to noise ratio but also offer better robustness, reliability and confidence in the results. Sensor data and information (e.g. flight parameters) can be combined using various fusion techniques such as physical models, parametric methods, information techniques and cognitive-based models.

The fewer sensors need to meet the requirements set for structural health monitoring, the better the overall reliability, signal processing effort and thus smaller cost for the damage monitoring system. The optimal sensor number and their locations can be established using various combinatorial techniques and mutual information approach. Sensor architectures also require validation procedures, which are important to detect sensor failures. These exist various methods in this area based on statistical analysis and neural networks.

3.2 BAYESIAN NETWORKS

Bayesian networks (*BNs*) have evolved as a powerful probabilistic graphical modeling tool, which encodes probabilistic relationships among variables of interest under domains of uncertainty. During the 1990s, they have attracted a great deal of attention from research communities as well as from industry [59]. They are widely used as a modeling tool for diagnosis, analysis, and decision making in real world of uncertain domains., e.g. modeling knowledge in gene regulatory networks, medical diagnostic systems, text analysis, and image processing. They do not necessarily require a commitment to the Bayesian methods. They are so called because they use Bayes' rule for probabilistic inference (it is articulated later in this section).

The following quotation [60] gives a very concise introduction to graphical models.

> *Graphical models are a marriage between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering −− uncertainty and complexity −− and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms. Fundamental to the idea of a graphical model is the notion of modularity −− a complex system is built by combining simpler parts. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data. The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.*

> *Many of the classical multivariate probabilistic systems studied in fields such as statistics, systems engineering, information theory, pattern recognition and statistical mechanics are special cases of the general graphical model formalism examples include mixture models, factor analysis, hidden Markov models, Kalman filters and Ising models. The graphical model framework provides a way to view all of these systems as instances of a common underlying formalism. This view has many advantages in particular, specialized techniques that have been developed in one field can be transferred between research communities and exploited more widely. Moreover, the graphical model formalism provides a natural framework for the design of new systems.*

BNs as graphical models have several advantages for data analysis, when used in conjunction with statistical techniques [72]:

1. They handle situations where some data entries are missing, because the model encodes dependencies among all variables. For example, consider a classification problem where two of the input variables are strongly anti-correlated. This correlation is not a problem for standard supervised learning techniques, provided all inputs are measured in every case. When one of the inputs is not observed, however, most models will produce an inaccurate prediction, because they do not encode the correlation between the input variables. *BNs* offer a natural way to encode such dependencies.

2. They can be used to learn causal relationships, and hence can be used to gain understanding about a problem domain and to predict the consequences of intervention. Learning about causal relationships are important for at least two reasons:

   - The process is useful when trying to gain understanding about a problem domain, for example, during exploratory data analysis.

   - Knowledge of causal relationships allow making predictions in the presence of interventions. For example, a marketing analyst may want to know whether or not it is worthwhile to increase exposure of a particular advertisement in order to increase the sales of a product. To answer this question, the analyst can determine whether or not the advertisement is a cause for increased sales, and to what degree. The use of *BNs* helps to answer such questions even when no experiment about the effects of increased exposure is available.

3. They are an ideal representation for combining knowledge (which often comes in causal form), because the model has both a causal and probabilistic semantics. *BNs* have a causal semantics that makes the encoding of causal prior knowledge particularly straightforward. In addition, *BNs* encode the strength of causal relationships with probabilities. Consequently, prior knowledge and data can be combined with well studied techniques from Bayesian statistics.

4. Bayesian statistical methods in conjunction with *BNs* offer an efficient and principled approach for avoiding the over-fitting of data. Using the Bayesian approach, models can be ”smoothed” in such a way that all available data can be used for training.

This chapter introduces the *BNs* by discussing the representation of the network, how can the hidden states of a system efficiently be inferred given partial and possibly noisy observation (the inference), how the parameters are estimated and the the networks are constructed (learning), and what happens when it is time to convert beliefs into actions. All of the examples shown in the section are related to damage detection in *EMs*. The chapter discusses also different types of *BNs* classifiers, methods for learning both the parameters and structure of a *BN*. In addition, the relation between *BN* methods for supervised and unsupervised techniques are also elaborated.

3.2.1 Representation of the Networks

Probabilistic graphical models (*PGMs*) compose of nodes and arcs (links) between nodes. The nodes represent random variables, and the lack of arcs between two variables represent conditional independency. Generally, *PGMs* can be divided into two kinds:

1. Undirected graphical models, which are popular in physics and vision communities. They are also known as Markov networks or Markov random fields. The links in these models have no direction.

2. Directed graphical models, which are known as Bayesian networks. They are more popular in machine learning and Artificial intelligence.

A third type of models can also be considered, which is called a chain graph. This type of models contains both directed and undirected links.

BNs formalism provides a powerful framework for the modeling of uncertain interactions among random variables in a domain. They are a brief representation of a joint probability distribution on a set of statistical variables [61]. *BNs* consist of a qualitative part, where features from graph theory are used, and an associated quantitative part consists of potentials, which are real-valued functions over a set of variables from the graph. The variables can take discrete and continues values. In this thesis only the variables with discrete finite valued attributes are considered.. The structure of *BNs* can be show as follows [62]:

- A network structure $G = \{V, E\}$, where $V = \{V_1, V_2, \ldots ,V_n\}$ represents a set of n variables and E represents a set of directed arcs between the variables.

- Each variable has a finite set of mutually exclusive states.

- A set of conditional probability tables (*CPTs*) associated with each variable.

The directions of the arcs in *BNs* often represent causal dependency between variables. The variables in an *BN* represent events in a domain. These events are connected with directed links. A link represents a causal relationship between the events and starts from the causal event. A causal network is a set of variables and directed links between the variables [58]. In *BNs*, a variable is a parent of a child, if there is an arc from the former to the later. *BNs* model the quantitative strength of the connections between them, allowing their probabilistic beliefs to be updated automatically as new information arrive. The arcs in any *BNs* are not permitted to be directed cycles, one cannot start from a variable and simply come back to it by following the direction of the arcs in the network. For this reason the networks are known as directed acyclic graphs (*DAGs*) [59, 62].

It is worth noting however, that in some applications where the amount of training was very limited, but a priori information about the spectra is available, the use of *BNs* may be useful.

The values of each variable should be mutually exclusive and exhaustive, that means the variable must take on exactly one of these values at a time. For example, if someone consider building a model to predict the presence of a damage in an *EM*, many factors might be taken into account, e.g. the age of the material (*Age*) and whether a tool dropped on the material (*ToolDrop*). These factors can be represented as variables in the model connected by directed links according to the direction of impacts (see Figure 3.5).

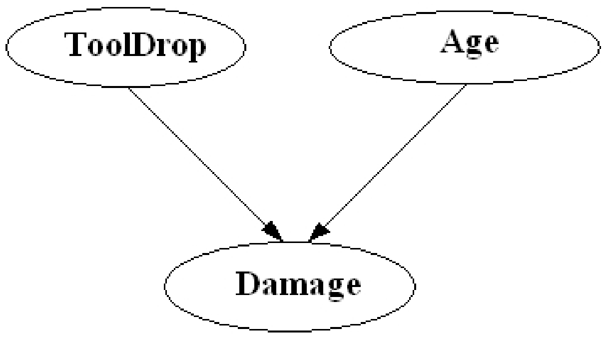


Figure 3.5: A small *BN* structure for damage detection in an *EM*

In the figure, the variables *ToolDrop* and *Age* have an impact on the variable *Damage*. That means the presence of the damage can be determined by the states of *ToolDrop* and *Age*. No one can argue that the damage was the cause of dropping the tool on the material or has an impact on the age of the material. Every variable can take one of a different type of discrete values (the states of the variable). The variables *Damage* and *ToolDrop* might be represented by states, which take boolean values *yes* and *no*. The variable *Age* might be represented by states that take ordered values, *new*, *medium*, and *old*.

3.2.2 Causal Networks and *d*-Separation

A causal network is a directed graph, which consists of a set of variables and a set of directed links between variables. It can be used to follow how a change of certainty in one variable may change the certainty of other variables. In order to understand that a set of rules for reasoning under uncertainty are illustrated bellow. These rules are independent of the particular calculus of uncertainty [59].

3.2.2.1 Serial Connections

Consider the causal chain of the three variables shown in Figure 3.6, where *Age* causes *Damage* which in turn causes *Amplitude* (causes the value of the wave's amplitude to increase).

Figure 3.6: Serial Connection.

In the figure, if we do not know whether the material has damage, but we are sure that it is an old material (*Age* = *old*), that would increase our belief that the material has damage and the amplitude would be high. However, if we already know that the material has damage, then the age of the material would not make any change to our belief about the amplitude. That means if we know the stage of the variable *Damage*, then evidence is blocked for transmission from *Age* to *Amplitude* (*Age* and *Amplitude* become independent). The conditional independence of this causal chain can be shown as follows:

$$P\ (Amplitude \mid Damage, Age) = P(Amplitude \mid Damage)$$

In this case, it can be said that *Amplitude* and *Age* are *d*-separated given *Damage*. The *d* in the *d*-separation stands for dependency. This kind of connection is known as serial connection in which evidence may be transmitted through it unless the state of the variable in the connection is known.

3.2.2.2 Diverging Connections

In Figure 3.7, the variable *Damage* has a cause on the two variables *Amplitude Value* and *Ultrasonic Velocity*. If there was damage in a material, the amplitude value would be increased and the ultrasonic velocity would be decreased.

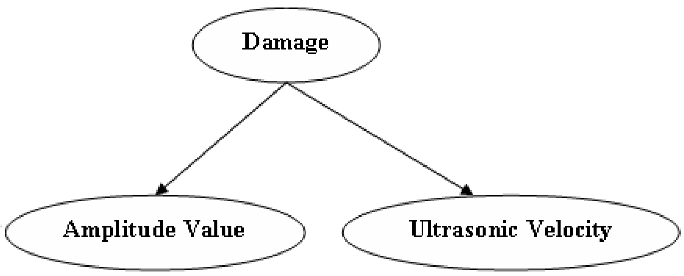


Figure 3.7: Diverging Connection

This situation of causal chain is called diverging connection and its conditional independence can be represented as:

$$P\ (Ultrasonic\ Velocity \mid Damage, Amplitude\ Value) = P(Ultrasonic\ Velocity \mid Damage)$$

The *Ultrasonic Velocity* is *d*-separated from the *Amplitude Value* given the *Damage*. If there is no evidence or information about damage, then learning that the ultrasonic velocity is high will increase the probability of ultrasonic velocity to be high. Evidence about *Ultrasonic Velocity* can pass to *Amplitude Value* unless the state of *Damage* is known. In this case the parent variable can has more than two children. It can be concluded that evidence my be transmitted through a diverging connection unless the state of the parent is known.

3.2.2.3 Converging Connections

The situation of the causal chain shown in Figure 3.8 is known as converging connection. The description of this connection needs a little of care, which is the inverse of the previously mentioned connections. If nothing is known about *Damage* except what may be inferred from knowledge of its parents *Age* and *ToolDrop*, then the parents are independent, that means evidence on one of them has no influence on the certainty of the others. However, if we are sure that there is damage or no damage in the material or one know the state of any one of the descendants of the *Damage*, then information on the age of the material may tell us something about whether a tool drooped in the material or not. For example, if we know that there is a damage in the material, then knowing that the material is not old will increase the probability of a tool drooped in the material.

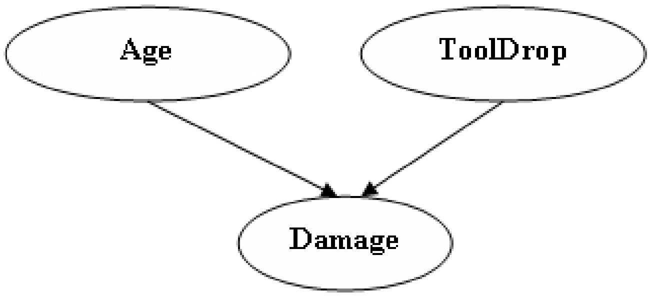


Figure 3.8: Converging Connection

The result is that evidence may only be transmitted through a converging connection from a parent of the variable in the connection to another parent, if either the variable in the connection or one of its descendants has received evidence.

3.2.2.4 *d*-Separation

The three preceding cases (serial, diverging, and converging connections) show how *BNs* represent conditional independence; how these independence affect belief change during updating, and cover all the ways in which evidence may be transmitted through a variable in causal chain. Following the above mentioned rules it is possible to decide whether any pair of variables in a causal network are independent given the evidence entered in the network. These rules can be formulated by the following definition [59].

Definition 1 (d-separation:)**.** *Two distinct variables* X *and* Y *in a causal network are d-separated if, for all pathes between* X *and* Y*, there is an intermediate variable* R *(distinct from* X *and* Y*) such that either :*

- *The connection is serial or diverging and the sate of* R *is known or*
- *The connection is converging, and neither the state of R nor any of the states of R's descendants is known.*

If *X* and *Y* are not *d*-separated, they are called *d*-connected. A similar definition of *d*-separation was given by Jensen [59].

There is no any requirement in the *BNs* definition that the links represent causal impact and the definition does not refer to causality. It is required that the *d*-separation properties implied by the structure hold.

3.2.3 The Conditional Probabilities

A way of structuring a situation for reasoning under uncertainty is to construct a graph representing causal relations between events. The basic concept in the *BN* treatment of certainties in causal networks is conditional probabilities.

If A is assumed to be a variable with n states $a_1, a_2, \ldots, a_n$, then $P(A)$ denotes a probability distribution over these states:

$$P(A) = (x_1, x_2, \ldots, x_n); \qquad x_i \geq 0; \qquad \sum_{i=1}^{n} x_i = 1 \tag{3.1}$$

where x_i is the probability of A being in state a_i. This can be written as $P\ (A = a_i) = x_i$ or $P\ (a_i) = x_i$, e.g. P(*Age* = *new*) = 0.8.

If the variable B has m states b_1, b_2, ..., b_n, the conditional probability statement can be shown as follows:

"The probability of the event a given the event b is x."

which can be written as $P(a \mid b) = x$. The probability $P(A \mid B)$ implies an $n \times m$ table including the probabilities $P(a_i \mid b_j)$.

The fundamental rule for probability calculus is:

$$P(a|b)P(b) = P(a, b), \tag{3.2}$$

where $P(a, b)$ is the probability of the joint event a and b. From this, it can be said that $P(a \mid b)\ P(b) = P(b \mid a)\ P(a)$, and this yields the well known *Bayes*' rule:

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)} \tag{3.3}$$

In Figure 3.5, the variable *Damage* has two parents and the variables *ToolDrop* and *Age* have no any parents. The joint probability distributions for the variables are shown as P(*Damage* | *Age*, *ToolDrop*) P(*ToolDrop*), and P(*Age*). These probabilities are determined by an expert or automatically extracted from a data set. Since the variables *ToolDorp* and *Age* have no parents, their prior probabilities can be specified as follows:

- $P(\textit{ToolDrop} = \textit{yes}) = 0.8$ and $P(\textit{ToolDrop} = \textit{no}) = 0.2$

- $P(\textit{Age} = \textit{new}) = 0.2$, $P(\textit{Age} = \textit{medium}) = 0.7$, and $P(\textit{Age} = \textit{old}) = 0.1$.

The variable *Damage* has *3* states and 2 parents, and each parent with *2* states. The conditional probability distribution of this variable can be shown as on Table 3.1. The table has *12* probability values ($3 \times 2 \times 2$).

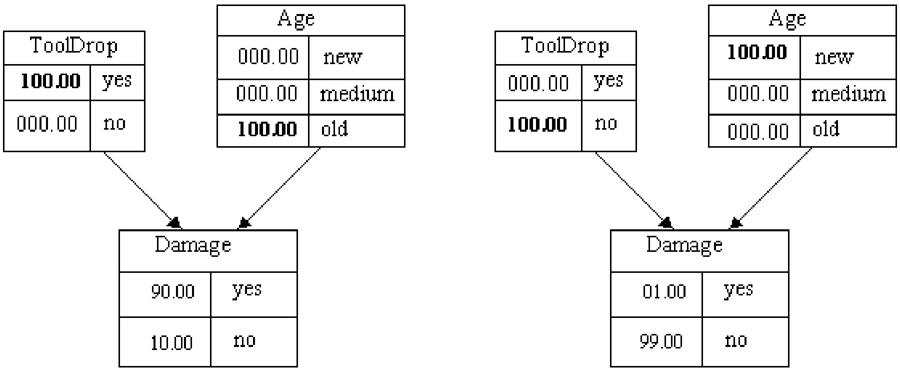
Table 3.1: *CPT* for $P(\textit{Damage} \mid \textit{Age}, \textit{ToolDrop})$.

ToolDrop	*yes*			*no*		
Age	***new***	***medium***	***old***	***new***	***medium***	***old***
yes	*0.2*	*0.4*	*0.9*	*0.01*	*0.5*	*0.4*
no	*0.8*	*0.6*	*0.1*	*0.99*	*0.5*	*0.6*

BNs give full representation of probability distributions over their variables. They can be conditioned on any subset of their variables, supporting any direction of reasoning. That means any variables may be query variables and any may be evidence variables. Whenever new information have arrived new beliefs can be calculated. We have shown that $P(\textit{ToolDrop} = \textit{yes}) = 0.8$ and $P(\textit{Age} = \textit{old}) = 0.1$. Suppose it has been discovered that a tool is dropped on the material and the material is very old, then $P(\textit{ToolDrop} = \textit{yes}) = 1.0$ and $P(\textit{Age} = \textit{old}) = 1.0$. These probabilities are shown in Figure 3.12 as percentages (*100.00* and *00.00*) on bold fonts. This kind of probabilities is sometimes referred as evidence or instantiation. In *BNs*, when new evidence arrive to some variables, the beliefs on other variables may be changed. This can be shown by carefully studying Figure 3.12. This process of conditioning on some variables, when observing the value of other variables is known as probability propagation, inference, or belief updating.

