

2013

Embedded sensor coherency for disaster assessment

Muttaz Khaled Mohammed Mohammed

University of Wollongong

Recommended Citation

Mohammed, Muttaz Khaled Mohammed, Embedded sensor coherency for disaster assessment, Master of Engineering - Research thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, 2013. <http://ro.uow.edu.au/theses/4083>

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

**UNIVERSITY OF
WOLLONGONG**



**School of Electrical, Computer and Telecommunications
Engineering**

Embedded Sensor Coherency for Disaster Assessment

Muttaz Khaled Mohammed Mohammed

**This thesis is presented as part of the requirements for the
award of the Degree of
Master of Engineering by Research
of the
University of Wollongong**

August 2013

Statement of Originality

I, Muttaz Mohammed, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Master of Engineering-Research, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Muttaz Mohammed

August 31, 2013

ABSTRACT

During natural disasters, rescue teams fight against time to save as many civilians as they can. Researchers can contribute to rescue operations by investigating different ways of collecting information from the crisis area and delivering it to rescue teams. This project proposes a novel approach for collecting information from disaster areas in relation to building collapses and collapse pattern analysis. Our approach is based on classifying building collapse patterns using Wireless Sensor Networks (WSNs) and data-mining algorithms. Classification time and reliability are considered crucial factors, and one of the main objectives of this research is to improve these elements in order to deliver accurate information to rescue teams regarding building status in a stricken area. WSNs were installed in a simulated building to capture a building's motion during an earthquake. Four different types of collapse patterns were simulated: first column (FC), first storey (FS), mid-storey (MS) and pancake (PCK). The captured data was inputted to three different classification algorithms (PCA, VQH and HMM) to classify building collapse types.

Two real-life case scenarios were designed to examine the algorithms' reliability under sensor failure. The first scenario was sensor failure on impact, which was designed to simulate sensor failure caused by interfering with an object or hitting the ground. The second scenario was the complete failure of random sensors, which was designed to simulate early malfunctioning sensors resulting from a power supply failure, communication problem or manufacture error. Moreover, the second scenario was designed to investigate the limit of each classification algorithm in terms of the number of failed sensors. The Hidden Markov Model (HMM) proved the most robust

and achieved 100% accuracy in least impact on accuracy (LIoA) case scenario and 60% accuracy as an algorithm limit when 33.3% of the sensors failed during a building collapse. By achieving this level of accuracy, the objective of classifying four possible collapse patterns in a short processing time was achieved.

ACKNOWLEDGEMENTS

The author wishes to thank several people. I would like to thank my parents, for their love, kindness and support they have shown during the past two years it has taken me to finalise this thesis.

Moreover, I would like to express my deepest appreciation to my supervisor Dr. Montserrat Ros and co-supervisor Dr. Golshah Naghdy as well for their assistance and guidance with this thesis.

Furthermore I would also like to acknowledge with much appreciation the crucial role of the staff of school of Electronic, Computer, and Telecommunication Engineering, who gave the permission to use all required equipment and the necessary materials to complete my experiments.

I also wish to thank Dr. Neaz Sheikh, Ahmed Manhl from the Faculty of Civil, Mining and Environmental Engineering and Ge Wang from the School of Electrical, Computer & Telecommunications Engineering for their help and support.

Finally, special thanks go to my roommates John, Megan, and Boris, who have encouraged and helped me during my research journey.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
LIST OF TABLES	x
Acronyms	xii
Chapter 1: Introduction	1
1.1 Research Objectives	4
1.2 Research Methodology	5
1.3 Thesis Organisation	7
1.4 Research Contributions	8
Chapter 2: Literature Review	10
2.1 Remote Sensing Images for Disaster Assessments	11
2.1.1 Satellite imagery for disaster assessment	11
2.1.2 Aerial imagery for disaster assessment	13
2.2 Building Structural Health Monitoring Systems	16
2.3 Theoretical Background on Classification Methods	19
2.3.1 Principal Component Analysis	19
2.3.2 Vector Quantisation	20
2.3.3 Hidden Markov Models	25
2.4 Previous Works on Classification Methods	30
2.4.1 Principal Component Analysis Implementations	30
2.4.2 Vector Quantisation Implementations	34
2.4.3 Hidden Markov Model Implementations	37
2.5 Chapter Summary	39
Chapter 3: Building Collapse Patterns and Simulation	40
3.1 Introduction	40
3.2 Building Collapse Pattern Types	42
3.3 Building Collapse Pattern Simulations	47
3.4 Training and Testing Databases	54
3.5 Summary	55

Chapter 4: Experimental Results.....	56
4.1 Sensor Behaviour During Building Collapse.....	56
4.2 The Experimental Results using PCA Algorithm.....	58
4.2.1 Basic PCA.....	58
4.2.2 Two-level PCA.....	61
4.2.3 Sensor-by-sensor based PCA.....	64
4.3 Vector Quantisation Histogram.....	66
4.4 Hidden Markov Model Results.....	70
4.5 HMM-PCA Hybrid Algorithm.....	73
4.6 Discussion.....	75
Chapter 5: Sensor Failure Scenarios.....	77
5.1 Scenario 1: Sensor Failure on Impact (SFoI).....	78
5.1.1 Experimental Setup.....	80
5.1.2 Basic PCA Algorithm Performance under SFoI.....	80
5.1.3 Two-level PCA under SFoI.....	82
5.1.4 Sensor-by-sensor PCA under SFoI.....	84
5.1.5 Vector Quantisation Histogram under SFoI.....	88
5.1.6 Hidden Markov Model under SFoI.....	89
5.2 Scenario 2: Complete Failure of Random Sensors (CFoRS).....	91
5.2.1 Basic PCA under CFoRS.....	92
5.2.2 Two-level PCA under CFoRS.....	94
5.2.3 Sensor-by-sensor PCA under CFoRS.....	96
5.2.4 VQH under CFoRS.....	97
5.2.5 HMM under CFoRS.....	102
5.2.6 HMM-PCA Hybrid Algorithm under CFoRS.....	104
5.3 Chapter Summary.....	105
Chapter 6: Conclusion.....	106
6.1 Thesis Summary.....	106
6.2 Conclusion.....	108
6.3 Future Work.....	111
REFERENCES.....	113

LIST OF FIGURES

Figure 1.1: Rescue operations after an earthquake in Turkey [1].	1
Figure 1.2: An example of a satellite images for earthquake-stricken zone (a) before earthquake (b) after earthquake [8].	3
Figure 2.1: Building classification in a stricken area: damaged buildings outlined in red, undamaged buildings in blue, and yellow represents unclassified building [8].	14
Figure 2.2: (a) the simulated building (b) the experimental results of four cases [28].	17
Figure 2.3: Fuzzy inference system for damage detection [31].	18
Figure 2.4: An example of binary VQ tree structure for $N_c = 8$ [33].	24
Figure 2.5: HMM example [53].	27
Figure 2.6: Using first and second PC breast cancer data clustering [67].	32
Figure 2.7: Typical examples of histograms [21].	35
Figure 3.1: Remote sensing image used for building collapse detection by a research institution in Iran during the Bam earthquake in 2003[75].	41
Figure 3.2: Shows the horizontal and vertical forces that can cause building collapse [79].	43
Figure 3.3: On the left, Ronan Point building local column collapse in 1968 [63]. On the right, a sketch of local column collapse [79].	44
Figure 3.4: First-storey collapse of residential building with shops in Koriyama city Japan after 2011 earthquake [17].	45
Figure 3.5: Sixth-storey collapse in the eight-storey Kobe city hall building in the 1995 Japan earthquake [81].	46
Figure 3.6: A collapsed four-storey building in the Turkey earthquake [82].	47
Figure 3.7: The basic building prototype that was used in this research.	49
Figure 3.8: Soil profile for the simulated building [90].	51
Figure 3.9: Building collapse patterns. From top to bottom: first column (FC), first-storey (FS), mid-storey (MS) and pancake (PCK) collapse patterns.	52
Figure 3.10: Illustrates the position of wireless sensors in each block.	54
Figure 4.1: Process steps of basic PCA.	58