3.2.4 Bayesian Networks as Classifiers

In many problem domains where a *BN* network is applicable and desirable, the label(s) for a subset of the variables (class variables) may be inferred given the states of the rest variables. *BN* classifiers model the conditional distribution of the class variables given



(a) The evidence that the tool is dropped (*ToolDrop* = *yes*) and the material is old (*Age* = *old*), increased our belief on the damage to 90%

(b) The evidence that the tool is not dropped (*ToolDrop* = *no*) and the material is new (*Age* = *new*), decreased our belief on the damage to 1%

Figure 3.9: Changing of believes on BNs, when some evidence are entered

the other variables and predict the class with the highest conditional probability. *BN* classifiers have been applied successfully in many application areas including computational molecular biology , computer vision, relational databases, text processing, audio processing and sensor fusion. Its simplest form, the naive Naïve bayes, has received significant amount of attention. *BNs* were not considered as classifiers until the discovery of the Naïve bayes classifiers. Since then the use of *BNs* for classification problems has received considerable attention [64].

Given a data set X with n variables (x_0, x_1, . . . x_n), the classification task can be defined as the prediction of the class label $x_c \in X$ given a set of variables (attributes) $X_a = X \backslash X_c$. A classifier $c : x \rightarrow x_c$ is a function that maps an instance of x to a value of x_c. A *BN* classifier represents the joint distribution $P(X_c, X_a)$ and converts it to conditional distribution $P(X_c|X_a)$.

The classifiers learn from training data the conditional probability of each attribute X_a given the class label X_c. Classification is done by applying Bayes rule to compute the probability of X_c given the particular instances of X_a and then predicting the class with the highest posterior probability. This computation is rendered feasible by making a strong independence assumptions: all the attributes X_i are conditionally independent given the value of the class X_c.

Based on the theory of learning *BNs*, classifiers can be induced from data. Among *BN* classifiers are Naïve bayes and tree augmented Naïve bayes.

3.2.4.1 Naïve bayes Classifier

Naïve bayes has a simple structure and a strong independence assumption that all variables in the network are independent given the classification variable (as shown in Figure 3.10). All connections in the Naïve bayes go from the parent to the children, no any connection is allowed between any pair of children. It is very easy to build a Naïve bayes network structure, because it does not require a structure learning algorithm. The performance of Naïve bayes is somewhat surprising given that this is clearly an unrealistic assumption. If one considered a classifier for assessing the risk in loan applications, it would be erroneous to ignore the correlations between age, education level, and income [65].

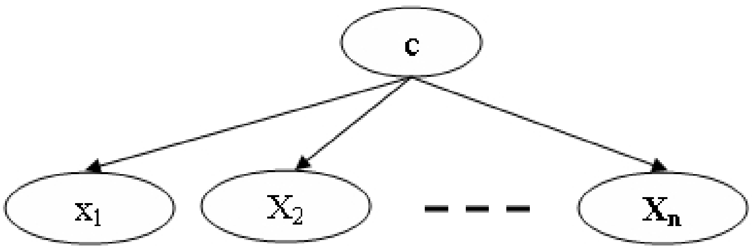


Figure 3.10: A simple Naïve bayes structure.

The Naïve bayes classifier learns from training data the conditional probability of each variable X_i given the class label C. The classification is then done by applying Bayes rule to calculate the probability of C given the particular instance of $X_1, X_2, \ldots, X_n$, and then predicting the class with the highest posterior probability [65], giving:

$$P(C_i|X) = \frac{P(C_i)P(X|C_i)}{P(X)} = \frac{P(C_i)\prod_{j=1}^{N} P(x_j|C_i)}{\sum_{K}^{k=1} P(C_k)\prod_{j=1}^{N} P(x_i|C_k)} \tag{3.4}$$

where K is the number of classes, J is the number of variables, and $P(x_j|C_k)$ is the conditional probability for the observed value of variable j given the class C_k. The product of conditional probabilities comes from the assumption that variables are independent given the class, which greatly simplifies the computation of the class scores and eases the induction process. After calculating $P(C_i|X)$ for each class, the algorithm assigns the instance to the class with the highest overall score or probability [66].

Although the above formulation of Naïve bayes is the traditional one we can express the score for each class in another form that is more tractable for analytical purpose. The basic idea is that, if we are concerned only with predictive accuracy, we can invoke any monotonic transformation that does not affect the ordering on class scores. One transformation involves removing the denominator, which is the same for each class, and another involves taking the logarithm for the numerator. Together, these produced a new score [66]:

$$S_C = \log P(C) + \sum_{variables} \log P(x_i|C) \tag{3.5}$$

In fact, this form is often used in practice, since it is efficient to calculate and reduces round-off errors due to small fractions [67]. The new score S_C is no longer a probability, but is quite sufficient to predict the most probable class.

The discussion so far has derived the independent feature model, that is, the Naïve bayes probability model. The Naïve bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori (*MAP*) decision rule. The corresponding classifier is the function classify defined as follows:

$$classify(X_1, X_2, \ldots, X_n) = argmax_c p(C = c) \prod_{i=1}^{n} p(X_i = f_i | C = c) \tag{3.6}$$

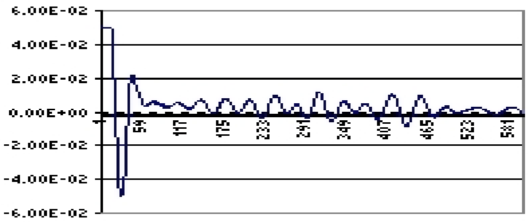
The Naïve bayes classifier has several properties that make it surprisingly useful in practice, despite the fact that the far-reaching independence assumptions are often violated. Like all probabilistic classifiers under the *MAP* decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class; class probabilities do not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying naive probability model. Other reasons for the observed success of the Naïve bayes classifier are discussed in the literature cited below.

Naïve bayes has two main advantages over other classifiers. First, it is easy to construct no learning procedure is required (as mentioned above). Second, the classification process is very efficient since it assumes that all the features are independent of each other. In practical classification problems, it is hardly to come across a situation where the variables are truly conditionally independent of each other. Nevertheless, the Naïve bayes classifier surprisingly outperformed many sophisticated classifiers on data sets where the variables are not strongly correlated.

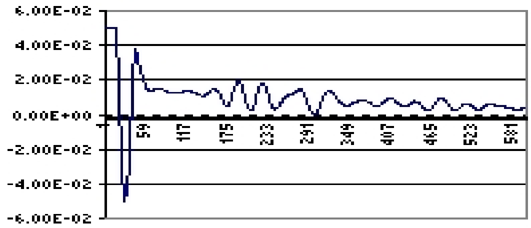
3.2.4.2 Naïve Bayes Classifier for Damage Detection

The amplitudes shown in Figure 3.11 represent voltage amplitudes of Lamb-waves produced and collected by *PZT* sensors and actuators mounted on the surface of quasi-isotropic graphite/epoxy laminates. The first specimen is a control unit (laminate without damage), and the rest of the specimen contain artificial damages.

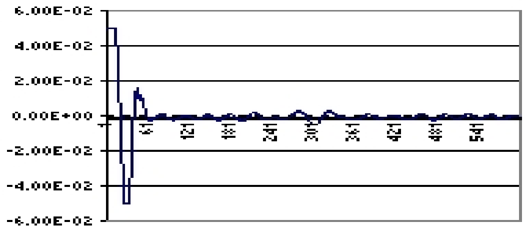
These damages are delamination, crack, and hole. The figure shows that sound waves behave differently when passing through the laminate without and with damage, and



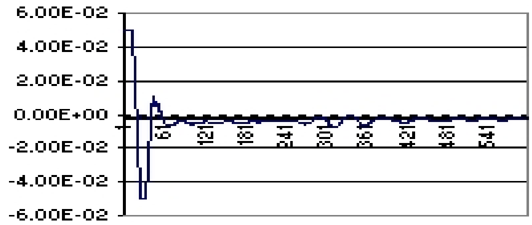
(a) Without damage (cronrol).



(b) Delmination.



(c) Crack.



(d) Hole.

Figure 3.11: Time trace of voltage amplitudes from graphite/epoxy laminates.

every damage produces differing amplitudes. Amplitudes with many cases and different kind of damages can be used to learn the conditional probability tables of variables ($P(Amplitude_i \mid Damage)$) in the network. Ultimately, the model can be used to predict the damages in laminated composite materials with the highest posterior probability. The probabilities of the damages are determined by entering the new evidence obtained from the amplitudes of the new case to the network. The model to predict the damages can be represented by the Naïve-bayes classifier shows in Figure 3.12.

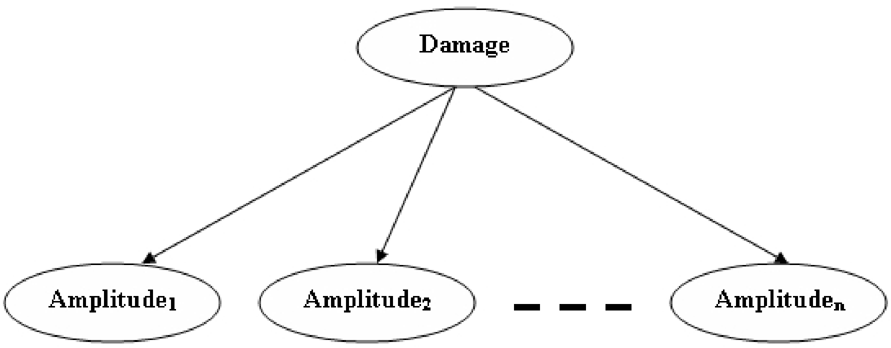


Figure 3.12: A Naïve Bayes classifier for damage detection using wave's amplitudes.

The amplitudes shown in Figure 3.11 were generated by using a constant interval of time (microseconds). For every laminate a set of *600* amplitudes were collected. If all of these amplitudes were used as variables on the damage detection model, the model would be overwhelmed, complicated, and its accuracy might slightly be decreased. Different techniques have been adopted for feature subset selections to decrease the size of the data and increase the accuracy. Some of these techniques extract the peaks of the amplitudes as feature subsets, but it is very difficult to be sure whether these peaks can be representative to the whole wave. The rest of the techniques have different kinds of limitations and disadvantages. So as to overcome some of these limitations and tackle some of these disadvantages, the f-folds feature subset selection algorithm has been developed.

3.2.5 Tree Augmented Naïve bayes Classifier

The strong assumption made by the Naïve bayes classifier that all the variables in the data set are conditionally independent given the value of the class is very likely not to be fulfilled. Nevertheless, the classifier works well in practice even when there are strong dependencies in the data set. Friedman *et al.* [65] introduced the tree augmented Naïve bayes classifier (*TAN*) as a natural extension to the Naïve bayes classifier (Figure 3.13 shows an example of a *TAN* structure). *TAN* models are based on the structure of the Naïve bayes network and a restricted family of *BNs* in which the class variable has no parents and each of the rest of variables has as parents the class variable and at most one other variable which means that there is an edge in the graph from variable X_i to variable X_j. This implies that these two attributes X_i and X_j are not independent given the class label. The influence of X_j on the class probabilities depends also on the value of Xi. Hence, the posterior probability $P(Y \mid X_1, \ldots, X_n)$ takes all the variables into account. Additionally, edges among the variables are allowed in order to capture the correlations among them. The maximum number of edges added to relax the independence assumption between n variables is n - 1 [68, 69].

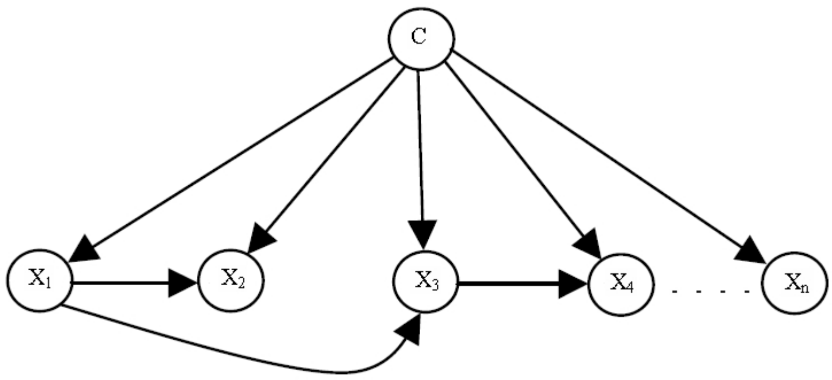


Figure 3.13: Structure of tree augmented Naïve bayes structure.

3.2.6 Selective Unrestricted Bayesian Network Classifier

The selective unrestricted bayesian network classifier (*SUN*) (cited in [69]) can be viewed as a generalization of the *TAN* network. The class variable may have variables as parents [69] (see Figure 3.14).

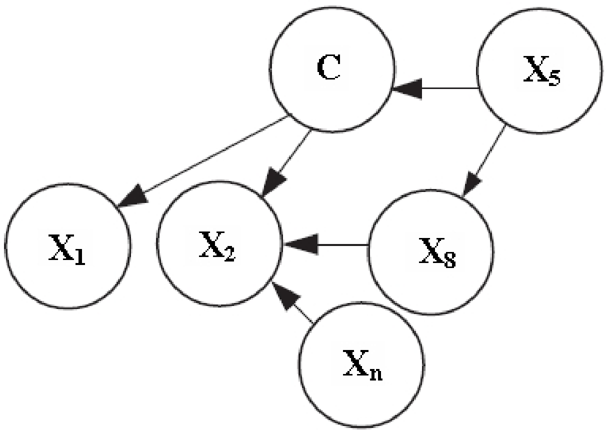


Figure 3.14: General structure of a selective unrestricted bayesian network

The variables need not be connected directly to the class variable as for the tree *TAN* network. After initialization the network may consist of the variables without any links. A search algorithm can be used to add links to the network according to an evaluation criterion. If there is no link between a variable and the classifier network then the variable is not considered during classification. During the determination of the network structure, irrelevant features are not included and the classifier is based on a subset of selected features. This unrestricted network structure maximizes the classification performance by removing irrelevant features and relaxing independence assumptions between correlated features. Since this network is unrestricted, the computational demands for determining the network structure is huge especially if there is a large number of variables available. Additionally, the size of the conditional probability tables of the variables increases exponentially with the number of parents. This might result in a more unreliable probability estimate of the variables, which have a large number of parents [69].

The posterior probability distribution of C given the value of all variables is only sensitive to those variables, which form the Markov blanket of node C [70]. The Markov blanket of the class variable$C$ consists of the direct parents of C, the direct successors (children) of C, and all the direct parents of the direct successors (children) of the class node C. All the features outside the Markov blanket do not have any effect on the classification performance. Introducing this knowledge into the search algorithm reduces the search space and the computational effort for determining the structure of the classifier [69].

Since there are means to represent and manipulate independence assertions of the *BNs*, better *BNs* can be induced by learning unrestricted networks.

3.2.7 Learning Bayesian Networks

Learning *BNs* from data is a rapidly growing field of research that has seen a great deal of attention in recent years [71–73]. Learning *BNs* can be decomposed into two major learning tasks. Learning the graphical structure and the parameters (*CPTs* entries) for the graphical structure. It is trivial to learn the parameters for a given structure that are optimal for a given corpus of complete data. Simply use the empirical conditional frequencies from the data [74]. The learning problem of *BNs* can be stated as follows [75]:

Definition 2 (*BN* learning)**.** *Given a data set, infer the topology for the belief network* *that may have generated the data set together with the corresponding uncertainty distribution.*

This definition is given from the data mining point of view, where the *BN* extracted by the learning algorithm is considered to be a model of the data set. Nevertheless, most of the parts of the learning algorithm can be identified.

In this part of the thesis general descriptions on learning *BN* is given.

3.2.7.1 Structure Learning

Learning the structure of *BN* is known to be *NP*-complete [63, 76]. Generally, methods of learning *BN* structures from data typically divided into two groups. The first approach is a dependence analysis method that poses learning as a constraint satisfaction problem. The algorithms in this approach try to discover the dependencies from the data, usually this is done using a statistical hypothesis test, such as χ_2-test. We then build a network that exhibits the observed dependencies and independencies. The second approach is a searching and scoring based method that poses the learning as an optimization problem. This kind of algorithm defines a "score" that describes the fitness of each possible structure to the observed data. Commonly used scores include Bayesian score [74, 77] and Minimum Description Length (*MDL*) score [78]. Then the structure learning problem becomes an optimization problem: find the structure S_{opt} that maximizes (or minimizes depending on how the score is defined) the score. An important property of some score functions is decomposability. That is, the score function can be decomposed as follows:

$$Score(S, D) = \sum_i Score(X_i, pa(X_i), D(X_i, pa(X_i))) \tag{3.7}$$

Here S denotes the *BN* structure, D denotes the entire data, and $D(X_i, pa(X_i))$ denotes the data involving only X_i and $pa(X_i)$.

One of the most widely used structure learning algorithm is the $K2$ algorithm [74]. It belongs to the second approach. The structure learning problem can be stated as follows: Given the complete training data set D (no missing value) and a node order, find a network structure S that best matches D. Suppose the prior of the parameters (when the structure is fixed) is Dirichlet:

$p(\Theta \mid S) \sim Di(\alpha_{ij1}, \alpha_{ij2}, \ldots, \alpha_{ijr_i})$. Let N_{ijk} be the number of samples in D for which $X_i = k$ and $pa(X_i) = j$. Then the posterior distribution is also Dirichlet:

$p(\Theta \mid S, D) \sim Di(\alpha_{ij1} + N_{ij1}, \alpha_{ij2} + N_{ij2}, \ldots, \alpha_{ijr_i} + N_{ijr_i})$.

We can then write

$$p(D|S) = K2(S, D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \times (\sum_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ij})}) \tag{3.8}$$

and

$$K2(X_i, pa(X_i)) = \sum_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \times (\sum_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ij})}) \tag{3.9}$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ and $\alpha_{ij} = \sum_{k=1}^{r_i}$. $P(D \mid S)$ is called Cooper-Herskovits scoring function. In this thesis, it is referred as the $K2$ score since it is the score function of $K2$ algorithm. Note that the $K2$ score satisfies the decomposability property. Having defined a score, the next step is to identify a network structure with the highest score. Generally, this search problem is NP-hard. So there is a need to use sub-optimal search methods. Most widely used search methods for BN structure learning use the decomposability property. These search methods make a series of arc changes (addition or deletion of one arc at a time). After each arc change, the resulting graph S must be checked whether it is a valid *DAG*. For each arc change, there is a score $Score_b$ for the *DAG* S_b before the change and $Score_a$ for *DAG* S_a after the change. Acceptance of the change depends on the difference between the two scores. If a score satisfies the decomposability property, the search can be done node by node. For each node, only $Score\,(X_i, pa(X_i)_a, D(X_i, pa(X_i)_a))$ needs to be evaluated and not the whole score. This can simplify the computation considerably.

3.2.7.2 Parameter Learning

Once the *BN* structure is specified is built, it constitutes an efficient device to perform probabilistic inference. Nevertheless, the problem of quantification of the network remains as a daunting task, which often requires filling a huge number of *CPTs*. On the other hand, the sensitivity of the networks performance to variations in different probability parameters may be quite different; thus, certain parameters should be specified with a higher precision than the others.

The *CPTs* can be filled by one of three ways [79]:

1. Elicit the probability parameter by consulting domain experts. However, this can be quite expensive and time consuming.

2. Another was is estimate the parameters from the available data. Unfortunately, when the number of probability parameters in *BNs* is huge, a quite large data sets may be required to estimated accurate parameters, especially for probability distributions that describe rare events. In real applications, the databases are often scarce and results in erroneous values for the rare-event probabilities.

3. Utilizing the domain knowledge as well as the data

The classical approach for learning parameters is the likelihood maximization. This leads, with the classical decomposition of the joint probability in a product, to estimate separately each term of the product with the data. This method asymptotically converges toward the true probability, if the proposed structure is exact. The bayesian method rather tries to calculate the most probable parameters given the data, and this is equivalent, with the Bayes theorem, to weight the parameters with an a priori law. The most used a priori is the Dirichlet distribution

Many algorithms have been investigated to learn parameter for *BNs* using samples of real databases (cited in [80]. Expectation-Maximization (EM) algorithms is common algorithm to calculate maximal log likelihood. Lauriten [81] had proved that the algorithm could be applied for parameter learning of *BNS*.

3.3 Summary

This first part of this chapter provides an overview of the practical issues that need to be considered for the implementation and usage of structural health monitoring systems. The issues are discussed in detail with the help of an example, piezoelectric based structural health monitoring system. Issues and solutions for integration of sensors and sensors networks have been presented along with some examples. Two types of piezoelectric based systems are discussed active and passive sensing systems. Finally, the intelligent signal processing has been shown as a key element, which builds the bridge between the sensor signal and the structural integrity interpretation.

The second part of this chapter has shown that *BNs* have incredible power to offer assistance in a wide range of endeavors. They support the use of probabilistic inference to update the probabilities of variables whose state has not been observed given some set of new observations. They automate this process, allowing reasoning to proceed in any direction across the network of variables. *BNs* are powerful tools for knowledge representation and inference under conditions of uncertainty. In doing so, they support complex inference modeling including rational decision making systems, value of information and sensitivity analysis. As such, they are useful for causality analysis and through statistical induction they support a form of automated learning. This learning can involve parametric discovery, network discovery, and causal relationship discovery. Naïve Bayes has a simple structure and a strong independence assumption that all variables in the network are independent of the classification variable

CHAPTER 4

The Methodology

4.1 Introduction

The objective of this chapter is to introduce the methodology of this thesis. The Bayesian networks paly important role in the thesis; their relevant theory is thoroughly investigated and presented in Chapter 3. Aspects of learning the network structures and the Bayesian network classifiers (the Naïve bayes classifier in particular) are presented in detail. The methodology used to introduce the Bayesian networks to the community of damage detection in engineering materials is through testing and evaluating the efficiency of the Naïve bayes classifier for the damage detection. The steps used to do that are collecting data, introducing a feature extraction algorithm (f-FFE: f-folds feature extraction), selecting a suitable tool for the classifier, and implementing and evaluating the extracted features in the classifier. The data sets used are without missing values and they are assumed to be without any outliers. The f-FFE method is implemented on the data set to extract features (form new data set), which believed to minimize the data set and increase the accuracy of the classifier. The new data set is divided into k subsets, and the holdout cross validation method is repeated k times. Each time, one of the k subsets is used as the test set for the Naïve bayes classifier and the other k - 1 subsets are put together to form a training set. Then the average accuracy across all k trials is computed. The f-FFE algorithm needs the number of clusters and folders to be specified (this is elaborated later in this chapter). Therefore, it is important to run the algorithm many times on the new data set so as to specify the optimum number of folders and clusters that maximize the accuracy of the classifier. The necessary processes of the methodology are shown in Figure 4.1.

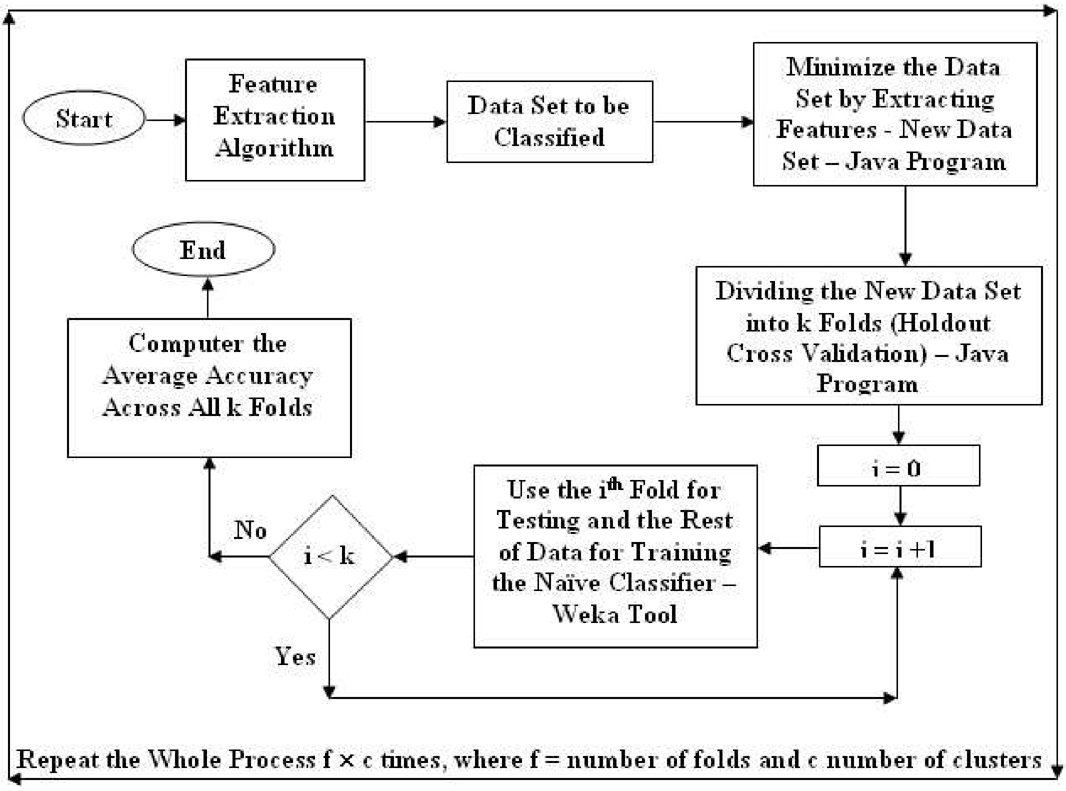


Figure 4.1: The methodology of the damage detection in engineering materials.

4.2 Data Sets

Data collection is a common bottleneck in damage detection. It is very difficult to create standard data and it is very difficult to create artificial data, which simulate the damages. Data can be expensive to create, in terms of all types of resources. For example, the data for damage detection need huge amount of real materials to be used for creating artificial damages; this needs huge amount of money and time. Therefore, the limitation of the time and money let two data sets used in the thesis to be collected from previous works.

The first data set ([7]) created using *25 cm* $\times$ *5 cm* rectangular $[90/\pm45/0]_s$ quasi-isotropic laminates of the AS4/3501-6 graphite/epoxy system. Three Piezoelectric Transducer (*PZT*) patches were mounted on the surface of each specimen. The *PZT* was cut into $2\ cm \times 0.5\ cm$ patches so that the longitudinal wave would be favored over the transverse one, and three patches were used on each specimen to actuate and accurately measure the transmitted and reflected waves. The first channel, which was served as the trigger for all of the channels, was connected to the output channel and actuating *PZT*, two others were connected to the sensing piezoceramic patches to the specimen to serve as a control channel in order to zero out drift. A few shapes of piezoceramic patches were used to produce Lamb waves, and as expected waves propagated parallel to each edge, i.e. longitudinally and transversely for a rectangular patch and circumferentially from a circular piezo. Various types of damages were introduced to the specimens including, holes, fiber fracture, matrix cracking, and delamination. Lamb waves were propagated to the specimens by using *15* and *50 KHz* frequencies.

The second data set is vibration data from a type of ball bearing operating under different fault conditions. The ball bearing is of the type *6204* with a steel cage. The raw measurement data took the form of an acceleration signal recorded on the outer casing for the bearing in five states [91]:

1. New ball bearing (*a*).
2. Outer race completely broken (*b*).
3. Broken cage with one loose element (*c*).
4. Damaged cage, four loose elements (*d*).
5. No evident damage, badly worn ball bearing (*e*).

The rotational frequency was *24.5625 Hz* and a tacho-signal was used for the measurement. The sampling frequency for the time data was *16384 Hz* and the acquisition system was a *Brüuel* and *K jaer* Spectrum analyzer. The points were recorded in *56* instances of *2048* samples, where *11* instances for case *1*, *9* for case *2*, *12* for each case of *3*, *4* and *5*.

The pre-processing was kept to a minimum. Each signal was divided into overlapping 64-point intervals each offset by eight points from its predecessor. Each set was Fourier transformed and the magnitude of each spectral line was recorded. This yielded a sequence of *32*-component vectors for classification [91].

4.3 f-FFE: f-FOLDS FEATURE EXTRACTION ALGORITHM

In machine learning, the problem of supervised classification is concerned with the prediction of class labels of the instances in a data set from a finite set of known class labels. The instances are described by a vector of numeric, nominal features (variables), or a combination of both. In classification techniques, using many features may slightly decrease the classifier accuracy and complicates the computational process of the classification. In addition, as the number of features grows, the number of training instances required will also grow exponentially. Therefore, in many practical situations, it is necessary to reduce the dimensionality of the data by decreasing the number of the features. It has been shown that redundant information can be removed and the classification result will be more reliable in the reduced subspace. Feature reduction methods have been widely adopted to reduce input dimension of data with large input variables before implementing the classification techniques. These methods can be divided into two types, feature selection and extraction. They have wide range of applications in different types of classifications, such as text classification, *DNA* micro-array data analysis, image recognition, image retrieval, and so on [82, 84, 89].

Consider a data set with a set D_n of n features ($D_n = \{d_1, d_2, \ldots, d_n\}$) :

1. **The feature extraction** : Can be considered as a mapping of D_n into a new set of features $V_m = \{v_1, v_2, \ldots, v_m\}$ ($D_n \rightarrow V_m$), where m is the number of features in V_m, and $m \ll n$. This approach needs an assumption that all the features are

relevant for the classification but their number is very big.

2. **The feature selection** : Can be considered as the selection and identification of an optimal subset of features V_m from D_m, where m is the number of features in V_m, $V_m \subseteq D_m$, and $m \ll n$. This approach needs an assumption that there are irrelevant and redundant features available in D_m to be removed.

In feature selection, the integrity of the original features is preserved. Although feature selection keeps the original physical meaning of selected features, it costs a great degree of time complexity for an exhaustive comparison if a large number of features is to be selected. The feature selection is an *NP*-hard problem [84]. In contrast, feature extraction is considered as a process to generate a new and smaller feature set by combining the original features. Strictly speaking, feature selection is less flexible than feature extraction in that feature selection is, in fact, a special case of feature extraction (with a coefficient of one for each selected feature and a coefficient of zero for any of the other features). This explains why an optimal feature set obtained by feature selection may or may not yield a good classification result [88]. The feature selection is problematic, when there is a large number of potential features for classification and the best method to use depends on the circumstances. Evaluation of the methods is generally comparative and based on simulations [83]. This thesis focuses on the feature extraction for the sake of flexibility and effectiveness.

Consider a data set X with C class labels and N instances, where $X = \{ (x_n, w_n)\}$, $n = 1, 2, \ldots, N$, $x_n \in \Re^N$, and $w_n \in \omega$, $\omega = \{1, 2, \ldots, C\}$. The feature extractor in the feature extraction process requires a mapping $f : X \to X'$, where X is the original feature space and $X' \subseteq \Re^M$ ($M \ll N$) is the reduced feature space. Subsequently, classification requires mapping (linear or nonlinear) the instances of the original data set using the reduced feature space to the class labels $C : X' \to \omega$ [85].

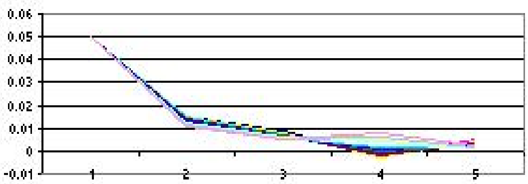
In the case of *BNs*, using many variables increases the size of the network and makes it very difficult to fill in the *CPTs*. Consequently, complicates the model and decreases its accuracy. Therefore, the feature extraction is an important process for classification techniques for damage detection in *EMs* in general and in the case of *BNs* in particular.

In this section, the f-Folds Feature Extraction algorithm is introduced. It is a novel feature extraction method intended to map the amplitudes of waves of an *NDT* technique (e.g. ultrasonic) used to detect damages in *EMs* to new values. These new values are the mean, maximum, and minimum values of the amplitudes of every instance after dividing each one into folders, grouping the amplitudes, and implementing a clustering algorithm on these groups for every instance separately. In the rest of this section first a preliminary study about the algorithm is discussed then the algorithm is introduced.

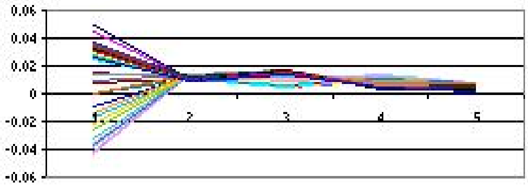
4.4 Preliminary Study

The data used for the preliminary study as a base for the f-Folds Feature Extraction algorithm were collected from Kessler et al [7]. The data were *25 cm* $\times$ *5 cm* rectangular $[90/\pm 45/0]_s$ quasi-isotropic laminates of the AS4/3501-6 graphite/epoxy system. Three Piezoelectric Transducer (*PZT*) patches were mounted on the surface of each specimen. The *PZT* was cut into *2 cm* $\times$ *0.5 cm* patches so that the longitudinal wave would be favored over the transverse one, and three patches were used on each specimen to actuate and accurately measure the transmitted and reflected waves. The first channel, which was served as the trigger for all of the channels, was connected to the output channel and actuating *PZT*, two others were connected to the sensing piezoceramic patches to the specimen to serve as a control channel in order to zero out drift. A few shapes of piezoceramic patches were used to produce Lamb waves, and as expected waves propagated parallel to each edge, i.e. longitudinally and transversely for a rectangular patch and circumferentially from a circular piezo. Various types of damages were introduced to the specimens including, holes, fiber fracture, matrix cracking, and delamination. Lamb waves were propagated to the specimens by using *15* and *50 KHz* frequencies.

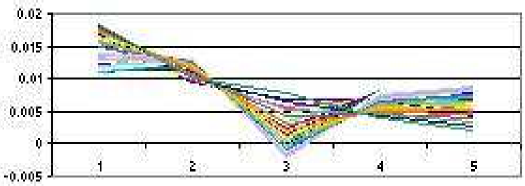
The amplitudes of these data were collected by using a constant interval of time (microseconds). A different data set might be acquired, if the interval value had been changed. If it had been assumed that the interval was increased *10* times more than the original one, then the original amplitudes would be divided into *60* folders (*10* amplitudes in each fold). In this case *10* different data sets would be formed each with *60* amplitudes. The amplitudes included in each set depend on the first amplitude selected from the first folder. If the first amplitude in folder number one was the first to be included, then the first amplitudes in the other folders would be included to the data set. If the second one was the first one to be included, then the second ones in all other folders would be included in the data set, etc.