Figure 4.2: shows two-level PCA decision making process.	62
Figure 4.3: Decision-making rules for sensor-by-sensor based PCA.	65
Figure 4.4: VQH implementation process.	67
Figure 4.5: PAA breaking points.	71
Figure 5.1: Sensor recording in three different cases: (a) theoretical case, (b) hitting the ground and (c) hitting an object during a fall.	79
Figure 5.2: Basic PCA decision rules.	81
Figure 5.3: Two-level PCA methodology.	83
Figure 5.4: Number of sensors that can generate a reliable rule for each collapse pattern under SFoI.	85
Figure 6.1 shows classification algorithms results under different case scenarios. .	109

LIST OF TABLES

Table 4.2: Decision rules for FS and PCK collapse patterns.	59
Table 4.3: Basic PCA results and Confusion matrix.....	60
Table 4.4: Two-level PCA results and Confusion matrix.	63
Table 4.5: Sensor-by-sensor based PCA results and Confusion matrix.....	66
Table 4.6: Final results of the PCA algorithm.	66
Table 4.7: VQH algorithm result and Confusion matrix.....	68
Table 4.9: HMM results.	72
Table 4.10: Confusion Matrix of HMM (a) using PAA (five breaking points) (b) using PAA (two breaking points) (c) using PAA (three breaking points).	73
Table 4.11: HMM_PCA results.	74
Table 5.1: Basic PCA results and confusion matrix under SFoI.....	82
Table 5.2: Two-level PCA results and confusion matrix under SFoI.	84
Table 5.3: Sensor-by-sensor PCA results and confusion matrix under SFoI.....	86
Table 5.4: Final results of the PCA algorithm under SFoI.....	87
Table 5.5: VQH results and confusion matrix under SFoI.....	88
Table 5.6: HMM results and confusion matrix under SFoI.	90
Table 5.7: Basic PCA results (LloA) and confusion matrix under CFoRS.	93
Table 5.8: Basic PCA results (MloA) and confusion matrix under CFoRS.	94
Table 5.9: Two-level PCA results (LloA) and confusion matrix under CFoRS.....	95
Table 5.10: Two-level PCA results (MloA) and confusion matrix under CFoRS.....	95
Table 5.11: Sensor-by-sensor based PCA results (MloA) and confusion matrix under CFoRS.	96
Table 5.12: VQH results (MloA) and confusion matrix under CFoRS.	97
Table 5.13: VQH results (MS sensors failure) and confusion matrix under CFoRS.	98
Table 5.14: An example of two misclassified unknown patterns.	100
Table 5.15: HMM results and confusion matrix under CFoRS.	102

Table 5.16: Confusion matrices of first- and second-storey cases using the HMM algorithm.	103
Table 4.1: Examples of sensor readings from different locations.....	156
Table 4.8: Example of generated histograms using VQ codebooks.	161

ACRONYMS

AF	Auto-feedback
AFX	Auto-feedback with exogenous input
BW	Baum-Welch
CHMM	Continuous Hidden Markov Model
CFoRS	Complete Failure of Random Sensors
CVQ	Classified Vector Quantisation
DHMM	Discrete Hidden Markov Model
DPGrid	Digital Photogrammetry Grid
FC	First Column
FIS	Fuzzy Inference System
FS	First Storey
GPS	Global Positioning System
HMM	Hidden Markov Model
ICs	Integrated Circuits
LBG	Linde-Buzo-Gray
LloA	Least Impact on Accuracy
MCEER	Multidisciplinary Centre for Earthquake Engineering Research
MFCCs	Mel Frequency Cepstral Coefficients
MloA	Most Impact on Accuracy
MS	Mid-Storey
MSE	Mean Square Error
NCEER	National Centre for Earthquake Engineering Research
PAA	Piecewise Aggregate Approximation

PC	Principle Component
PCA	Principle Component Analysis
PCK	Pancake
PLS	Partial Least Squares
PNN	Pairwise Nearest Neighbour
PSNR	Peak Signal to Noise Ratio
RGB	Red Green Blue
RMA	Rendering and Matching Analysis
ROI	Region of Interest
SAR	Synthetic Aperture Radar
SCHMM	Semi-continuous Hidden Markov Model
SFoI	Sensor Failure on Impact
SHM	Structure Health Monitoring
SSIM	Structural Similarity
SVM	Support Vector Machine
UBC	Uniform Building Code
UC	Uncollapsed
US	United State
VQ	Vector Quantisation
VQH	Vector Quantisation Histogram
WSN	Wireless Sensor Network
3D	Three Dimensions
2D	Two Dimensions

Chapter 1: Introduction

Natural disasters can strike cities in many different ways. Some natural disasters, such as floods and tornadoes, may be predicted, which allows sufficient time to evacuate civilians. Others, such as earthquakes and tsunamis, hit suddenly, so there is no warning to enable the timely evacuation of people (see Figure 1.1). This can lead to high numbers of casualties.



Figure 1.1: Rescue operations after an earthquake in Turkey [1].

Earthquakes that have affected cities have resulted in 30–70% destruction of buildings and infrastructure [2, 3]. This degree of destruction occurs because not all buildings have been constructed to anti-seismic standard. As a high percentage of buildings are typically occupied (for a range of purposes), building collapse is the main cause of death during an earthquake [4]. Therefore, many countries over the past decade have introduced standards for anti-seismic construction. However, these generally do not apply retrospectively and not all countries can afford the high financial cost.

Despite the high number of casualties from earthquakes, not all victims are killed instantly: some may survive for several hours, or as long as three days [5, 6]. This provides only a narrow opportunity to save lives, so rescue teams must work efficiently and professionally to maximise the chances of survival. Rescue teams place a high priority on rescuing people who are known to be alive [4]. One of the most important factors in rescue teams achieving their goals is the accuracy and availability of information about the earthquake-stricken area [5]. This information includes earthquake strength, population, road condition, communication, and position and number of collapsed buildings [5]. More details mean saving time and thus lives [5].

Authorities can gather a wide range of information at two different stages:

- 1) after the earthquake strikes and before rescue teams enter the area
- 2) after rescue teams reach the crisis zone.

In the first stage, satellite and aerial images of the stricken areas are vital information resources for rescue teams [7-9]. The crisis control room can access a wide range of information about the crisis area based on these images (see Figure 1.2). For instance, images can provide information about the position and number of damaged buildings [7, 8]. Moreover, knowledge of the population of the stricken area can assist in estimating the number of casualties [10]. However, the level of detail of such information is insufficient for building rescue operations. Most systems classify buildings as either collapsed or un-collapsed [7, 11], and others as either damaged or undamaged [8, 12]. These types of classification can be useful for general plans, but not for individual rescue operations, because they cannot provide the rescue team with a complete picture of the collapse type. In addition, rescue teams cannot

prioritise their work or determine what equipment is necessary using this kind of classification.



Figure 1.2: An example of a satellite images for earthquake-stricken zone (a) before earthquake (b) after earthquake [8].

In the case of individual buildings, rescue teams use the second approach: gathering the missing information once they are in the crisis zone [5]. Building collapse type is usually determined by direct observation once on-scene [13]. To gain information about internal damage to the building infrastructure, rescue teams may employ mobile robots because of the danger to human rescuers [14, 15]. This vital information is sent to the control unit, which then mobilises the necessary workers, food and equipment for the specific rescue operation. However, this approach is much less time efficient than obtaining all of the necessary information prior to leaving the station, which would allow rescuers to be better prepared and to prioritise locations.