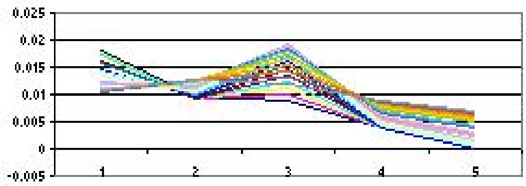
(a)



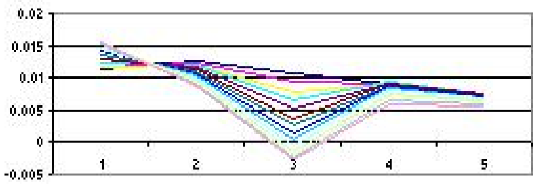
(b)



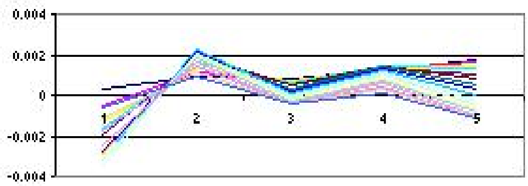
(c)



(d)



(e)



(f)

Figure 4.2: The similarity of wave shapes of the data sets using *10* folders.

Every instance of the data sets was divided into a different number of f folders ($3 \leq f \leq 10$) and subsets of data were created from these folders for every data set as mentioned above. When the graphs of the subsets of every data set were plotted, there were many subsets that have shown similar shape of graphs as depicted in Figure 4.2 (many figures have been shown in Appendix A). This gives an indication that the subsets of the data set can be divided into clusters, from which the mean, maximum, and minimum values of these clusters can be used as representatives to these clusters for damage detection. This has been used as a base to formalize the k-Folds Feature Extraction Algorithm shown bellow (Algorithm 1 and Figure 4.5).

4.5 f-Folds Feature Extraction Algorithm

Algorithm 1 k-Folds Feature Extraction Algorithm

Input:

$Amps = amp_1, amp_2, \ldots, amp_n$ (Amplitudes to be clustered).
k (number of clusters), f (number of folders).

Outputs:

$Means = \{m(c_1), m(c_2), \ldots, m(c_k)\}$
$Maxs = \{max(c_1), max(c_2), \ldots, max(c_k)\}$
$Mins = \{min(c_1), min(c_2), \ldots, min(c_k)\}$

procedure Clustering

1. Divide $Amps$ into f folders ($fold(1)$, $fold(2)$, $\ldots$, $fold(f)$), where $|fold(1)| = |fold(2)| = \ldots = |fold(f)|$, $fold(i) = \{fold(j)_1, fold(j)_2, \ldots fold(j)_m\}$, $m = n / f$, and $1 \leq j \leq f$.
2. Create a new data set $NewAmp = nAmp(1), nAmp(2), \ldots, nAmp(m)$, where $\forall A = fold(k)_i, A \in nAmp(i), 1 \leq i \leq m, and \quad 1 \leq k \leq f$ (the number of elements in each fold is m = n / f).
3. Implement a clustering algorithm (e.g. k-means) on NewAmp, to return k clusters.
4. Return the mean, maximum, and minumum values of the clusters.

The input to the f-Folds Feature Extraction Algorithm (Algorithm 1 is a set of n amplitudes ($Amps = amp_1, amp_2, \ldots, amp_n$). In step *1* the algorithm divides the data set into f folders. All folders contain the same number of m amplitudes, where $m = n / f$. In step *2* the algorithm forms a new set of data containing m records by assigning the amplitudes with the same index in all folders to the data set as one record (e.g. the first amplitudes in all folders form the first record and so on). This creates the data set $NewAmp$ ($nAmp(1)$, $nAmp(2)$, $\ldots$, $nAmp(m)$). The number of variables in each record is f (the number of folders). In step *3* the algorithm implements a clustering algorithm (e.g. k-means algorithm [86, 87]) on $NewAmp$ to divide their instances into

k clusters. Since each record has f variables, the algorithm returns f mean values, f maximum values, and f minimum values of each cluster. These values will be considered as representatives to the clusters and when combined together they can replace the original data set. For example, if there are *100* instances in the cluster, only *3* instances are used (means, maximums, and minimums). The total number of the variables (t) in each damage type will be reduced to $3 \times f \times k$, when the means, maximums, and minimums of the clusters are considered. Finally, it will be reduced to $f \times k$, if only the means are considered. The values of f and k must be determined by the user such that $t \ll n$, which believed to decrease the number of variables to a minimum that highly increase the accuracy of the model and simplify it.

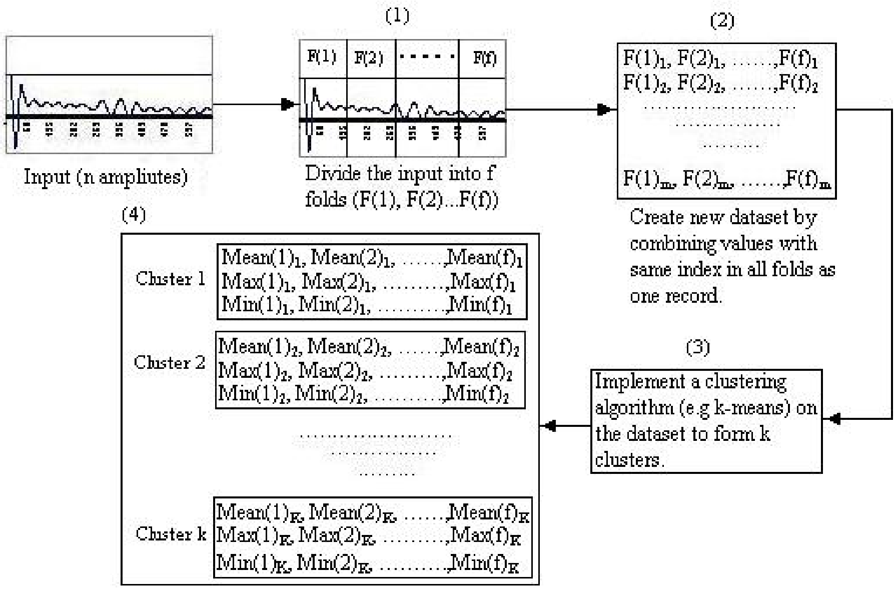


Figure 4.3: The k-Folds Feature Extraction algorithm.

4.6 Summary

Feature reduction methods have been widely adopted to reduce input dimension of data with large input variables before implementing the classification techniques. These methods can be divided into two types, feature selection and extraction. In feature selection, the integrity of the original features is preserved. In contrast, feature extraction is considered as a process to generate a new and smaller feature set by combining the original features. Strictly speaking, feature selection is less flexible than feature extraction in that feature selection is, in fact, a special case of feature extraction. This thesis focuses on the feature extraction for the sake of flexibility and effectiveness.

In this section, the f-Folds Feature Extraction algorithm was introduced. The f-Folds Feature Extraction algorithm is a novel method intended to map the amplitudes of waves of an *NDT* technique (e.g. ultrasonic) used to detect damages in *EMs* to new values. These new values are the mean, maximum, and minimum values of the amplitudes of every instance after dividing each one into folders, grouping the amplitudes, and implementing a clustering algorithm on these groups for every instance separately.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

The intention of this chapter is to demonstrate the potential of Naïve bayes classifier for damage detection in engineering materials, implement the f-FFE algorithm, and test and show its results using different number of folders and clusters. The f-FFE algorithm extracted features from a set of vibration data from a type of ball-bearing data operating under different fault conditions. The Naïve bayes classifier used in this study was implemented in the open-source machine learning package Weka. This thesis assumes that appropriate data preprocessing has been performed on the data set used. Since k-means algorithm can handle continues features, there is no need to discretize any of the features. It was important for the sake of the algorithm to divide the data into folders, extract subsets of data from these folders, implement a clustering algorithm on these sets, and calculate the mean, maximum, minimum values of the clusters. Since these potentials were not offered by the Weka tool, two programs were written in Java to fill this gap.

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a data set or called from a separately written Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes [90]).

In this section, first the Ball-bearing data set is discussed. Second, the implementation of the f-FFE algorithm using the two Java programs are illusterated. Third, the implementation of the Naïve bayes classifier on the features extracted by the f-FFE algorithm is discussed. Lastly, an evaluation has been done.

5.2 Implementation

The first Java program (shown in Appendix B1) written implements step *1* and *2* of the f-FFE algorithm. Every instance in the data set was divided by the program into different number of folders (*4*, *6*, *8*, *10*, and *12*). As mentioned before, the number of the samples in every instance was *2048*. Therefore, when the number of the folders was *4* the samples of every instance were divided into *512* subsets, when the number of folders was *8*, the number of the subsets was *256*, and so on. The steps of creating these subsets are mentioned in Algorithm 1. The program automatically writes the data into Weka format. The k-means clustering algorithm was implemented separately on every group of subsets of all instances with different number of clusters ranging from two to eight. The subsets for every instance were saved in seven separate files, every file represents the clustering results of the different number of clusters (*2*, *3*, *4*, *5*, *6*, and *7*).

The second Java program (shown in Appendix B2) implements step *3* of the f-FFE algorithm. This program implemented on all files created by the first program. This program creates the mean, maximum, and minimum values of the clusters. For example, if the number of the folders was four and the number of clusters was two, the number of subsets would be *512* and the number of samples in every subset would be four. This can be shown in a table of size 512×4 (rows $\times$ columns). After implementing the k-means algorithm on this table, every row in this table would be assigned a cluster label of *0* or *1*. That means this table would be grouped in to two parts. Every part would be represented by the mean , maximum, and minimum values of the four columns. The total number of the features (samples) of every instance in this case would be reduced to *24* samples ($3 \times 4 \times 2$). Table 5.1 shows the number of the reduced features for all numbers of folders and clusters tested in this thesis.

Table 5.1: The number of reduced features for all features.

Number of Folders	Number of Clusters ⇒						
⇓	2	3	4	5	6	7	8
4	24	36	48	60	72	84	96
6	36	54	72	90	108	126	144
8	48	72	96	120	144	168	172
10	60	90	120	150	180	210	240
12	72	108	144	180	216	252	288

Table 5.2 shows the number of the reduced features when using the combination of mean and maximum features.

Table 5.2: The number of reduced features for mean and maximum

Number of Folders	Number of Clusters ⇒						
⇓	2	3	4	5	6	7	8
4	16	24	32	40	48	56	86
6	24	36	48	60	72	84	96
8	32	48	64	80	96	102	128
10	40	60	80	100	120	140	160
12	48	72	96	120	144	168	192

The the Nïave bayes classifier was implemented on the files created by the second program mentioned above. Labeling the instances in these files involve applying a previously learned classifier to an unlabeled data set to predict instance labels. Testing takes a labeled data set, temporarily removes class labels, applies the classifier, and then analysis the quality of the classification algorithm by comparing the actual and the predicted labels. The *k*-fold cross validation was used, which partitions the data set into *k* folds, and performs *k* training and testing iterations. On each iteration, one fold is used as a test set, and the rest of the data is used as a training set. The classifier is learned on the training set and then validated on the test data. The number of folds used in this thesis was *10*. The classifier was implemented on every file separately and the results were records. The *10* results from the folds for each file were combined to produce single estimation.

5.3 Testing

It has been assumed that the maximum features represent the peaks of the amplitudes. It was decided to test the efficiency of the mean, maximum, and minimum features of the clusters together, separately, and in combination. The classifier was firstly tested using the mean, maximum, and minimum features, secondly using the mean and maximum features, thirdly using the mean features only, and lastly using the maximum features only. The percentages of the correctly classified instances together with the confusion matrices for the classification result for each case were recorded.

Table 5.3 shows the confusion matrix of the classification result for *4* folders and *2* clusters, when using the mean, maximum, and minimum features. In the table, the number of correctly classified instances is *51* out of *56* (91.0714%).

Table 5.3: Confusion matrix for all features (*4* folders and *2* clusters).

Correctly classified 51 (91.0714%)					
a	b	c	d	e	← classified as
11	0	0	0	0	a
0	6	3	0	0	b
0	0	11	1	0	c
0	0	0	11	1	d
0	0	0	0	12	e

Table 5.4 shows the confusion matrix of the classification result for *4* folders and *2* clusters, when using the mean and maximum features. In the table, the number of correctly classified instances is *51* (91.0714%). The results of this table is similar to the previous one. The removal of the minimum features does not change the classification accuracy.

Table 5.5 shows the confusion matrix of the classification result for *4* folders and *2* clusters, when using the mean features only. In the table, the number of correctly

Table 5.4: Confusion matrix for combination features (*4* folders and *2* clusters).

Correctly classified *51* (91.0714%)					
a	b	c	d	e	← classified as
11	0	0	0	0	a
0	6	3	0	0	b
0	0	11	1	0	c
0	0	0	11	1	d
0	0	0	0	12	e

classified instances is *47* (83.9286%). The classification accuracy is still higher but decreased.

Table 5.5: Confusion matrix for mean features only(*4* folders and *2* clusters).

Correctly classified *47* (83.9286%)					
a	b	c	d	e	← classified as
10	1	0	0	0	a
3	4	2	0	0	b
0	1	10	1	0	c
0	0	0	11	1	d
0	0	0	0	12	e

Table 5.6 shows the confusion matrix of the classification result for *4* folders and *2* clusters, when using the maximum features only. In the table, the number of correctly classified instances is *27* (48.2143%). The classification accuracy is highly decreased when using only the maximum features.

Table 5.6: Confusion matrix for maximum features only(*4* folders and *2* clusters).

Correctly classified *27* (48.2143%)					
a	b	c	d	e	← classified as
11	0	0	0	0	a
0	6	3	0	0	b
0	0	8	4	0	c
0	0	10	2	0	d
0	0	10	2	0	e

The percentages of correctly classified instances and the confusion matrices concerning the rest of the other different number of folders and clusters used in this thesis are shown in Appendix C.

5.4 Summary

The Naïve bayes classifier together with the proposed f-FFE algorithm have shown great potential, when tested on a set of vibration data from a type of ball bearing operating under different fault conditions, new ball bearing, outer race completely broken, broken cage with one loose element, damaged cage, four loose elements, no evident damage, and badly worn ball bearing. The features extracted by the algorithm contain, the mean, maximum, and minimum features of the clusters. The implementation has been done using different number of folders and clusters.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Introduction

This section evaluates the results of Nïave bayes classifier and the f-FFE algorithm, when implemented using different number of folders and clusters. The first purpose of the evaluation is to compare the classification accuracies based on folders for all number of clusters considered in this thesis and to specify the number of clusters that give the best results. The second purpose of the evaluation is to compare the classification accuracies based on clusters for all number of folders considered in this thesis and to specify the number of folders that give the best classification accuracies. The third purpose is to determine the features that give the best results.

6.2 Comparison of Results Based on Folders

Figure 6.1 shows the classification accuracies of the classifier for all number of clusters when the number of folders was *4*.

It is quite obvious from the graph that the classification accuracies are very low in the case of maximum features. The average accuracy approximately is 50%. The classification accuracies for the classifier show similar trend when using the all features, the combination of the mean and maximum features, and the mean features. In these cases, all accuracies are above 80%. The best results are obtained when the combi-

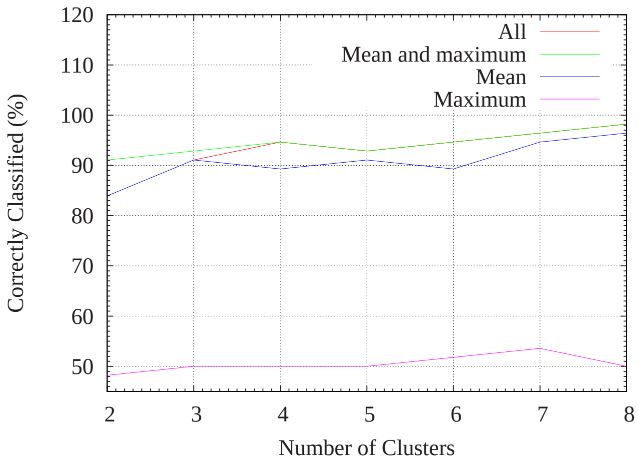


Figure 6.1: Comparison of classification accuracies (*4* folders)

nation of the mean and maximum features are used and all accuracies are more than 90%. In general, when the mean features are used, their accuracies are less than the accuracies of all features and the combination of the mean and maximum features. In the case of the all features and the combination of mean and maximum features, the results are similar to each other in *4* clusters. The increase of the accuracies when the number of clusters are greater than *4* (approximately 95%) are not high and can be ignored. Therefore, in this figure, the 4 clusters can be considered the optimum number of clusters to give the best results.

Figure 6.2 shows the classification accuracies of the classifier for all number of clusters when the number of folders is *6*.