Most rescue operations in the stricken area will be based around collapsed buildings. Therefore, providing more information on the nature of those collapsed buildings can

help rescue teams in their operations. During earthquakes, buildings collapse in different ways [16]. Examples of collapse patterns include first column, first storey, mid-storey and pancake [17]. Information on how the target building has collapsed will assist rescue teams in planning the rescue operation, and will answer many questions, such as:

1. How should they enter the building?
2. What is the best equipment to use in the rescue operation?
3. How many workers are needed?
4. In what voids are survivors likely to be found?
5. Should the building be considered a high priority compared to other collapsed buildings?

For instance, in a pancake collapse, the chance of civilians surviving is much lower than in other collapse types [4]. In first-storey collapses, rescue teams know that all building entrances will be blocked by debris.

The research work in this thesis focuses on classifying building collapse patterns caused by earthquakes. Different collapse patterns are simulated using the Blender game engine. For each model, a simulated wireless sensor network (WSN) records building behaviour during the earthquake. The WSN's output feeds into the classification algorithm that provides information regarding whether or not the building has collapsed, and its collapse pattern.

1.1 Research Objectives

The objective of this project is to design and examine three algorithms to classify building collapse patterns resulting from natural disasters. The study builds on

previous research on structural monitoring systems and classification algorithms. To date, the focus of most research work has been on pre-emptive building damage detection and structural monitoring systems including WSNs [18]. Such research has emphasised building monitoring to assist civil engineers in building maintenance and design, but few studies have been in the area of this research [19, 20], and thus substantial pioneering research work is needed. While structural monitoring systems use WSNs to record the forces exerted on buildings during earthquakes, the research in this thesis uses WSNs to record the three-dimensional (3D) acceleration sensory data of falling or moving building parts. This information is used in algorithms implemented to classify building collapse patterns.

Finally, a comparison of algorithms that have been used in the classification of building collapse patterns is presented, leading to a recommendation of the most reliable and suitable algorithm.

To achieve the project goals, the research proceeds as follows:

1. Simulating building collapse patterns
2. Designing WSNs for building collapse identification
3. Examining output and implementing building collapse pattern classification algorithms
4. Examining classification algorithms under sensor failure scenarios.

1.2 Research Methodology

The approach chosen to achieve the research objectives is based on simulating real-world collapse scenarios without including detailed structure model, which would increase the research complexity without enhancing the outcome. It proceeds by:

1. Simulating building collapse patterns. Four collapse patterns are simulated using the Blender game engine software. During simulations, sensor information (e.g. position, speed and time) is extracted using simulated WSNs and the Python program cooperating with the Blender game engine. This information is used to study the behaviour of WSNs during building collapse by feeding it into the chosen classification algorithms.
2. Designing WSNs for building collapse classification. To design WSNs for this research, the building structure itself must be identified. In general, WSN design considers two primary components of the building structure: the floors and the body of the building. Sixteen sensors are installed on each floor and on the outside of the building body without specifying a particular topology, which is beyond the scope of this research.
3. Examining output and implementing building collapse pattern classification algorithms. After simulating the building collapse, the data was ready to be used by algorithms that can classify the building collapse patterns. Three methods of pattern recognition are used, such as principal component analysis (PCA), hidden Markov models (HMMs), and vector quantisation (VQ). The study also involves algorithm implementation and integration. The outcome of this process is the classification of four collapse patterns, plus the un-collapsed building pattern, into five categories based on features extracted from the pattern database by the classification algorithms. This work may lead to further advancements in classifying building collapse, and thus reduce casualties.
4. Examining the classification performance under sensor failure scenarios. This stage of the research aims to increase overall system reliability by examining

the classification algorithm performance under the influence of sensor failure, which is typical during a real-world building collapse. Two scenarios are under investigation:

- a) The first scenario covers sensor failure due to debris impact or impact with an object falling at a certain speed. This scenario represents a situation in which WSNs are maintained regularly and there are no randomly malfunctioning sensors or issues with their power source.
- b) The second scenario is an additional situation added to first scenario. It covers the early malfunctioning of sensors caused by either manufacturing issues or failure in the sensor power supply. Each classification algorithm is challenged to stand on its limit and capability before it fails.

1.3 Thesis Organisation

Many different disciplines are involved in this research: pattern recognition and data mining in classification algorithms; earthquake engineering in building collapse patterns and earthquake force; and Blender game engine and other software such as Python and MATLAB in building collapse simulations, algorithms and WSNs implementation. Also important is disaster aid research, which is the main motivation behind this study. The thesis is organised as follows:

- **Chapter 2** presents a comprehensive literature review in the areas of crisis relief and classification algorithms. It also provides a background to the technologies and methods used in the research in relation to crisis aid and classification algorithms.

- **Chapter 3** presents an overview of building collapse patterns, which are the focus of this research, and an explanation of how the Blender game engine is used to simulate building collapse patterns.
- **Chapter 4** presents the results of the classification algorithms research. The classification algorithms used in this research are PCA, VQ and HMMs. These results are considered ‘theoretical’ because sensors are not allowed to fail under any circumstances. This chapter presents a comprehensive analysis and discussion of the key results.
- **Chapter 5** examines the classification algorithm performance when sensor failure occurs. Two main scenarios of sensor failure are investigated, each covering a real-world scenario that could be faced by any WSN during an earthquake.
- **Chapter 6** presents the final discussion and conclusions of this research. It includes a comparison of classification algorithms and a final recommendation regarding which algorithm is more reliable, considering algorithm performances during the experiments.

1.4 Research Contributions

The main contributions of this thesis are:

- An investigation into building collapse pattern classification using classification techniques and sensory data collected from WSNs. Such a classification scheme has not been used to classify collapse patterns previously.
- An assessment and comparison of the performance of three algorithms (PCA, VQ and HMMs), and the use of this information to develop final recommendations. It is driven from the outcome results of algorithms

performance in classifying data of five different collapse patterns that captured during building collapse simulation.

- An examination of the effect of sensor failure on the above classification algorithms. Two scenarios are examined: failure on impact and complete failure of random sensors. Each scenario covers a certain number of challenges that can be faced by WSNs during real-world building collapses. These give a measure of robustness of the classification algorithms.

Chapter 2: Literature Review

The primary goal of this chapter is to present a thorough background of the various technology and research areas related to rescue operation after earthquakes as well as the methods and algorithms used in this study to reach the final goal: classifying building collapse during earthquakes. Section 2.1 provides an overview of the work done in other studies towards helping rescue teams during earthquakes by classifying damaged buildings in the stricken areas through satellite and aerial images. This section explains what modern research has achieved in terms of building classification. Section 2.2 provides an overview of how wireless sensor networks (WSNs) are used in structural health monitoring (SHM) systems. This section shows how civil engineers have used WSNs to serve their needs regarding to measure vibration level and structure strength during an earthquake. In addition, it highlights the gap in collapsed buildings between the aspect of civil engineering and the aspect of rescue teams. Section 2.3 discusses the theoretical background of pattern recognition algorithms that have been used in this thesis. Finally, an overview is provided of studies that have used these algorithms to interpret and classify data collected from different types of implementation such as motion and pattern recognition. A wide range of algorithm implementations have been used to examine the algorithms' performance under different data that were generated in various environments.

2.1 Remote Sensing Images for Disaster Assessments

Image processing research is involved in a wide variety of applications [10, 21, 22]. One of the main reasons for this is the immediacy of the information and data provided by images [7]. When natural disasters such as earthquakes strike, the immediate information is very useful for drawing a complete picture about the stricken areas and provides technical support for rescue operations [7]. Remote sensing images can be produced in a variety of ways, including by satellite or aerial imagery. The following sections show the recent works in emergency aid using satellite and aerial imagery. In addition, they show strength and weakness of each technique.

2.1.1 Satellite imagery for disaster assessment

Researchers had taken advantage in the disaster assessment area of remote sensing technology development. Most developed countries have multifunctional satellites that serve disaster assessment by providing stricken areas' images before and after natural disasters.

Dongjian *et al.* [12] took the advantage of satellite technology by investigating the features of buildings and roads damaged by earthquakes in high-resolution images taken by the satellite remote sensor GeoEye-1, which launched in 2008. This remote sensor can provide high-resolution images with an update every two to three days. Shape, colour, shadow and texture are the main features for diagnosing damaged buildings or roads. The interpretation of each image based on these features needs a clear and high-resolution image. The authors explained the benefit of this method, which helps to reduce the number of casualties after each earthquake. They described the main features that other researchers have used or suggested to locate the

collapsed buildings and to identify such details as building dimensions and roof materials.

With a similar goal, Masfumi *et al.* [10] used satellite images to diagnose a disaster stricken area. Using information about the earthquake such as calculations of the earthquake's intensity and magnitude and the crisis area in terms of population and number of buildings, the authors were able to estimate the number of collapsed buildings dependent on the building types in the crisis zone using the magnitude distance attenuation equation. Instead of locating damaged buildings, the results showed images that represented the seismic area and number of casualties in it. These results were based on vital information about the area itself. Masfumi *et al.* posit that this research would be useful for global rescue operations, but they did not mention whether it could be used for local rescue operations.

To summarise, satellite imagery has its advantages and disadvantages. Advantages are:

- Swift method to collect information.
- Low cost solution if satellite already exists.
- No need for training staff to take images because it is a computer based methodology.
- Suitable for global or general rescue operations plans.

This method has its disadvantages which are:

- Satellite technology is not available in most developing countries.
- Images cannot cover a wide area and locate buildings across large areas simultaneously.
- Image resolution is the prime barrier for large areas especially during the harsh weather conditions after natural disasters.
- The types of driven information from satellite images do not reach rescue teams' expectation in term of helping them in ongoing operations.

2.1.2 Aerial imagery for disaster assessment

It is another known methodology to obtain images related to the stricken area. An airplane occupied with high resolution camera travel over the stricken area in order to take images for certain areas or creating a grid of images covered whole stricken area.

Dominik *et al.* [8] used high-resolution synthetic aperture radar (SAR) images to locate the damaged buildings after the earthquake. They estimated the three dimensions for each building in the post-images of the stricken area. In the second stage, they estimated a signature for each building using rendering and matching analysis (RMA). These signatures were used to find similarities between recent and latest images. High similarity indicated that buildings were undamaged. Buildings were classified into two categories: damaged and undamaged buildings (see Figure 2.1). The results accuracy rate was 90 % based on 30 buildings that had various dimensions. This is one of the high accuracy scored works on crisis assessment, especially in locating damaged buildings.

Jianwen and Sixian [7] depicted a novel algorithm that can detect a collapsed building using a high-resolution image taken after the earthquake only, whereas most research in this area is based on a comparison of images taken before and after earthquakes. The authors claimed that by understanding the relationship between building morphology behaviour in high-resolution images and the electromagnetic response of pixels in the image, they achieved 80 % output accuracy in classifying collapsed or uncollapsed buildings. They obtained the images from an aircraft flying over the stricken area after the earthquake. It took 30 minutes to cover the whole area. This algorithm has already been used in China in the 2008 and 2010 earthquakes.



Figure 2.1: Building classification in a stricken area: damaged buildings outlined in red, undamaged buildings in blue, and yellow represents unclassified building [8].

Considering the computing time required for a large number of images taken by an aircraft over 30 minutes and the size of the data collected, Tao *et al.* [23] solved this problem by using parallel processing to process the Digital Photogrammetry Grid (DPGrid) of aerial images. The system can process thousands of images and build the DPGrid with the help of Global Positioning System (GPS) in 111 hours. This computing time is six times faster than serial processing. However, four days processing time is not suitable for rescue operations.

Sumer and Turker [24] used aerial images that were taken before and after an earthquake for building damage assessments. From the post disaster aerial images, the authors used the grey level to identify the boundary of each building in the crisis zone and save them in a boundary vector. Boundary vector helps to focus on the buildings only, neglecting any other objects such as roads, trees and cars. They found that collapsed buildings had a high grey level compared with uncollapsed buildings. The final result of this approach was 89.44 % accurate. Sumer and Turker mentioned that light intensity can vary from one building to another based on the roof materials.

To summarise, aerial imagery has its advantages and disadvantages. Advantages are:

- Locating buildings in stricken area and classifying them as either damaged or undamaged, which is useful for rescue teams.
- Does not need advance technology to gather images and achievable in most developing countries.
- Can cover a wide area in the stricken area.
- Can avoid the weather condition by flying at varying altitude.

This method has some disadvantages and they are;

- Time consuming process that can exceed 3 days, which is not suitable for human rescue.
- Requires highly trained pilot to fly over the stricken area who may be not available in developing countries.
- Simple buildings' classification that cannot determine priority between collapsed buildings.

2.2 Building Structural Health Monitoring Systems

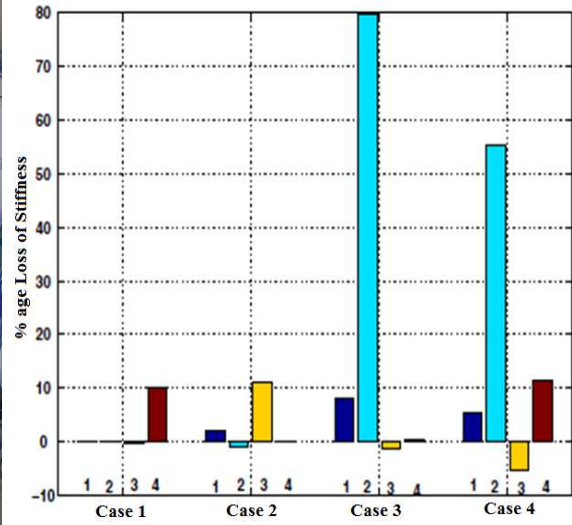
In this section, examples of how WSNs have been used in monitoring buildings structures are presented. These examples provide a good introduction to the types of information that civil engineers are interested to investigate [25-30]. Damage detection and identification have been targeted by many researchers in various works. Krishna *et al.* [28] addressed damage detection and localisation issues using structural health monitoring systems (SHMs). They investigated two SHM systems, Wisden and NetSHM. A laboratory prototype of a four-storey building used in this study is shown in Figure 2.2a. MicAz sensors were used to gather data and a vibration device was used to make the building vibrate during the simulation. Figure 2.2b shows the NetSHM systems results in damage detection and localisation. The results show how each storey lost stiffness because of vibration during the building's life cycle.

The authors did not give any indication of when the building was going to collapse because of stiffness loss. In [28], sensors in both systems recorded the vibrations and sent them to a server to implement detection and identification algorithms that can classify the level of vibration and its impact on each story of the building.

In contrast, Ajay *et al.* [29] focused on designing a powerful wireless sensor node that could perform damage detection and identification algorithms for SHM systems. They used a three-storey laboratory structure as a research environment. A time-history database was created in the sensor node that represented the undamaged building coefficients. These coefficients were generated using auto-feedback (AF) models. The residual error of the AF model, which represents the sensitivity feature,



(a)



(b)

Figure 2.2: (a) the simulated building (b) the experimental results of four cases [28].

was produced by computing the Euclidian distance between the current coefficient vectors of the AF model and the time-history database. An auto-feedback with exogenous input (AFX) time-series model was fitted in the sensor node to find the correlation between the residual error of the AF model and the measured response. This was novel work because it implemented two damage detection and identification algorithms in a sensor node. However, the authors' damage detection and identification did not reach a level that can help other researchers interested in crisis relief, because damage was investigated on concrete strength and not based on whole building.