As can be seen from Figure 6.1, the classification accuracies for the maximum features are very low when compared to the other cases but higher than the same case for the previous graph. In some cases they have reached more than 75%. The classification accuracies in the rest of cases are greater than 90% and they show results, which can be considered similar to each other. The accuracies of the all features are similar to that of the combination of mean and maximum features when the number of clusters is greater than *2* and less than *8*, but they are a bit less when the number of clusters is *2* and increases when the number of clusters is *8*. The figure gives an indication that the

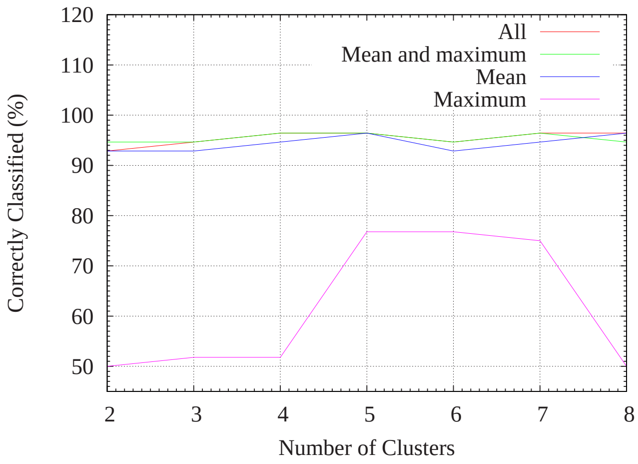


Figure 6.2: Comparison of classification accuracies (*6* folders).

decrease will be very high when the number of clusters gets bigger than *8* clusters. It can be concluded from this figure that the best results can obtained when the number of clusters is *4*.

Figure 6.3 shows the accuracies for all number of clusters when the number of folders is *8*.

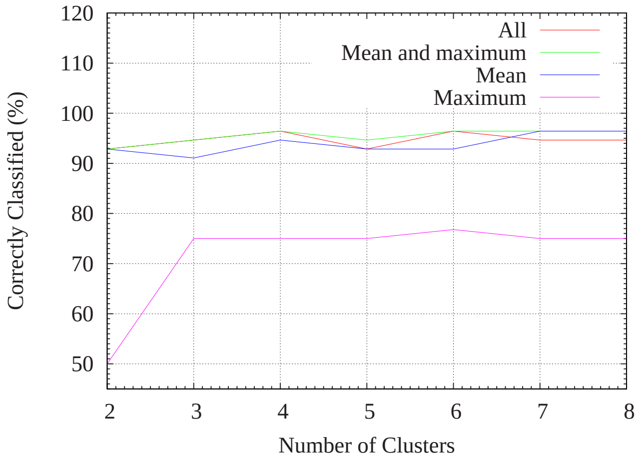


Figure 6.3: Comparison of classification accuracies (*8* folders)

In this figure, the accuracies are still low for the maximum features, but generally have shown an improvement when compared to the previous cases. All of the accuracies are greater than 75% except in the *2* clusters. The accuracies of the other cases do not show big variations among each other. Nevertheless, in all cases, the accuracies of the combination of mean and maximum features are the best. All accuracies in these cases are greater then 90%. The highest accuracies are shown when the number of clusters is equal to *4* and greater than *5*. Therefore, the best results can be considered for the *4* clusters.

Figure 6.4 shows the classification accuracies for all number of clusters when the number of folders is *10*.

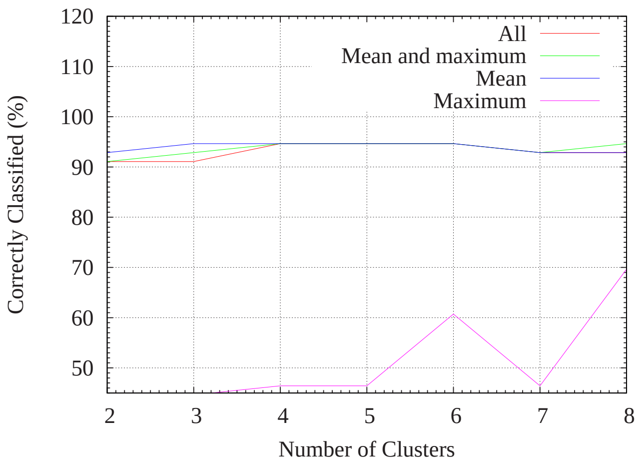


Figure 6.4: Comparison of classification accuracies (*10* folders)

In this figure the accuracies for the maximum features are very low when compared to the previous cases. The accuracies of the other cases are greater than 90% and they are equal when the number of clusters are greater than *4* and less than *8*. Generally, the combination of mean and maximum features show better accuracies than the others when the number of the clusters is *8*. In this graph it can also be accepted that the *4* clusters give the optimum accuracies.

Figure 6.5 shows the classification accuracies for all number of clusters when the number of folders is *12*.

As in the previous figures the accuracies are still low for the maximum features, even though, the accuracy is greater than 80% in *7* clusters. The accuracies in the other cases are greater than 90% and in general the accuracies of the combination of mean and maximum features are showing better results than the others. The accuracies of

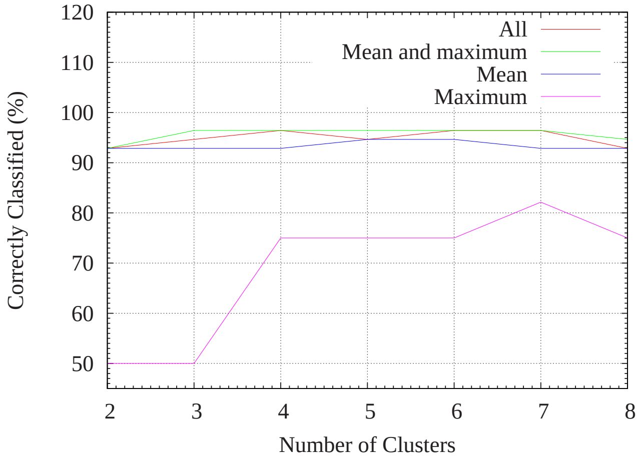


Figure 6.5: Comparison of classification accuracies (*12* folders)

all features are better than the mean features, except when the number of clusters is *5*, where they are equal.

The experimental results of comparing the classification accuracies based on folders illustrate that the best accuracies can be obtained when using the combination of mean and maximum features with *4* clusters.

6.3 Comparison of Results Based on Clusters

Figure 6.6 shows the classification accuracies for all number of folders when the number of clusters was *2*.

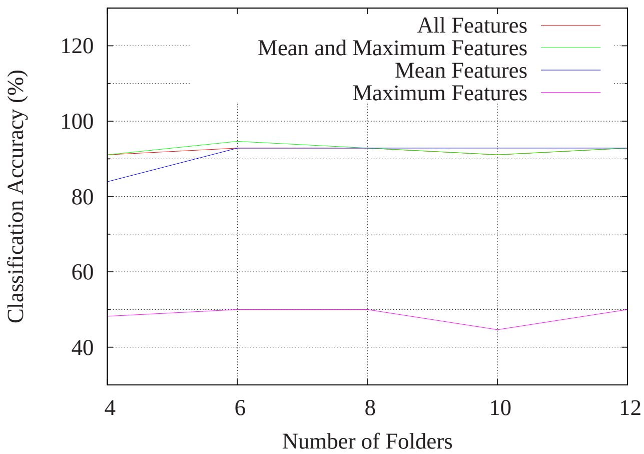


Figure 6.6: Comparison of classification accuracies (*2* clusters)

In this figures, the accuracies of the maximum features are very low when compared to the other cases. This case will repeat for the figures shown below. In the other cases the accuracies some how are similar to each other. The best accuracy is obtained when the number of folders is *6*. It is very obvious from the figure that when the number of folders is greater than *6*, the accuracy does not show any considerable change. Therefore, the *6* folders can be considered as the most convenient number of folders in case of 2 clusters.

Figure 6.7 shows the classification accuracies for all number of folders when the number of clusters was *3*.

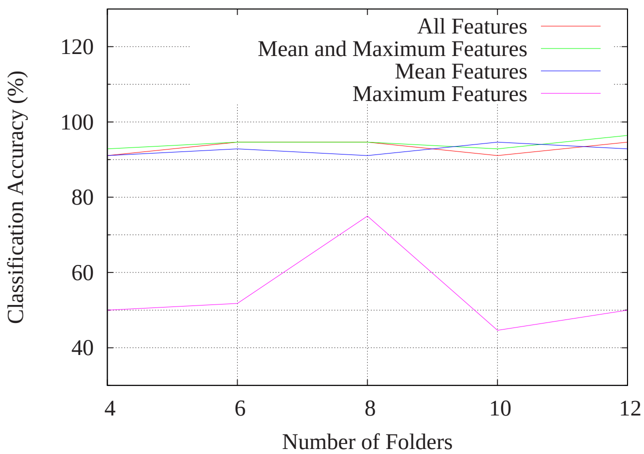


Figure 6.7: Comparison of classification accuracies (*3* clusters).

In this figures, the combination of mean and maximum features have shown the highest accuracy when the number of folders is *10*, but in the rest of the folder numbers, the combination of the mean and maximum features have shown higher accuracy than the mean features. It has also shown greater accuracy than all features except when the number of folders is *6* an *8*, which they show equal accuracies. Nevertheless, it is acceptable by scrutinizing the figure to consider the best accuracies where given when the number of folders is *6*.

Figure 6.8 shows the classification accuracies for all number of folders when the number of clusters was *4*.

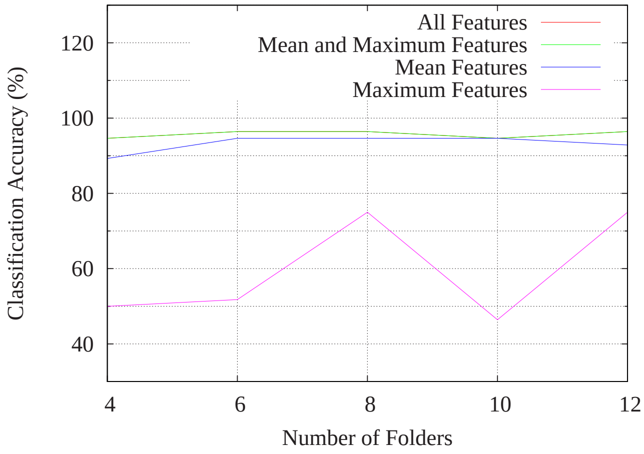


Figure 6.8: Comparison of classification accuracies (*4* clusters)

In this figure, the combination of the mean and maximum features have shown similarity of accuracies to all features in all number of clusters. Most of these features have shown accuracies greater than 95%. The mean features have shown less accuracies than the all and the combination of mean and maximum features, just at the 10 folders they have shown similar accuracy. It possible to consider the number of the folders that gives the best accuracies is *6*, because the change of the accuracies when the number of folders is greater than 6 is considerably not tremendous.

Figure 6.9 shows the classification accuracies for all number of folders when the number of clusters was *5*.

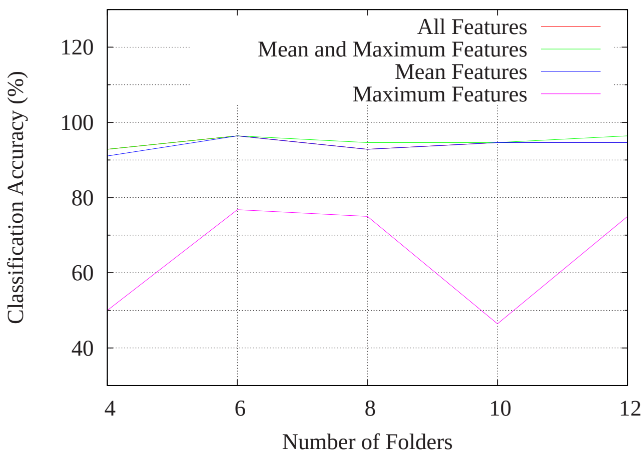


Figure 6.9: Comparison of classification accuracies (*5* clusters)

Figure 6.10 shows the classification accuracies for all number of folders when the number of clusters was *6*.

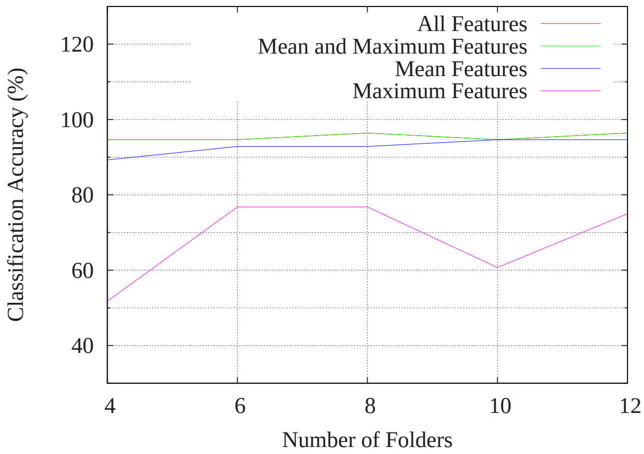


Figure 6.10: Comparison of classification accuracies (*6* clusters)

In this figure, most of the accuracies are greater than 60% and the highest accuracies are recorded when the number of folders is *6* and *8*, which are greater than 75%. Still they are lower than other features. The accuracies of the all features and combination of mean and maximum features have shown similar trend. The highest accuracies are recorded when the number of the folders is *8*, but still not so great than other accuracies. The accuracies of the mean features are lower than the all and combination of mean and maximum features, except in *10* folders, where the accuracies are similar.

Figure 6.11 shows the classification accuracies for all number of folders when the number of clusters was *7*.

In this figure, the accuracies of the maximum features show diversity, the accuracy of *12* features is greater than 80%, greater than 50% for *4* folders, and less than 50% for *10* folders. Nevertheless, they still low when compared to other features. The accuracies of the other features are quite similar to each other and all of them are greater than 90%. The highest accuracies are recorded for the mean and combination of mean and maximum features when the number of features is *8*, but they still not that so much greater than that when the number of folders is *6*.

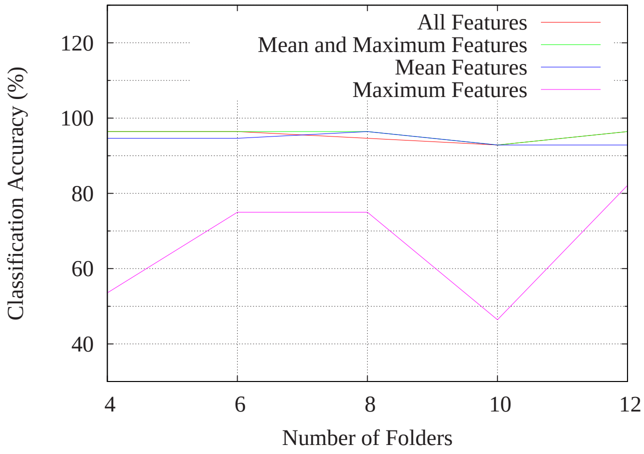


Figure 6.11: Comparison of classification accuracies (*7* clusters)

Figure 6.12 shows the classification accuracies for all number of folders when the number of clusters was *8*.

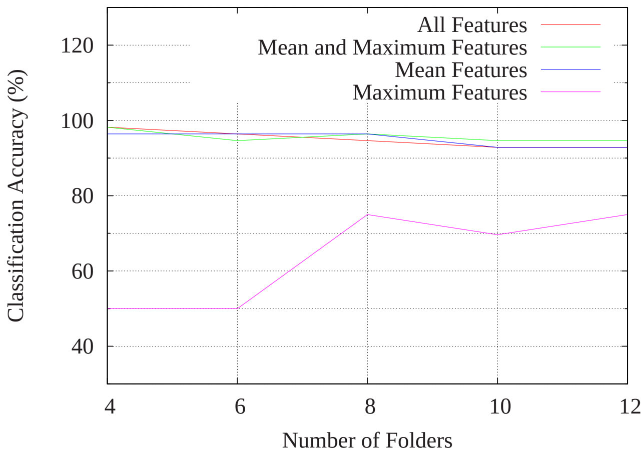


Figure 6.12: Comparison of classification accuracies (*8* clusters)

In this figure, all accuracies of the maximum features are between 50% and 85%, but still less than the accuracies of other features. The accuracies of other features are very similar to each other and the differences can be ignored. The accuracies of these features are very close to 100% when the number of the folders is *4* and the accuracies in general are slightly decreasing when the number of the folders are increasing.

The experimental results of comparing the classification accuracies based on clusters show that the best accuracies can be obtained when using the combination of mean and maximum features with *6* folders. It can also be concluded that using greater number of folders requires fewer number of clusters to obtain higher accuracy.

6.4 Summary

The experimental results conducted in this section have shown the efficiency of the Naïve bayes classifier and the e-FFE algorithm for damage detection in *EMs*. In many cases, the accuracies of the classifier using the features extracted by the algorithm have reached values greater than 95%. The best classification accuracies were obtained when the combination of mean and maximum features with *6* folders and *4* clusters were used. In this case the number of the features will be decreased for each instance from *2048* to *48* (this can be checked from Table 5.2). It has also shown that using the maximum features alone for classification will highly decrease the accuracy of the classifier but the mean features alone have shown very good accuracies when compared to the maximum feature, but a less better than the combination of mean and maximum.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

The main objective of this thesis is to introduce the Bayesian networks for the community of damage detection in engineering materials. The methodology used to satisfy that is to introduce a feature extraction algorithm (the f-FFE: f-folds feature extraction algorithm) so as to extract features from input data, which maximize the classification accuracy, and implement the Naïve bayes classifier on these features. Two data sets are used as part of the methodology. The first set represents voltage amplitudes of Lamb-waves produced and collected from quasi-isotropic laminates. The second set is a vibration data from a type of ball bearing operating under different five fault conditions. The ball bearing is of the type *6204* with a steel cage. The raw measurement data took the form of an acceleration signal recorded on the outer casing for the bearing in five states. The main contributions of this thesis can be summarized as follows:

- The Bayesian networks have been widely implemented as classifiers in many disciplines such as gene regulatory networks, medical diagnostic systems, text analysis, and image processing. However, to the author's knowledge, they have not been explored in damage detection in engineering materials. The thesis has shown that the Bayesian networks in general and the Naïve bayes classifiers in particular are competitive classifiers to other machine learning classifiers that are implemented for the damage detection (e.g. Neural networks, and genetic algorithms).