Liang *et al.* [30] focused on using three different types of sensor readings to monitor the strain of a single location in a cantilever beam (see Figure 2.3). The authors used a fuzzy inference system (FIS) to process three different types of measurement that came from an accelerometer, strain gauge and piezoceramic transducer.

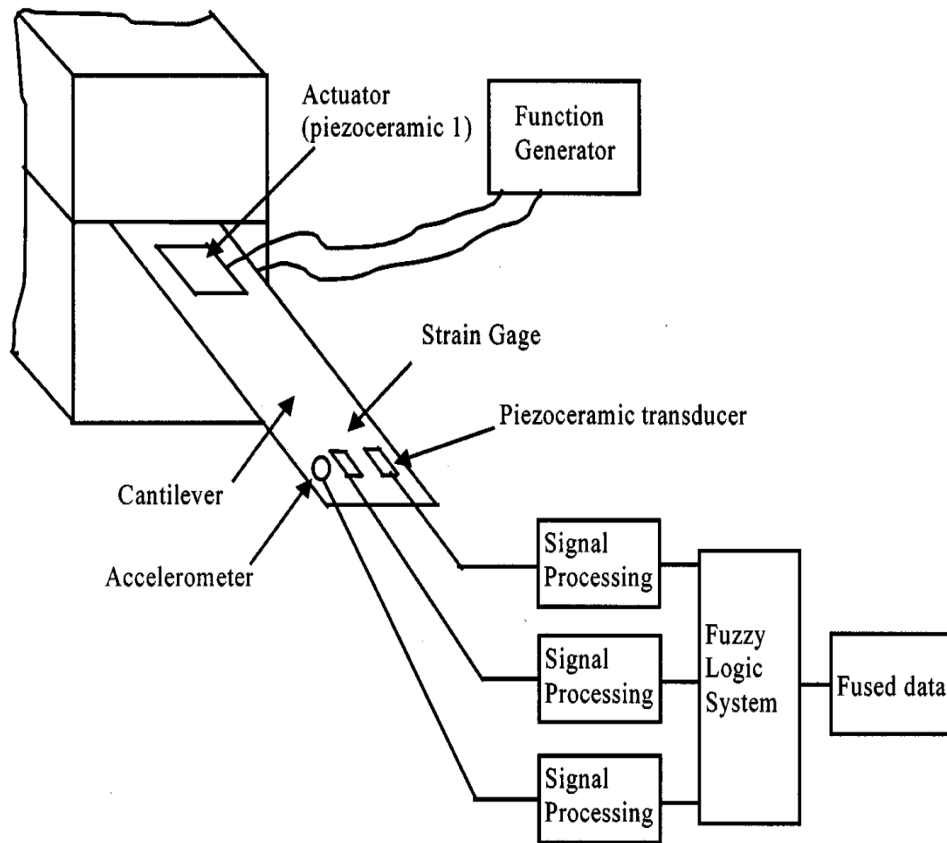


Figure 2.3: Fuzzy inference system for damage detection [31].

They detailed how they used membership functions and weights to detect and ignore the incorrect sensor reading to achieve accurate results. Based on field-gathered data and simulated FIS in MATLAB, the authors achieved a low percentage of errors in the fused data, which was the aim of the fusion model.

As shown from above examples [28-30], WSNs have been used to gather various types of information and analyse it in different algorithms to serve only one goal: to discover how different parameters (such as temperature and vibration) affect concrete strain. Concrete strain is considered one of the main parameters that civil engineers use in deciding if a building can be occupied or not. In addition, civil engineers consider a building is collapsed if there is a crack in one of the main beams

[30], which reflects the meaning of building collapsed terminology from their point of view. As a result, WSNs have not yet been used to detect and classify a physical building collapse.

2.3 Theoretical Background on Classification Methods

In this section, we preset a theoretical background on principal component analysis (PCA) algorithm, Vector quantisation (VQ), and Hidden Markov models (HMM) that used in this thesis. The aim of this section is to present an overview on each classification methods from the aspect of mathematical background and features.

2.3.1 Principal Component Analysis

Principal component analysis (PCA) is one of the best-known multivariate analysis techniques [31-35]. Authors maintain it first appeared in 1901 in a study by Pearson and was later developed by Hotelling in 1933 [32]. PCA is used primarily for both dimension reduction or pattern recognition [35-38]. PCA has the capability of processing uncorrelated dimensions in high-dimension datasets in a short time to less correlated dimensions, depending on the similarity and difference between dimensions. PCA can give an indication of the most important dimensions and the relationship between features in the same dimension. To implement PCA in a high-dimension dataset, certain steps need to be followed. The first step is to find the covariance matrix. The covariance matrix represents the variance between a combination of the dimensions.

$$\text{var}(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{(n-1)}, \quad (2.1)$$

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{(n-1)}, \quad (2.2)$$

where x and y are vectors, while \bar{X} and \bar{Y} are the mean of each vector. Equation (2.2) shows the covariance between two dimensions while equation (2.3) shows the overall covariance matrix

$$C = \begin{pmatrix} \text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) \end{pmatrix}, \quad (2.3)$$

where C is the covariance matrix of three dimensions. Each column in C represents a principal component (PC). Each PC represents the correlation between all dimensions in the original dataset. To select the best PC or PCs that include the most variance between dimensions, eigenvectors and eigenvalues need to be calculated using equation (2.4)

$$C \times v = \lambda \times v \quad (2.4)$$

where v is the eigenvectors and λ is the eigenvalues. The eigenvectors with the highest eigenvalues are the most valuable PCs in the dataset. After choosing the valuable PCs, it is easy to make an observation on the correlation between features.

2.3.2 Vector Quantisation

Vector quantisation (VQ) is a well-known technique that has been used for many different applications such as data compression and pattern recognition [39-43]. Both applications include common steps such as using a codebook to generate the index vector for data compression. Codebook is used to represent one category or pattern in pattern recognition [44, 45]. The following are the steps to implement the VQ technique:

1. Data preparation: The first step in VQ is to order data to n -dimensional vector (X). For example, RGB images can be ordered in three-dimensional vector. The way to order the data and the number of dimensions needed are application-specific.
2. Codebook design: There are two main type of codebooks: local and global. The main difference between them is the source of training data that are used in the training phase [45-48]. The local codebook training data source is the targeted data. For instance, a codebook can be trained with the same image or pattern that is to be compressed or classified [44]. In this way, all the crucial features will be included in the codebook. This type of codebook can generate high-accurate results in data compression and classification [39]. However, local codebooks are considered too specific and need to be sent to the receiver side in the data compression scenario. In addition, a codebook must be generated for every dataset, which is considered a high overhead. In contrast with global codebooks, the training data is taken from several samples. This type of codebook can cover more data compression and classification cases, but the performance is significantly degraded in contrast with local codebooks.

Linde-Buzo-Gray (LBG) algorithm is the best-known algorithm for codebook design [48, 50]. A codebook initialisation is required for LBG algorithm because it is an iterative algorithm. There are three steps to implement LBG algorithm.

- The first step is to initialise [45, 47, 48, 50]: the codebook $\hat{X}_i^{(1)}$ where $i = 1, 2, 3, \dots, N_C$, distortion measure d , and frictional distortion change

threshold μ . In addition, a high number must be assigned for average distortion for all training vectors $D^{(0)}$ as an initial value and the iteration counter $L = 1$. There are three main methods for codebook initialisation: random codes, splitting and pairwise nearest neighbour (PNN) clustering. Choosing random vectors from the training data is the primary idea behind the random codes method in codebook initialisation. In the splitting method, the initial codebook contains only one codeword, which is the centroid for the entire training data. The next stage in the splitting method is to split the first codeword into two words. Therefore, the initial codebook contains two codewords. LBG algorithm will use the codebook in the first iteration and the process is repeated again until the codewords equal to N_C . In the PNN method, the initial codebook starts with N number of vectors, each vector representing a cluster. The two vectors with the closest observation are merged together to create a new codebook with $(n-1)$ clusters. This process is continuously repeated until the codebook reach N_C clusters and the codebook represents the initial codebook for LBG algorithm.