- To the knowledge of the author, most of the feature reduction algorithms used in the damage detection are based on feature selection algorithms. However, the feature extraction has not been implemented for damage detection in engineering materials. The proposed f-folds feature extraction algorithm has shown good efficiency in damage detection, specially when compared to the techniques that are based on the peaks of the waves of amplitudes.

- The proposed algorithm has been tested with different number of folders and clusters. The best results obtained when the number of clusters is four, the number of folders is six, and the combination of mean and maximum values has been used. The highest accuracy of the classifier obtained exceeds 95%. It has been shown that the maximum values only (the peaks) have shown the worst classification results in comparison to other cases and the mean values have show good results, which can be compared to the combination of the maximum and mean values. The number of the features extracted is highly decreased to 48, while the original data contain 2048 amplitudes.

- Most of the techniques used for the feature reduction in the damage detection are borrowed and implemented from other techniques, the thesis has shown that some techniques can be developed based on the damage detection, which can be efficient when compared to other techniques.

The present thesis demonstrates the efficiency and applicability of Bayesian networks as classifiers for damage detection in engineering materials. It is concluded that Bayesian networks as classifiers have indeed offer many advantages for the damage detection in engineering materials.

7.2 Recommendations for Future Work

Thus the methodology has many advantages for the community of damage detection in engineering materials. However, from an overall point of view, the whole processes and works presented in this thesis can be extended by purely theoretical development, by use of the models in large real-world applications, or by implementing advanced specialized algorithms applied in existing software tools.

The limitation of data resources have forced the research to be limited only to two sets of data. The complexity of the damage detection may grow beyond tractability, and a need for more, huge, and different types of data to generalize the results of the Bayesian network classifiers is apparent.

Nevertheless, the performance of the Naïve bayes classifiers has been proven surprisingly to be successful and competitive to many classifiers in many disciplines, it is important to compare it in the damage detection to the performance of other types of Bayesian network classifiers, e.g. tree augmented Naïve bayes classifiers, selective unrestricted Bayesian network classifiers, or any other Bayesian classifiers discovered by a structure learning algorithm.

The proposed f-FFE algorithm needs the number of folders and clusters pre-specified, it can be extended by adding techniques to automatically determined the optimum number of clusters and folders that maximize the classification accuracies.

REFERENCES

[1] R. Shotblasting, *Airbus A380: A new dimension*, Aerospace Edition (unpublished), 2006.

[2] J. Schaffer, A. Saxena, S. Antolovich, T. Sanders, and S. Warner, *The Science and Design of Engineering Materials, Second Edition*, WCB McGraw-Hill, New York, 1999.

[3] Z. Gurdal, R. Haftka, and P. Hajela, *Design and Optimization of Laminated Composite Materials*, John Wiley & Sons, Inc., 1999.

[4] C. Son, I. Kim, and J. Paik, *Experimental study on structural behavior and vibrational characteristics of sandwich plate with aluminum honeycomb*, International Journal of Offshore and Polar Engineering, 8:1053-5381, 1998.

[5] S. Hong, J. Pan, T. Tyan, and P. Prasa, *Quasi-static crush behavior of aluminum honeycomb specimens under non-proportional compression-dominant combined loads*, International Journal of Plasticity, 22:1062-1088, 2006.

[6] P. Mauk, *Roller bearing*, unpublished, 2005.

[7] S. Kessler, S. Spearing, M. Atalla, E. Cesnika, and C. Soutisb, *Structural health monitoring in composite materials using frequency response methods*, Composites Part B, 33:87-95, 2002.

[8] R. Miller and P. McIntire, *Nondestructive Testing Handbook*, American Society for Nondestructive Testing, 1987.

[9] P. Dempsey, J. Certo, and W. Morales, *Current Status of Hybrid Bearing Damage Detection*, Annual Meeting and Exhibition, the Society of Tribologists and Lubrication Engineers, Toronto, Canada, 2004.

[10] A. Takeuchi, *Detection of Operational Abnormality of Ball Bearing with ultrasonic technique*, Engineering Materials, 252-257, 2004.

[11] S. Hall, *The effective management and use of structural health data*, Proceedings of the Second International Workshop on Structural Health Monitoring, Stanford, CA, 265-275, 1999.

[12] D. Chakraborty, *Artificial neural network based delamination prediction in lam-*

inated composites, Materials & Design, 26:1-7, 2005.

[13] Z. Su and L. Ye, *Lamb wave-based quantitative identification of delamination in CF/EP composite structures using artificial neural algorithm*, Composite Structures, 66:627-637, 2004.

[14] F. Chang, *Structural Health Monitoring*, A Summary Report Proceedings of the Second International Workshop on Structural Health Monitoring, Stanford, CA, 8-10, 1999.

[15] V. Giurgiutiu, B. Jingjing, and W. Zhao, *Active Sensor Wave Propagation Health Monitoring of Bean and Plate Structures*, Proceedings of the SPIE International Symposium on Smart Structures and Material, Newport Beach, CA, 2001.

[16] S. Kessler, S. Spearing, M. Atalla, C. Cesnik, and C. Soutis, *Damage detection in composite materials using frequency response methods*, Composites Part B: Engineering, 33:87-95, 2002.

[17] P. M. Mujumdar and S. Suryanarayan, *Flexural vibrations of beams with delaminations*, Sound and Vibration, 125:441-461, 1988.

[18] P. Cawly and R. Adams, *A vibration technique for non-destructive testing of fibre composite structures*, Composite Material, 13:161-175, 1979.

[19] S. Kulkarni and D. Fredrick, *Frequency as a parameter in delamination problem - a preliminary investigation*, Composite Materials, 1971.

[20] M. Duggan and O. Ochoa, *Natural frequency behaviour of damaged composite materials*, Sound and Vibration, 158:545-551, 1992.

[21] P. Kaminsk, *The approximate location of damage through the analysis of natural frequencies with artificial networks*, Process Mechanical Engineering, 209:117-123, 1995.

[22] A. Islam and K. Craig, *Damage detection in composite structures using piezoelectric materials*, Smart Material Structures, 3:318-328, 1994.

[23] A. Reddy, L. Rehfield, and R. Haag, *Influence of prescribed delaminations on stiffness-controlled behaviour of composite laminates effects of defects in composite materials*, ASTM STP 836 Am. Soc. for Testing Mater., 1984.

[24] R. Dua, S. Watkins, D. Wunsch, K. Chandrashekhara, and F. Akhavan, *Detection and classification of impact induced damage in composite plates using neural networks*, Proceedings of the IJCNN, Washington D. C., 681-686, 2001.

[25] R. Jones and J. Sirkis, *Detection of impact location and magnitude for isotropic plates using neural networks*, Intelligent Material Systems and Structures, 7:90-99, 1997.

[26] R. Ceravolo, A. De Stefano, and D. Sabia, *Hierarchical use of neural techniques in structural damage recognition*, Smart Material Structures, 4:270-280, 1995.

[27] C. Li and A. Ray, *Neural network representation of fatigue damage dynamics*, Smart Material Structures, 4:126-133, 1995.

[28] F. Akhavan, S. Watkins, and K. Chandrashekhara, *Predictions of impact contact forces of composite plates using fiber optic sensors and neural networks*, Mechanics of Composite Materials Structure, 7:195-205, 2000.

[29] S. Watkins, W. Sanders, A. Farhad, and K. Chandrashekhara, *Modal analysis using fiber optic sensors and neural networks for prediction of composite beam delamination*, Smart Material Structures, 11:489-495, 2002.

[30] D. Crispin and F. Gerard, *Detecting impact damage in a composite material with an optical fiber vibration sensor system*, Smart Material Structures, 7:543-549, 1998.

[31] J. Lew, *Optimal controller design for structural damage detection*, Sound and Vibration, 281:799-813, 2005.

[32] M. Sahin and R. Shenoi, *Quantification and localisation of damage in beam-like structures by using artificial neural networks with experimental validation*, Engineering Structures, 25:1785-1802, 2003.

[33] Y. Yan, H. Hao, and L. Yam, *Vibration-based construction and extraction of structural damage feature index*, Solids and Structures, 41:6661-6676, 2004.

[34] F. Akhavan, S. Watkins, and K. Chandrashekhara, *Measurement and analysis of impact-induced strain using extrinsic fabry-perot fiber optic sensors*, Smart Material Structures, 7:745-751, 1998.

[35] D. Seo and J. Lee, *Effect of embedded optical fiber sensors on transverse crack of smart composite structures*, Composite Structures, 32:51-58, 1995.

[36] D. Lee, J. Lee, and S. Yun, *The mechanical characteristics of smart composite structures with embedded optical fiber sensors*, Composite Structures, 32:39-50, 1995.

[37] P. Irving and C. Thiagarajan, *Fatigue damage characterization in carbon fiber composite materials using an electric potential technique*, Smart Materials and Structures, 7:456-466, 1998.

[38] J. Abry, S. Bochard, A. Chateauminois, M. Salvia, and G. Giraud, *In-situ detection of damage in CFRP laminates by electric resistance measurements*, Composites Science and Technology, 95:925-935, 1999.

[39] S. Dae-Cheol and L. Jung-Ju, *Damage detection in CFRP laminates using elec-*

trical resistance measurement and neural network, Composite Structures, 47:525-530, 1999.

[40] A. Todoroki and H. Suzuki, *Health monitoring of internal delamination cracks for graphite/epoxy by electric potential method*, Applied Mechanics of Engineering, 5:238-94, 2000.

[41] A. Todoroki and Y. Tanaka, *Delamination identification of cross-ply graphite/epoxy composite beams using electric resistance change method*, Composites Science and Technology, 62:629–639, 2001.

[42] A. Todoroki, Y. Tanaka, and Y. Shimamura, *Measurement of orthotropic electric conductance of CFRP laminates and analysis of the effect on delamination monitoring with electric resistance change method*, Composites Science and Technology, 62:619–628, 2002.

[43] A. Todoroki, Y. Tanaka, and Y. Shimamura, *Delamination monitoring of graphite/epoxy laminated composite plate of electric resistance change method*, Composites Science and Technology, 62:1151-1160, 2002.

[44] A. Todoroki, Y. Tanaka, and Y. Shimamura, *Electrical resistance change method for monitoring delaminations of CFRP laminates: effect of spacing between electrodes*, Composites Science and Technology, 65:37-46, 2005.

[45] H. Lamb, *On Waves in an Elastic Plate*, Proceedings of the Royal Society of London, 93:114-128, 1917.

[46] K. Diamantia, C. Soutisb, and J. Hodgkinson, *Lamb waves for the non-destructive inspection of monolithic and sandwich composite beams*, Composites: Part A, 36:189-195, 2005.

[47] M. Rguiti, S. Grondel, F. El youbi, C. Courtois, M. Lippert, and A. Leriche, *Optimized piezoelectric sensor for a specific application: Detection of Lamb waves*, Sensors and Actuators, 126:362-368, 2006.

[48] C. Paget, S. Grondel, K. Levin, and C. Delebarre, *Damage assessment in composites by Lamb waves and wavelet coefficients*, Smart Materials and Structures, 12:393-402, 2003.

[49] Y. Shenfang, W. Lei, and P. Ge, *Neural network method based on a new damage signature for structural health monitoring*, Thin-Walled Structures, 43:553-563, 2005.

[50] Z. Su and L.Ye, *Fast damage locating approach using digital damage fingerprints extracted from Lamb wave signals*, Smart Materials and Structures, 14:1047-1054, 2005.

[51] DC. Worlton, *Experimental confirmation of Lamb waves at megacycle frequen-*

cies, Applied Physics, 32:967-971, 1961.

[52] T. Lindh, J. Ahola, P. Spatenka, and A. Rautiainen, *Automatic bearing fault classification combining statistical classification and fuzzy logic*, Norpie, Trondheim, 2004.

[53] R. Reddy and R. Ganguli, *Structural damage detection in a helicopter rotor blade using radial basis function neural networks*, Smart Materials and Structures, 12:232-241, 2003.

[54] W. Staszewski, C. Boller, and G. Tomlinson, *Health Monitoring of Aerospaces Structures : Smart Sensor Technologies and Signal Processing*, John Wiley and Sons, Munich, 2003.

[55] S. Bearda, A. Kumar, X. Qing, H. Chana, C. Zhang, and K. Ooi, *Practical issues in real-world implementation of structural health monitoring systems*, Smart Structures and Materials, 5762:196-203, 2005.

[56] S. Kessler and S. Spearing, *Design of a piezoelectric based structural health monitoring system for damage detection in composite materials*, Proceedings of the SPIEs nineth International Symposium on Smart Structures and Materials, San Diego, CA, 2002.

[57] C. Marantidis, C. Van Way, and J.N. Kudva, *Acoustic-emission sensing in an onboard smart structural health monitoring system for military aircraft*, Proceedings of the SPIE Conference on Smart Structures and Integrated Systems, 2191:258-264, 1994.

[58] C. van Koten and A. Gray, *An application of Bayesian network for predicting object-oriented software maintainability*, Information and Software Technology, 48:59-67, 2006.

[59] F. V. Jensen, *Bayesian Networks*, Decision Graphs, Springer-Verlag, New York, 2001.

[60] M. Jordan, *Learning in Graphical Models*, MIT Press, Cambridge, 1999.

[61] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Palo Alto, 1988.

[62] K. Korb and A. Nickolson, *Bayesian Artificial Intelligence*, Ph.D. Dissertation, 2004.

[63] H. Schneiderman, *Learning a restricted Bayesian network for object detection*, IEEE Conference on Computer Vision and Pattern Recognition, 2:639–646, 2004.

[64] Y. Jing, V. Pavlovic, and J. Rehg, *Efficient discriminative learning of Bayesian network classifier via boosted augmented Naive bayes*, Preceedings of the Tewenty

Second International Conference on Machine Learning, Bonn, 2005.

[65] N. Friedman, D. Geiger, and M. Goldszmidt, *Bayesian network classifiers*, Machine Learning, 29:131-163, 1997.

[66] P. Langley and S. Sage, *Tractable average-case analysis of Naive bayesian calssifiers*, Proceedings of the Sixteenth Conference on Machine Learning, Bled, Solvenia: Morgan Kaufman, 220-228, 1999.

[67] P. Langley and S. Sage, *Induction of selective Bayesian classfiers*, Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Seattle, WA: Morgan Kaufmann, 399-406,1994.

[68] P. Langley, W. Iba, and K. Thompson, *An Analysis of Bayesian Classifiers*, In Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI Press and MIT Press, 223-228, 1992.

[69] F. Pernkop, *Bayesian network classifiers versus selective k-NN classifier*, Pattern Recognition, 38:1-10, 2005.

[70] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, Los Alios, CA, 1988.

[71] D. Heckerman, D. Geiger, and D. Chickering, *Learning Bayesian networks: The combination of knowlege and statistical data*, Machine Learning, 20:197-243, 1995.

[72] D. Heckerman, *A tutorial on learning Bayesian networks*, Technical Report MSR-TR-95-06 Microsoft Researc, 1995.

[73] J. Peña, J. Björkegren, and J. Tegneér, *Learning dynamic Bayesian network models via cross-validation*, Pattern Recognition Letters, 26:2295-2308, 2005.

[74] G. Cooper and E. Herskovits, *A Bayesian Method for the induction of probabilistic networks from data*, Machine Learning, 9:309-347, 1992.

[75] J. Alcob, *Incremental Methods for Bayesian Network Structure Learning*, Chapman & Hall, 2004.

[76] D. Chickering, *Learning equivalence classes of Bayesian-network structures*, Machine Learning Research, 2:445-498, 2002.

[77] D. Heckerman, *A tutorial on learning with Bayesian networks*, Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models, Kluwer Academic Publishers, 301-354, 1998.

[78] J. Suzuki, *A construction of Bayesian networks from databases based on an MDL scheme*, Proceedings of the Ninth Conference on Uncertainty in Artificial Intelli-

gence, Morgam Kaufmann, 266-273, 1993.

[79] H. Wang, I. Rish, and S. Ma, *Using Sensitivity Analysis for Selective Parameter Update in Bayesian Network Learning*, Proceedings of 2002 AAAI Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction, Stanford, Palo Alto, 25-27, 2002.

[80] S. Zhang, H. Yu, H. Ding, N. Yang, and X. Wang, *An application of online learning algorithm for bayesian network parameter*, Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2003.

[81] S. Lauritzen, *The EM algorithm for graphical association models with missing data*, Computational Statistics and Data Analysis, 19:191-201, 1995.

[82] N. Abe and M. Kudo, *Non-parametric classifier-independent feature selection*, Pattern Recognition, 2006.

[83] C. Sima, S. Attoor, U. Brag-Netob, J. Lowey, E. Suh, and E. Dougherty, *Impact of error estimation on feature selection*, Pattern Recognition, 38:2472-2482, 2005.

[84] R. Meiri and J. Zahavi, *Using simulated annealing to optimize the feature selection problem in marketing applications*, European Journal of Operational Research, 171:842-858, 2006.

[85] D. Li, W. Pedrycz, and N. Pizzi, *Fuzzy wavelet packet based feature extraction method and its Application to biomedical signal classification*, IEEE Transactions on Biomedical Engineering, 52:1132-1139, 2005.

[86] , H. Guldemir and A. Sengur, *Comparison of clustering algorithms for analog modulation classification*, Expert Systems with Applications, 30:642-649, 2006.

[87] N. Godin, S. Huguet, and R. Gaertner, *Integration of the Kohonens self-organising map and k-means algorithm for the segmentation of the AE data collected during tensile tests on cross-ply composites*, NDT&E International, 38:299-309, 2005.

[88] P. Hsieh, D. Wang, and C. Hsu, *A linear feature extraction for multiclass classification problems based on class mean and covariance discriminant information*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28:223-235, 2006.

[89] E. Gasca, J. Sanchez, and R. Alonso, *Eliminating redundancy and irrelevance using a new MLP-based feature selection method*, Pattern Recognition, 39:313-315, 2006.

[90] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*, Morgan Kaufmann, 2005.

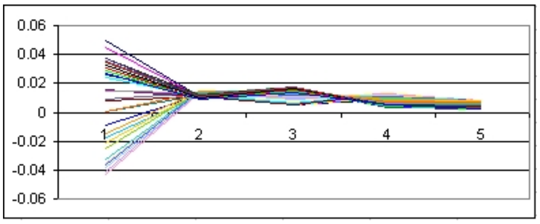
[91] K. Worden and A. Lane, *Damage identification using support vector machines*, Smart Materials and Structures, 10:540-547, 2001.

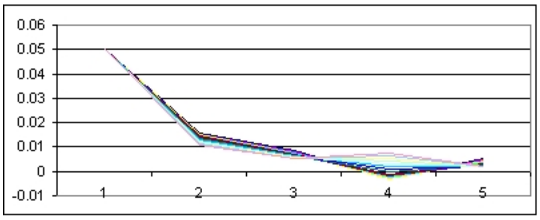
APPENDICES

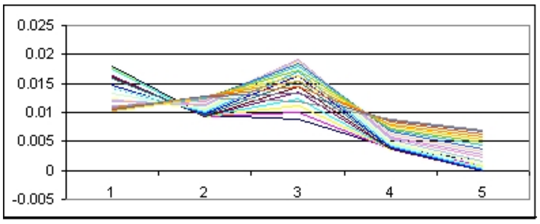
Appendix A

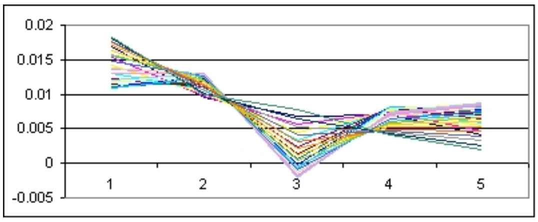
Graphs Used as Preliminary Study for the f-FFE Algorithm

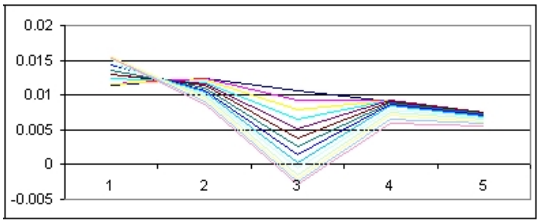
The figures represent the data of the waves collected from the laminates with bend after dividing them into five folders.

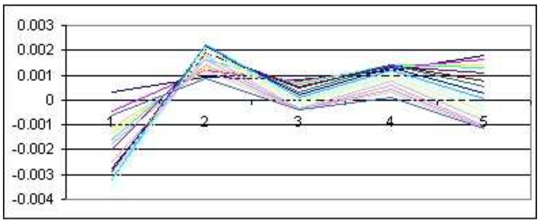


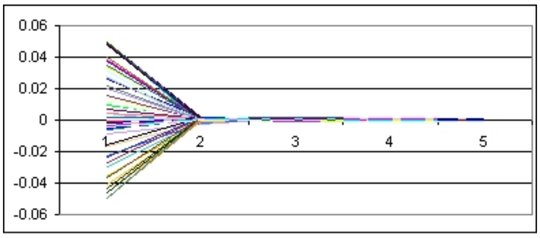


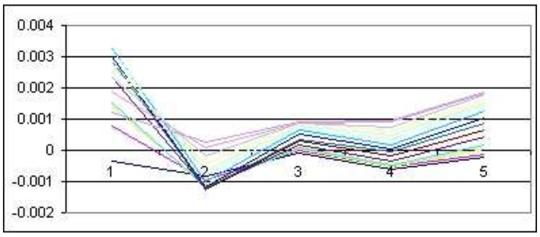


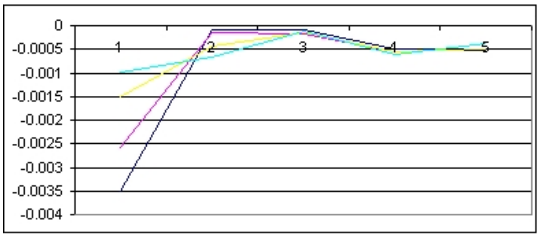


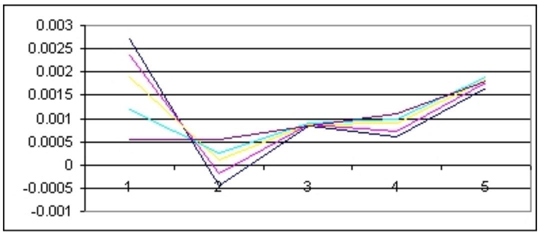






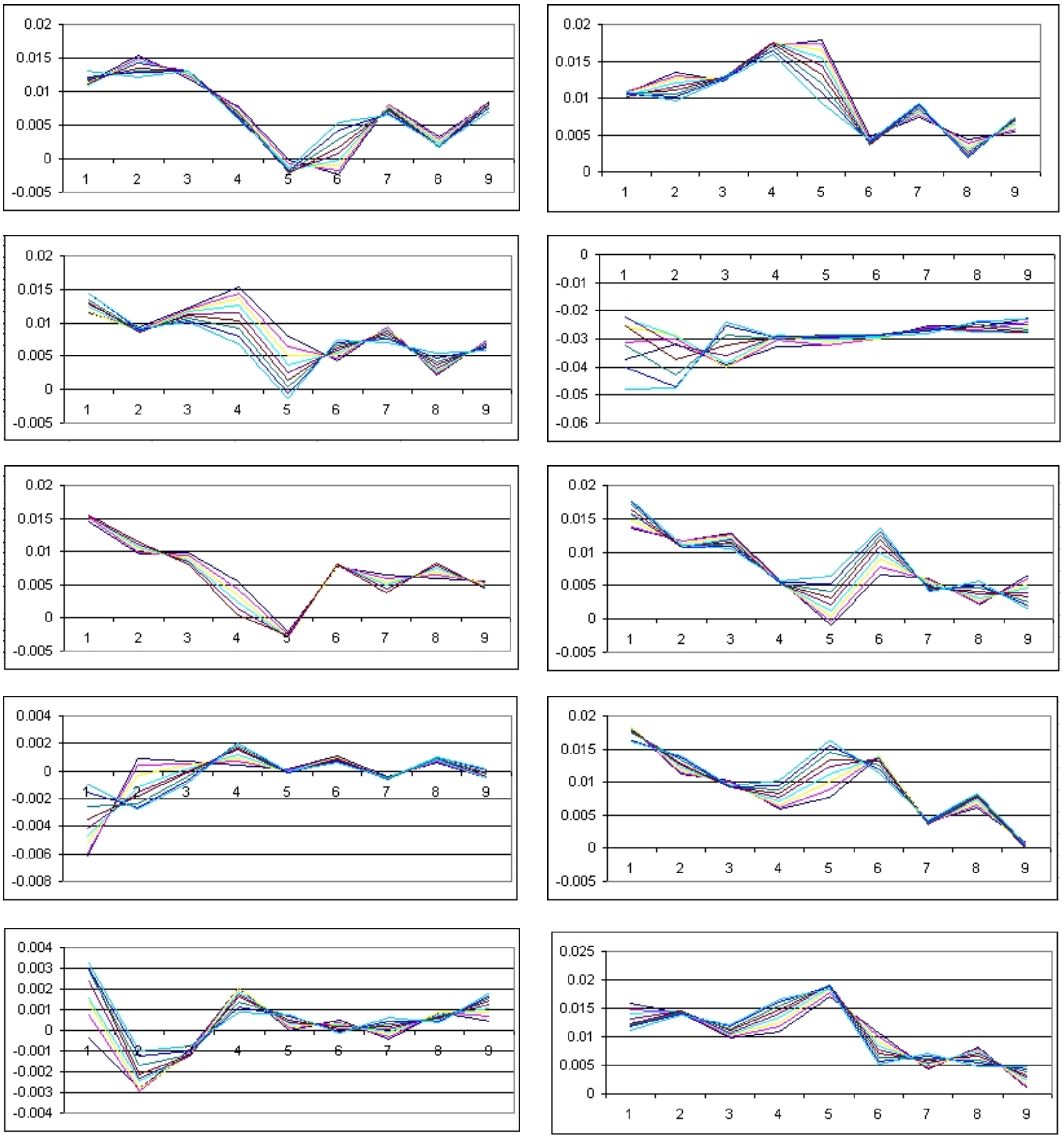






The figures represent the data of the waves collected from the laminates with bend after dividing them into ten folders.

The figures represent the data of the waves collected from the laminates with bend after dividing them into twenty folders.

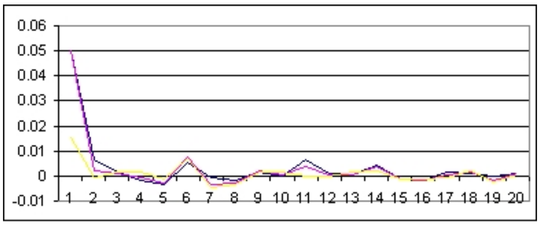


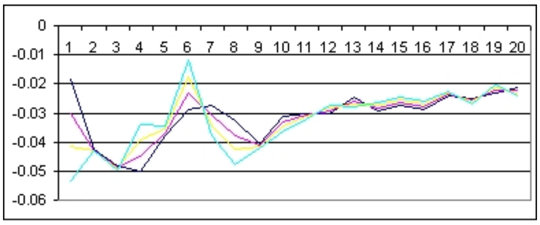
Appendix B1

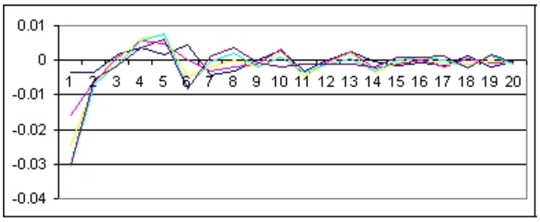
The Java Source Code of f-FFE Algorithm

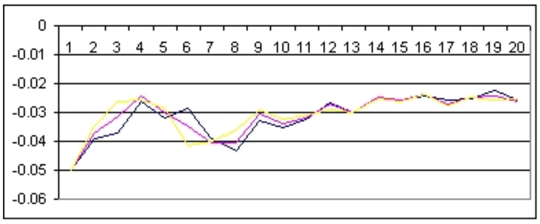
This section contains the source code of the program, which divides the amplitudes of every reading into folders then select the amplitudes with the same index in all folders and assign them as one record of a database.

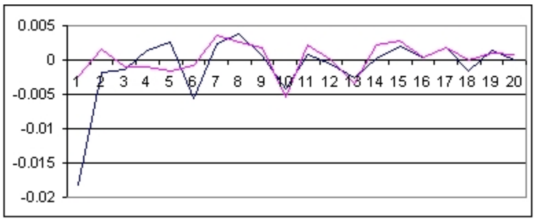
```
import java.io.*;
import java.util.StringTokenizer;
public class
create_Folders{

public static void main(String[] args) throws
IOException {
```

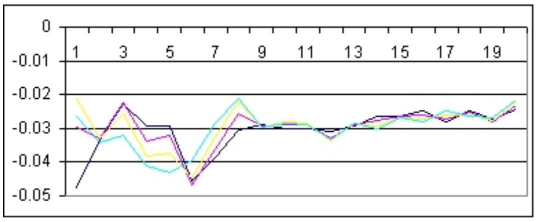


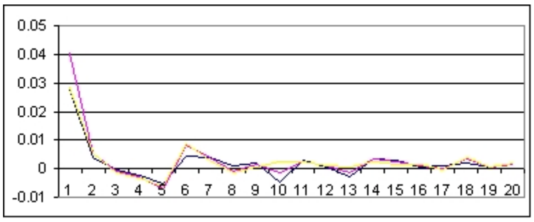


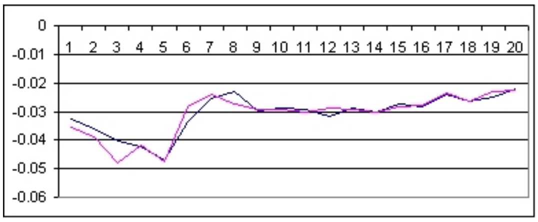


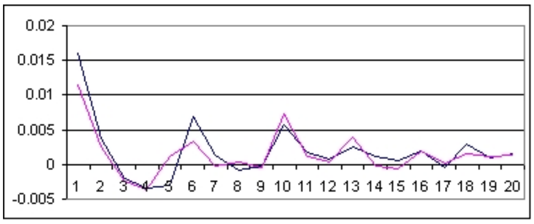













```
//Number of folders
int NumberOfFolds = 8;
//Open all files in the directory.
String FDir = "C:\\PhD_Work\\Data\\Sheffield\\";

//Get the names of all files in the directory.
File dir = new File(FDir + "ball_bearing\\");
String[] children = dir.list();

//Create folders from 4 to 16.
for (int f = 2; f < 7; f++) {
  NumberOfFolds = f * 2;
  convert_File_Folds convert_FFolds = new
  convert_File_Folds(
```

```
     NumberOfFolds );
        convert_FFolds.setFileDirectory (FDir);
        convert_FFolds.setNoFols(NumberOfFolds);

        if (children == null) {
          // Either dir does not exist or is not a
          //directory
          System.out.println("Either dir does not
          exist or is not a directory...");
        } else {
          //Get the names of the file one by one.
          for (int k = 0; k < children.length; k++) {
             //Set the new name of the file.
             convert_FFolds.setFileName(children[k]);
             //Open the output file for writting new data.
             convert_FFolds.setFOPStream();
             //Open the input file to read data.
             convert_FFolds.setBReader();
             //Convert the data by dividing them into folders.
             convert_FFolds.transferData();
           }
         }
        }
    }
}

class convert_File_Folds{
    private BufferedReader input = null;
    private String FileDirectory, FileName, txt_token, line;
    private int NoFolds, NoOfRecords;
    private FileOutputStream fos;
    private DataOutputStream outData;
    private double[][] ConvertedData;
    final int NoAmplitudes = 2048;

    //Contstruct convert_File_Folds with default file name.
    convert_File_Folds () {
        this.FileDirectory =
        "C:\\PhD_Work\\Data\\Sheffield\\ball_bearing\\";
        this.FileName = "l1_1.txt";
        this.NoFolds = 4;
        this.NoOfRecords = NoAmplitudes / NoFolds;
    }

    //Contstruct convert_File_Folds with specified file naem.
    convert_File_Folds (int NFolds) {
        this.NoFolds = NFolds;
        this.NoOfRecords = NoAmplitudes / NoFolds;
}
```

```
public void setFileDirectory (String FD){
   FileDirectory = FD;
}

    public void setFileName (String FN){
        FileName = FN;
        System.out.println("The File Name : " + FileName);
    }

    public void setNoFols (int NF){
        NoFolds = NF;
    }

public void setFOPStream (){
   String txt;
   try {
      txt = FileName.substring(0, FileName.length() - 4);
      fos = new FileOutputStream( FileDirectory +
      "folders_" + NoFolds + "\\" + txt + ".ARFF");
      outData = new DataOutputStream(fos);
   } catch (FileNotFoundException ex) {
      System.out.println("Error in output file");
      ex.printStackTrace();
   } catch (IOException ex){
      System.out.println("Error in output file");
      ex.printStackTrace();
   }
}

public void setBReader (){
   try {
      input = new BufferedReader( new FileReader
      (FileDirectory + "ball_bearing\\" + FileName) );
   } catch (FileNotFoundException ex) {
     System.out.println("Error in input file");
     ex.printStackTrace();
   } catch (IOException ex){
     System.out.println("Error in input file");
     ex.printStackTrace();
   }
}

public void transferData (){
   String tempStr;
   ConvertedData   = new double [NoOfRecords][NoFolds];
   System.out.println("Number of Folders : " + NoFolds);
   System.out.println("Number of Records : " + NoOfRecords);
```

```
    //Organize the data into a matrix.
    try {
       for (int i = 0; i < NoFolds; i++){
          for (int j = 0; j < NoOfRecords; j++){
             line = input.readLine();
             ConvertedData [j][i] = Double.parseDouble(line);
          }
       }

       //Add the Weka headings.
       tempStr = "@relation " + FileName.substring(0,
       FileName.length() - 4) + "\n\n";
       outData.write(tempStr.getBytes());
       for (int i = 1; i <= NoFolds; i++){
          tempStr = "@attribute AMPLITUDE" + i +
          " numeric"  + "\n";
          outData.write(tempStr.getBytes());
       }
       tempStr = "\n";
       outData.write(tempStr.getBytes());
       tempStr = "@data\n";
       outData.write(tempStr.getBytes());

       //Read the matrix
       for (int i = 0; i < NoOfRecords; i++){
          tempStr = Double.toString(ConvertedData [i][0]);
          for (int j = 1; j < NoFolds; j++){
              tempStr = tempStr + "," + Double.toString(
              ConvertedData [i][j]);
          }
          tempStr = tempStr + "\n";
          outData.write(tempStr.getBytes());
       }
    } catch (Exception e) {
       System.out.println("Something went wrong: " +
       e.toString());
    }
  }
}
```