- After initiating the codebook, the second step in LBG algorithm can be started by training the codebook. In this level of codebook training, the decision region is mapped and every training vector is encoded to the nearest code vector in the map. The minimum distortion rule making both defining decision region and assigning the training codevector in the map. The common distortion measure is mean

square error (MSE), which represents the square of the Euclidean distance between two vectors as shown in equation (2.5).

$$d(X + \hat{X}) = \frac{1}{n} \sum_{i=0}^n (x_i - \hat{x}_i)^2 \quad (2.5)$$

In some applications, a weighted MSE has been used for the same purpose, as shown in equation (2.6).

$$d(X + \hat{X}) = \frac{1}{n} \sum_{i=0}^n w_i (x_i - \hat{x}_i)^2 \quad (2.6)$$

If the average distortion $D^{(L)}$ is less or equal to the threshold, the LBG algorithm achieved convergence and the codebook is ready to use for

$$\frac{D^{(L-1)} - D^{(L)}}{D^{(L-1)}} \leq \epsilon \quad (2.7)$$

VQ technique. Otherwise, the codebook update step is required.

- The final step is to update each code vector in the codebook by replacing the code vector $\hat{X}_i^{(l)}$ with a new vector $\hat{X}_i^{(l+1)}$, which minimises the quantisation error in that decision region. After updating the codebook, a new iteration will start from step two and this process will repeat itself until the algorithm reaches the convergence. The threshold is play man rules in number of iterations, but a specific number of iterations can be signed to control the processing time if the threshold has low value.

3. Comparing codebook with test data: After the codebook is trained, each vector from testing data X is assigned to one of codebook vectors \hat{X} based on

MSE and its index k will be either transmitted to the receiver side or saved in the new vector for classification purposes.

As mentioned previously, codebooks can be initiated in various ways. In addition, the codebook itself can be structured in various ways to obtain the best results [39, 40, 49]. For instance, if the codebook words N_c is high and the number of vectors in the testing data n is high too, this means $n \times N_c$ comparison process. To reduce time consumption, a tree-structured codebook (see Figure 2.4) can help in reducing the number of codewords in the codebook, as shown in equation (2.8)

$$p = \log_m N_c \quad (2.8)$$

where p is the number of levels and m is the number of branches.

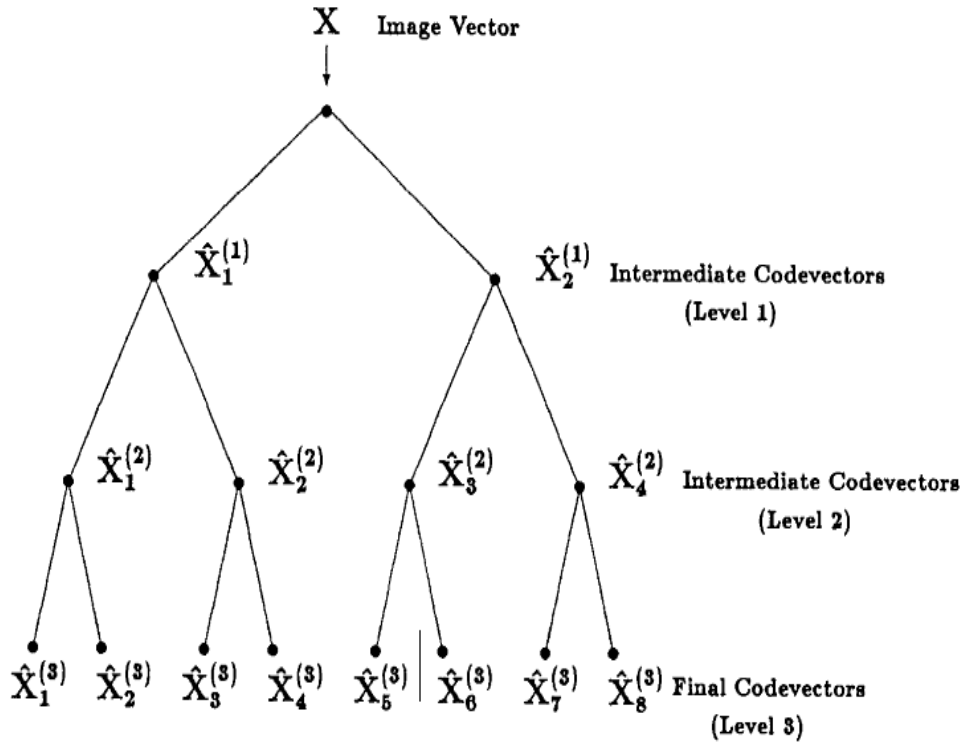


Figure 2.4: An example of binary VQ tree structure for $N_c = 8$ [33].

This codebook structure will reduce computational cost to $(nm \log_m N_c)$. However, tree structure will increase the storage cost to $n \times m \times (N_c - 1) / (m - 1)$. Classified VQ (CVQ) is another way to use codebooks. For classification purposes, each category or data set has its own codebook, but in some applications it does not produce satisfactory outcomes. The prime reason for these outcomes is that VQ index vector can provide misleading features in the training phase. Instead of using one codebook per dataset, CVQ uses more several codebooks for each dataset. Every codebook in CVQ represents a specific feature in the dataset. In this way, the comparison process will be between the features codebook instead of the main dataset codebook.

In summary, understanding the nature of the application can help to take the final decision of how to structure and design the codebook for VQ technique.

2.3.3 Hidden Markov Models

The main theory behind the hidden Markov model (HMM) is the Markov chain, which was introduced more than a century ago [50]. In the last three decades, HMM has been used as a pattern recognition algorithm in many different applications [51-55]. It is considered a powerful statistical tool for modelling generative sequences categorised by a set of observable sequences [53, 56]. HMMs consist of two primary stochastic processes. The Markov chain is the first stochastic process and is completely ‘hidden’. The second stochastic process, which is the visible part, produces the sequence of observed symbols based on a state-dependent probability distribution.

2.3.3.1 Definition of Hidden Markov Model

There are five main elements that define every HMM: state, state probabilities, transition probabilities, emission probabilities and initial probabilities. For complete HMM definition, the five elements have to be defined:

- N represents the number of states of the model

$$S = \{S_1, \dots, S_N\} \quad (2.9)$$

M represents the number of observation symbols per state

$V = \{v_1, \dots, v_M\}$. M is infinite, if the observations are continuous.

- A set of state transition probabilities $A = \{a_{ij}\}$.

$$a_{ij} = p\{q_{t+1} = j | q_t = i\}, \quad 1 \leq i, j \leq N \quad (2.10)$$

where (q_t) represents the current state. The transition probabilities should satisfy the normal stochastic constraints,

$$a_{ij} \geq 0, \quad 1 \leq j, j \leq N \quad \text{and} \quad \sum_{j=1}^N a_{ij} = 1, \quad 1 \leq i \leq N$$

- Each state probability distribution, $B = \{b_j(k)\}$ where $b_j(k)$ is the probability of v_k is emitted in state S_j .

$$b_j(k) = p\{o_t = v_k | q_t = j\}, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$$

where o_t donates the current parameter vector, and v_k is the k^{th} observation symbol in the alphabet. The following stochastic constraints must be satisfied:

$$b_j(k) \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad \text{and} \quad \sum_{k=1}^M b_j(k) = 1, \quad 1 \leq j \leq N$$

In continuous observations, we need to use a continuous probability density function, usually the probability density is approximated by a weighted sum of M Gaussian distributions N ,

$$b_j(o_t) = \sum_{m=1}^M c_{jm} N(\mu_{jm}, \Sigma_{jm}, o_t) \quad (2.11)$$

where c_{jm} = weighting coefficients, μ_{jm} = mean vectors, and Σ_{jm} = covariance matrices. c_{jm} should satisfy the stochastic constraints,

$$c_{jm} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq m \leq M \text{ and } \sum_{m=1}^M c_{jm} = 1, \quad 1 \leq j \leq N$$

- The initial state distribution for HMM $\pi = \{\pi_i\}$; however, π_i is the probability when the model is in state S_i at time $t = 0$ with

$$\pi_i = p\{q_1 = i\} \text{ and } 1 \leq i \leq N \quad (2.12)$$

It is critical to decide if the model will be discrete, continuous or a hybrid. For a discrete model, the following formula has been used in some studies [57]:

$$\lambda = (A, B, \pi) \quad (2.13)$$

The following formula is used to denote a continuous model:

$$\lambda = (A, c_{jm}, \mu_{jm}, \Sigma_{jm}, \pi) \quad (2.14)$$

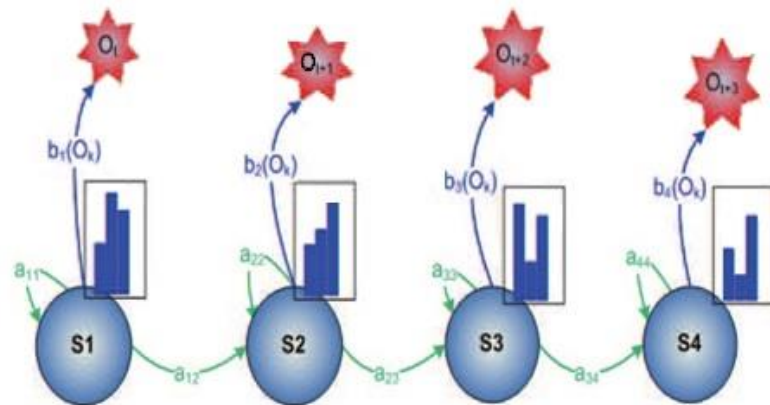


Figure 2.5: HMM example [53].

2.3.3.2 Hidden Markov Models Theory Assumptions

HMM considered one of the mathematical and computational complex theory and for this reason some assumption is made as shown below.

1. The Markov Assumption: In this assumption, the next state is dependent only on the current state and it becomes a first-order HMM. If the next state depends on previous k states, it is a k^{th} order HMM.

$$a_{i_1 i_2 \dots i_k, j} = p \{q_{t+1} = j | q_t = i_1, q_{t-1} = i_2, \dots, q_{t-k+1} = i_k\},$$

$$1 \leq i_1, i_2, \dots, i_k, j \leq N$$

2. The Stationary Assumption: It is assumed that the state transition probabilities and the time when the transition takes place are independent.

$$p\{q_{t_1+1} = j | q_{t_1} = i\} = p\{q_{t_2+1} = j | q_{t_2} = i\}, \text{ for any } t_1 \text{ and } t_2$$

3. The Output Independence Assumption: It is assumed that the present output $O = o_1, o_2, \dots, o_T$ is statistically independent of the pervious output

$$p\{O | q_1, q_2, \dots, q_T, \lambda\} = \prod_{t=1}^T p(o_t | q_t, \lambda). \quad (2.15)$$

2.3.3.3 Hidden Markov Model Types

In this section, three main types of HMM are presented. They are discrete HMM (DHMM), continuous HMM (CHMM) and semi-continuous HMM (SCHMM). Depending on the application's nature and complexity, one of these HMM types need to design a modern HMM-based system.

DHMM is used to solve the issue when continuous valued feature vectors are available and an HMM is needed. To solve this problem, three primary steps are needed:

1. Use dimension reduction methods such as VQ to reduce a set of d -dimensional real valued vectors to k -dimensional vectors.
2. Select the closest codeword vector for the current feature vector.
3. Using the index vector that is produced by VQ as an input to DHMM for training to generate the final HMM.

CHMM is a three-phase stochastic process. The first phase is equal to the first and second steps in DHMM, which represented selection of the next state. The second and third phases have the same title in DHMM, which is the training phase, but in a different way. The second phase in CHMM is the selection of the mixture density by mixture coefficient, and the third phase includes the selection of the output vector by the Gaussian density.

SCHMM represents the comparison between DHMM and CHMM. Comparing with CHMM, Gaussian mixture densities are used for all states in SCHMM. Baum-Welch/Viterbi algorithms have been used in all previous HMMs for training purposes including some modification in CHMM and SCHMM [57, 58]. HMM complexity, which causes an increase in computational expense, is a crucial factor in choosing HMM type.

It considered bias to assume there is a previous knowledge of HMMs [59, 60]. For that reason Baum-Welch (BW) algorithm used to estimate the HMM hidden parameters [58] from training database. The main objective of BW algorithm is using training data to compute maximum likelihood estimates and posterior mode

estimates [59, 61]. After obtaining maximum likelihood, a comparison between likelihoods that obtained from the training and testing data can be made to find the correct pattern or class.

2.4 Previous Works on Classification Methods

In the last three decades, many studies have reached a high level in classifying different types of databases. For instance, in image processing, different applications have been invented to help in such matters as crisis time, face recognition and object classification [7, 11]. In sound recognition, many applications in mobile phones and computers are used to detect the user's identity or to detect instructions or orders. In addition, many other researchers have used classification algorithms for understanding and improving knowledge about different types of datasets that come from specific applications. Nevertheless, the performance of classification algorithms can be diverse from one application to another based on variance in the data itself and the variance between each category or pattern. This section presents an overview of previous works on different pattern classification algorithms such as PCA, VQ, and HMM. Because our building collapse pattern classification research is a novel scenario for these algorithms, different types of implementation for these algorithms are described.

2.4.1 Principal Component Analysis Implementations

PCA is one of the widely used common pattern recognition algorithms [62]. PCA is used to reduce dimensionality, to extract features, to find correlations between

variables and pattern [62-68]. Ashish *et al.* [63] addressed dimension reduction in integrated circuits (ICs) classification using the PCA algorithm. A database of ICs with high dimensionality from Texas Instruments for a high volume device manufactured was used. Part of the database was used as training data and the rest used as test data. To classify ICs based on whether they were qualified or unqualified, the first PC was used to generate a score for each IC. Both the PC and score vector were used to generate clusters in the training phase. In the testing phase, each IC went first through a score generating, and then the score vector was compared with the centroids of different clusters. The ICs' cluster is the cluster with the closest match. Increased process efficiency and reduction of computing time are the main achievements of this work.

Another example of extracting features and finding the correlations between features is presented in [64, 65]. Wu and Huang [64] addressed groundwater contamination patterns using PCA. By finding patterns for each contamination type, the authors aimed to allocate different areas in the study area. Variance of 82.4 % was represented by four PCs only. These four PCs were used for clustering assignment. Subtracting the data from mean and dividing by the standard deviation of each column was the data preparation stage. For data classification, a Euclidean distance was used to find a measure of similarity between clusters. The combination of PCA information and clustering information outlined the spatial allocation of groundwater contamination within the study area. This research opened up an opportunity to understand and identify the source of the contamination. Continually with previous work, Wu and Huang [65] used PCA to find a correlation between groundwater contamination sources for identification and tracking groundwater contamination. By

using the most valuable PC, the authors were able to discover the linear correlation between potential sources of groundwater contamination and the route path in the study area.

In both of these examples, groundwater contamination and IC classifying, only good results were mentioned and no consideration was given to the confusing or unclassified cases. PCA is a lossy dimension reduction. This means that patterns that have low variance between them can be a challenging case for PCA. An example of such cases is represented in [66]. Xiong *et al.* [66] addressed early diagnosis of breast cancer using data mining methods. One of the data mining methods is PCA. The authors used PCA to classify breast cancer based on a fine needle aspiration (FNA) database. Using the first and the second PCs, PCA showed clear clusters representing infected or not infected, as shown in Figure 2.6.