Appendix B2

The Java Source Code of the Second Program of f-FFE Algorithm

This section contains the source code of the program that calculates the mean, maximum, and minimum values of the clusters for each case and combine them to for a new feature.

```
import java.io.*; import java.util.StringTokenizer;
importjava.text.NumberFormat;

public class kmeans_CentroidMaxMinARFF{
  public static void main(String[] args) throws IOException {
  final int Fold_No = 8, Clust_No = 8;

  FileOutputStream fos = new FileOutputStream(
  "C:\\PhD_Work\\Sheffield\\folders_" + Fold_No +
  "\\centroids_maxmin_" + Fold_No + "_" + Clust_No
  + ".ARFF");

  DataOutputStream outData = new DataOutputStream(fos);
  BufferedReader input = null;
  String txt_token = null;
  double[][] clusterAmpTotals   = new double[Clust_No][Fold_No];
  double[][] clusterMax = new double[Clust_No][Fold_No];
  double[][] clusterMin = new double[Clust_No][Fold_No];
  int[]   clusterCount = new int[Clust_No];
  double[]   instanceAmplitudes = new double[Fold_No];
  int indx = 0;
  String txt = null;
  String line = null;
    NumberFormat nf = NumberFormat.getInstance();

  for (int i = 0; i < Clust_No; i++){
     clusterCount[i] = 0;
     for (int j = 0; j < Fold_No; j++){
        clusterAmpTotals[i][j] = 0.0000;
        clusterMax[i][j] = 0.0000;
        clusterMin[i][j] = 0.0000;
     }
  }

  String txt_all;
  double total = 0;
  int c = Fold_No * Clust_No;
  //Write the headings of the ARFF file.
  txt_all = "@relation  MeanMaxMin" + Fold_No + "_"
  + Clust_No +  "\n\n";
  outData.write(txt_all.getBytes());
```

```
for (int i = 1; i <= c; i++){
  txt_all = "@attribute AMP_MEAN" + i + " numeric \n";
  outData.write(txt_all.getBytes());
  txt_all = "@attribute AMP_MAX" + i + " numeric \n";
  outData.write(txt_all.getBytes());
  txt_all = "@attribute AMP_MIN" + i + " numeric \n";
  outData.write(txt_all.getBytes());
}

txt_all = "@attribute DAMAGE {l1, l2, l3, l4, l5}\n";
outData.write(txt_all.getBytes());
txt_all = "\n@data\n";
outData.write(txt_all.getBytes());

//Open all files in the directory.
File dir = new File("C:\\PhD_Work\\Sheffield\\folders_"
+ Fold_No + "\\clusters_" + Clust_No);
String[] children = dir.list();

if (children == null) {
  // Either dir does not exist or is not a directory
  System.out.println("Either dir does not exist or
  is not a directory...");

} else {
  //double str_double = 0;

try {
for (int k = 0; k < children.length; k++) {
 String filename = children[k];
 System.out.println(filename);

 StringBuffer contents = new StringBuffer();
 String aFile = "C:\\PhD_Work\\Sheffield\\folders_"
 + Fold_No + "\\clusters_" + Clust_No + "\\" + filename;
 input = new BufferedReader( new FileReader(aFile) );

 //This readLine let the ARFF headings to be ignored.
 for (int i = 1; i <= Fold_No + 6; i++)
   line = input.readLine();

 while (( line = input.readLine()) != null){
   StringTokenizer st = new StringTokenizer(line, ",");
   txt_token = st.nextToken();
   for (int i = 0; i < Fold_No; i++){
   txt_token = st.nextToken();
   //instanceAmplitudes[i] = Double.parseDouble(txt_token);
   instanceAmplitudes[i] =
```

```
    Double.valueOf(txt_token.trim()).doubleValue();
  }

  txt_token = st.nextToken();
  txt = String.valueOf(txt_token.charAt(7));
  indx = Integer.valueOf(txt).intValue();
  clusterCount[indx]++;
  for (int i = 0; i < Fold_No; i++){
    clusterAmpTotals[indx][i] += instanceAmplitudes[i];

    if (clusterCount[indx] == 1){
      clusterMax[indx][i] = instanceAmplitudes[i];
      clusterMin[indx][i] = instanceAmplitudes[i];
    }
    else{
      if (instanceAmplitudes[i] > clusterMax[indx][i])
        clusterMax[indx][i] = instanceAmplitudes[i];
        if (instanceAmplitudes[i] < clusterMin[indx][i])
          clusterMin[indx][i] = instanceAmplitudes[i];
        }
      }
    }
    txt_all = "";

    for (int i = 0; i < Clust_No; i++){
       for (int j = 0; j < Fold_No; j++){
         txt_all += nf.format(clusterAmpTotals[i][j] /
         clusterCount[i]) + ",";
         txt_all += nf.format(clusterMax[i][j]) + ",";
         txt_all += nf.format(clusterMin[i][j]) + ",";
       }
    }

    txt_all += "l" + String.valueOf(filename.charAt(1))
     + "\n";
    outData.write(txt_all.getBytes());

 }
 outData.close();
 }
 catch (FileNotFoundException ex) {
   ex.printStackTrace();
 }
 catch (IOException ex){
   ex.printStackTrace();
 }
 finally {
   try {
    if (input!= null) {
```

```
        //flush and close both "input" and its
        // underlying FileReader
        input.close();
      }
  }
  catch (IOException ex) {
     ex.printStackTrace();
  }
}}}}
```

Appendix C

The Classifier Accuracies and the Confusion Matrices

The appendix shows the classifier accuracies and the confusion matrices of the instances grouped by folder and cluster numbers.

Number of Folders 4

Number of Clusters 2											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 51 (91.0714%)						Correctly classified 51 (91.0714%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	6	3	0	0	b	0	6	3	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	11	1	d	0	0	0	11	1	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 47 (83.9286%)						Correctly classified 27 (48.2143%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
10	1	0	0	0	a	11	0	0	0	0	a
3	4	2	0	0	b	0	6	3	0	0	b
0	1	10	1	0	c	0	0	8	4	0	c
0	0	0	11	1	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 3											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 51 (91.0714%)						Correctly classified 52 (92.8571%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 51 (91.0714%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class.as
10	1	0	0	0	a	11	0	0	0	0	a
0	9	0	0	0	b	0	7	2	0	0	b
0	1	10	1	0	c	0	0	8	4	0	c
0	0	1	11	0	d	0	0	10	2	0	d
0	0	0	1	11	e	0	0	10	2	0	e

Number of Clusters 4

Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	10	2	0	c	0	0	10	2	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 50 (89.2857%)						Correctly classified 50 (89.2857%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
9	2	0	0	0	a	9	2	0	0	0	a
1	8	0	0	0	b	1	8	0	0	0	b
0	1	9	2	0	c	0	1	9	2	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e

Number of Clusters 5

Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 52 (92.8571%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	7	2	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 51 (91.0714%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
10	1	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	10	1	0	c	0	0	8	4	0	c
0	0	1	11	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e

Number of Clusters 6

Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e

Mean values only						Maximum values only					
Correctly classified 50 (89.2857%)						Correctly classified 29 (51.7857%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
10	1	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	8	1	0	0	b
0	2	9	1	0	c	0	0	8	4	0	c
0	0	1	11	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 7											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 30 (53.5714%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	11	0	0	c	0	0	2	8	2	c
0	0	1	11	0	d	0	0	0	8	4	d
0	0	0	0	12	e	0	0	0	10	2	e
Number of Clusters 8											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 55 (98.2143%)						Correctly classified 55 (98.2143%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 54 (96.4286%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	11	0	0	c	0	0	8	4	0	c
0	0	0	12	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e

Number of Folders 6

Number of Clusters 2											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	9	0	0	0	b
0	0	10	2	0	c	0	0	10	2	0	c
0	0	0	11	1	d	0	0	0	11	1	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	0	10	2	0	c	0	0	8	4	0	c
0	0	0	11	1	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 3											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 29 (51.7857%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	8	4	0	c
0	0	1	11	0	d	0	0	10	2	0	d
0	0	0	1	11	e	0	0	10	2	0	e
Number of Clusters 4											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e

Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 29 (51.7857%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	8	4	0	c
0	0	1	11	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 5											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 54 (96.4286%)						Correctly classified 43 (76.7857%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	9	0	3	d
0	0	0	0	12	e	0	0	0	0	12	e
Number of Clusters 6											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	10	1	d	0	0	1	10	1	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 43 (76.7857%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	8	1	0	0	b
0	1	11	0	0	c	0	0	12	0	0	c
0	0	1	10	1	d	0	0	9	0	3	d
0	0	0	0	12	e	0	0	0	0	12	e

Number of Clusters 7											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	9	0	0	0	b	0	8	1	0	0	b
0	1	11	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	9	0	3	d
0	0	0	1	11	e	0	0	1	0	11	e
Number of Clusters 8											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	1	7	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 54 (96.4286%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	1	7	1	0	0	b
0	0	12	0	0	c	0	0	8	4	0	c
0	0	1	11	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e

Number of Folders 8

Number of Clusters 2											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 52 (92.8571%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	7	2	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e

Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	7	2	0	0	b
0	0	11	1	0	c	0	0	8	4	0	c
0	0	0	12	0	d	0	0	10	2	0	d
0	0	0	1	11	e	0	0	10	2	0	e
Number of Clusters 3											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	7	2	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 51 (91.0714%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
10	1	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	11	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	9	0	3	d
0	0	0	1	11	e	0	0	0	0	12	e
Number of Clusters 4											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	0	11	1	d	0	0	0	11	1	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	11	0	0	c	0	0	12	0	0	c
0	0	0	11	1	d	0	0	9	0	3	d
0	0	0	0	12	e	0	0	0	0	12	e

Number of Clusters 5											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	10	1	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	9	0	3	d
0	0	0	1	11	e	0	0	0	0	12	e
Number of Clusters 6											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 43 (76.7857%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	8	1	0	0	b
0	1	11	0	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	9	0	3	d
0	0	0	1	11	e	0	0	0	0	12	e
Number of Clusters 7											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e

Mean values only						Maximum values only					
Correctly classified 54 (96.4286%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	9	0	3	d
0	0	0	1	11	e	0	0	0	0	12	e
Number of Clusters 8											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 54 (96.4286%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	0	12	0	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	8	1	3	d
0	0	0	1	11	e	0	0	0	0	12	e

Number of Folders 10

Number of Clusters 2											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 51 (91.0714%)						Correctly classified 51 (91.0714%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
10	1	0	0	0	a	10	1	0	0	0	a
1	7	1	0	0	b	1	7	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 25 (44.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	10	1	0	0	0	a
1	7	1	0	0	b	1	5	3	0	0	b
0	0	11	1	0	c	0	0	8	4	0	c
0	0	0	12	0	d	0	0	10	2	0	d
0	0	0	1	11	e	0	0	10	2	0	e

Number of Clusters 3											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 51 (91.0714%)						Correctly classified 52 (92.8571%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
10	1	0	0	0	a	10	1	0	0	0	a
1	6	2	0	0	b	1	8	0	0	0	b
0	0	11	1	0	c	0	1	10	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 25 (44.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	10	1	0	0	0	a
0	9	0	0	0	b	1	5	3	0	0	b
0	2	9	1	0	c	0	0	8	4	0	c
0	0	0	12	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 4											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	1	7	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 26 (46.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	1	5	3	0	0	b
0	1	10	1	0	c	0	0	8	4	0	c
0	0	0	12	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 5											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	1	7	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e

Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 26 (46.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	1	5	3	0	0	b
0	1	10	1	0	c	0	0	8	4	0	c
0	0	0	12	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 6											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	1	7	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 34 (60.7143%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	1	7	1	0	0	b
0	1	11	0	0	c	0	0	6	5	1	c
0	0	0	12	0	d	0	0	0	8	4	d
0	0	0	1	11	e	0	0	0	10	2	e
Number of Clusters 7											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 52 (92.8571%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	1	7	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 26 (46.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	1	5	3	0	0	b
0	1	10	1	0	c	0	0	8	4	0	c
0	0	0	12	0	d	0	0	10	2	0	d
0	0	0	1	11	e	0	0	10	2	0	e

Number of Clusters 8											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 39 (69.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	6	3	0	0	b
0	1	11	0	0	c	0	0	11	1	0	c
0	0	1	11	0	d	0	0	1	1	10	d
0	0	0	1	11	e	0	0	0	2	10	e

Number of Folders 12

Number of Clusters 2											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 52 (92.8571%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	11	1	d	0	0	0	11	1	d
0	0	0	1	11	e	0	0	0	1	11	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	0	11	1	0	c	0	0	8	4	0	c
0	0	0	11	1	d	0	0	10	2	0	d
0	0	0	1	11	e	0	0	10	2	0	e
Number of Clusters 3											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e

Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 28 (50%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
10	1	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	11	0	0	c	0	0	8	4	0	c
0	0	1	11	0	d	0	0	10	2	0	d
0	0	0	0	12	e	0	0	10	2	0	e
Number of Clusters 4											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	7	1	0	0	b	0	7	2	0	0	b
0	1	10	1	0	c	0	0	12	0	0	c
0	0	0	12	0	d	0	0	9	0	3	d
0	0	0	0	12	e	0	0	0	0	12	e
Number of Clusters 5											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 53 (94.6429%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	8	1	0	0	b
0	0	12	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	11	0	0	c	0	0	12	0	0	c
0	0	1	11	0	d	0	0	9	0	3	d
0	0	0	0	12	e	0	0	0	0	12	e

Number of Clusters 6											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 53 (94.6429%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	10	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	8	1	3	d
0	0	0	0	12	e	0	0	0	0	12	e
Number of Clusters 7											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 54 (96.4286%)						Correctly classified 54 (96.4286%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	8	1	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	0	12	0	d	0	0	0	12	0	d
0	0	0	0	12	e	0	0	0	0	12	e
Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 46 (82.1429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	8	1	0	0	b
0	1	10	1	0	c	0	0	6	6	0	c
0	0	0	12	0	d	0	0	6	6	0	c
0	0	0	1	11	e	0	0	0	0	12	e
Number of Clusters 8											
Mean, Maximum, and Minimum						Mean and Maximum					
Correctly classified 52 (92.8571%)						Correctly classified 53 (94.6429%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
0	7	2	0	0	b	0	8	1	0	0	b
0	0	11	1	0	c	0	0	11	1	0	c
0	0	1	11	0	d	0	0	1	11	0	d
0	0	0	0	12	e	0	0	0	0	12	e

Mean values only						Maximum values only					
Correctly classified 52 (92.8571%)						Correctly classified 42 (75%)					
a	b	c	d	e	← class. as	a	b	c	d	e	← class. as
11	0	0	0	0	a	11	0	0	0	0	a
1	8	0	0	0	b	0	7	2	0	0	b
0	1	10	1	0	c	0	0	11	1	0	c
0	0	1	11	0	d	0	0	8	1	3	d
0	0	0	0	12	e	0	0	0	0	12	e

BIODATA OF THE AUTHOR

BIODATA OF THE AUTHOR

Mr. Addin Osman Mohamed has obtained his B.Sc. degree in Computer Science from Istanbul University in 1995. After getting his B.Sc., he worked for different private companies in Turkey where he developed many software applications for educational and health sectors. Also he taught some courses related to Computer Science in a private college in Istanbul.

In 2001, he started studying a master program in Computational Engineering at Erlangen University - Germany. After finishing his first semester, he had been awarded a scholarship in cooperation with the company of Hewlett Packard (HP) and the Danish Government to pursue his master degree in Aalborg University - Denmark. In 2003, he obtained a master of Software Engineering from Aalborg University. After getting his master degree he worked as a teaching research assistant for the Computer Science Department at Aalborg University from 2003 to 2004. As part of this job, he has taken part in an *EU* funded project (the MERIT project). The MERIT project, the Management of the Environment and Resources using Integrated Technologies, was a three-year project that started in 2001. The aim of the project was to generate a highly adaptable, integrated water resource assessment and management tool that can be applied at the catchments and aquifer unit scale throughout Europe. The Bayesian networks had been considered as the basic framework. Mr. Addin, has also worked as a teaching research assistant at the Faculty of Engineering - University Putra Malaysia (2005-2006).

During his Ph.D. studies, Mr. Addin has published the following papers:

- Journal papers:

  1. A Naïve bayes classifier for damage detection in engineering materials, Materials And Design (2006), UK (Submitted).

  2. Prediction and detection of failures in composite materials using neural networks$---$review (2006), Polymer and Polymer Composites, UK (Accepted).

- Conference papers and Presentations:

  1. Bayesian network approach to classify damages and f-folds feature extraction algorithm in engineering materials, International Conference on Composite Materials and Nano-Structures, Shah Alam, Malaysia, 2006.

2. f-FFE : f-Folds feature extraction for structural health monitoring systems, A Series of Seminar and Frontier Science and Advanced Technology, Institute of Advanced Technology, University Putra Malaysia, Malaysia, 2006.

3. An intelligent system for failure prediction in LCMS using Bayesian networks, A Series of Seminar and Frontier Science and Advanced Technology, Institute of Advanced Technology, University Putra Malaysia, Malaysia, 2005.

4. Bayesian network approach to classify damages and f-folds feature subset selection method in laminated composite materials , International Conference on Intelligent Systems and Robotics iCISAR2005, Putrajaya, Malaysia, 59, 6-8 December 2005.