Despite the good results, there were a few cases in which two clusters overlapped, and thus generated misclassifications. The authors combined PCA with partial least

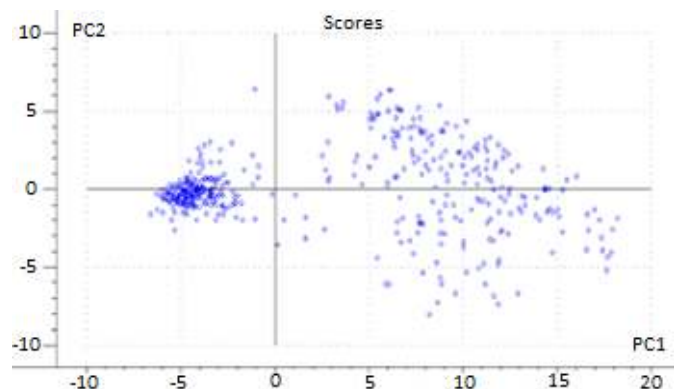


Figure 2.6: Using first and second PC breast cancer data clustering [67].

squares (PLS) to achieve more accurate and reliable results. This is a good example of how PCA is not always highly accurate and can miss classifying some patterns

based on the type of object or data needing to be classified. Another contrast for PCA has been found in image processing. Qiu *et al.* and YunBing *et al.* [62, 68] respectively came to completely different conclusions in regard to PCA and 2D-PCA performance and process time. Both authors used images, but for different applications. Qiu *et al.* [62] focused on comparing PCA, 2D-PCA and two-stage 2D-PCA performances on SAR radar images. Instead of changing the images to vectors to be compatible with PCA, the authors used 2D-PCA to extract the column and row features. Despite PCA extracting only one-dimension features, the results show that PCA can reach the same accuracy as the 2D-PCA in less time. This means that extracting more features increases the process time without enhancing the accuracy.

On the other hand, YunBing *et al.* [68] addressed a comparison between PCA and 2D-PCA performance in wood identification. The authors claim that 2D-PCA is faster and more accurate than PCA. They used PCA and 2D-PCA to extract features from wood images and classified them into categories. Images that had similar features were in the same category. The final results proved that 2D-PCA is better than PCA in wood classification. To classify wood based on images, many details need to be extracted from the images, which were taken from a horizontal cut of the tree. This means that using both dimension features were useful in this case.

In the radar image scenario, three objects needed to be classified. In this case, the level of complexity was lower compared with the wood scenario and one-dimension features were more than enough for classifying the object. Using features from two dimensions did not add any accuracy enhancement.

The above studies indicate that PCA can be time efficient, but the output accuracy cannot be guaranteed, especially when using PCA with a new dataset that came completely from a new application.

2.4.2 Vector Quantisation Implementations

The flexibility of designing a VQ codebook and the capability of fitting different types of data structures in the codebooks enables the VQ technique to be used extensively in different types of application such as image assessment, face recognition, motion detection and human body action recognition [21, 49, 69, 70].

Cui *et al.* [69] addressed the vector quantisation histogram (VQH) method for image quality assessment. Image quality assessment is based on the difference between a distorted image and reference image features. Different methods have been used to assess the images, including MSE and peak signal to noise ratio (PSNR). However, these methods do not reflect what the human visual system can observe. The authors used the summation of absolute difference between the projection histogram from the reference image using VQ and the histogram of distorted image at the receiver side. This method gave better results than five different distortion types such as MSE, PSNR and structural similarity (SSIM). In addition, sending a reference image histogram does not require much bandwidth because it is a small amount of data.

In [21], Kotani *et al.* focused on using image histograms that were generated using a VQ codebook for face recognition purposes (see Figure 2.7). A low pass filter was used to avoid the high-frequency noise and focus more on low-frequency face

features. The second step was to divide the images into blocks and then use VQ to generate the image histogram. The Manhattan distance method was used to find matching measurements between the histograms. This simple method achieved high-accuracy results (95.6 %) and required low computational power.

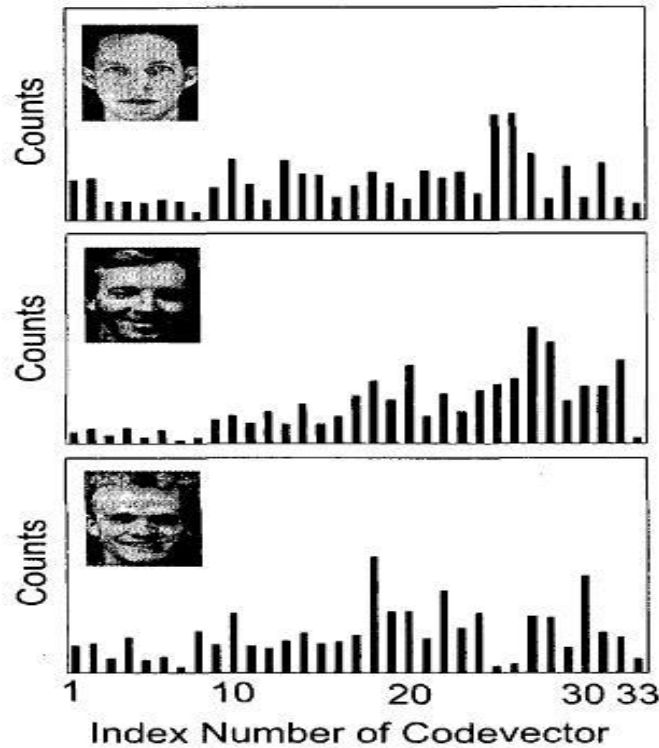


Figure 2.7: Typical examples of histograms [21].

In the area of motion detection, Hao and Shibata [49] addressed ego-motion detection algorithms using VQ techniques. In ego-motion detection, the prime goal is to detect the observer motion and a determination of global motion is needed for this purpose. In contrast, normal motion detection focuses on recognising a specific object's motion. The authors studied four types of motion: vertical, horizontal, zooming and rotation. Each motion type was summarised in a vector that represented the edge movements of objects in each frame. The VQ technique used these vectors for edge histogram projection. These histograms were used as patterns to detect the

observer motion using Manhattan distance. The overall results reached over 93% and had low computational expense.

Feng and Perona [70] addressed human body action recognition in a sequence of images using VQ and HMM. A human body was partitioned into numbers on regions of interest (ROI). The movements of the ROI were then interpreted to numbers of movelet codewords. A movelet represents the RIO features of the main parts of the body. These features are shape, motion and position. VQ was used to put these movelets into one codebook, generating a sequence of numbers that represented the action in the tested image. HMM interoperated the outcome from VQ to meaningful action. The authors used three periodic and eight no periodic actions imaged as testing actions. In the overall results, accuracy of 80% was achieved in this work.

In [21, 49, 69], the authors used VQH in different ways for different purposes. In [21, 69], the authors used the original images as a source for a VQ codebook, but for comparing the same image in [69] and comparing with different images in [21]. In [49], the authors used the features of the original data to create the codebook. Conversely, the VQ codebook contained the wide range of human body actions, which were represented as a global codebook that detected the human body's movement, without understanding what these sequences of movement meant.

To summarise, VQ is a technique that can combine two important features: simplicity and efficiency. These two features make it worth using in many new implementations, such as this research. In addition, its outcome can be easily used as an input for other classification methods.

2.4.3 Hidden Markov Model Implementations

HMM has been extensively and successfully used in different research areas that aim to classify patterns based on creating models for each category. For instance, HMM has been used in speech, image and motion recognition [51, 52, 71-74].

In the speech recognition area, Abushariah *et al.* [51] used acoustic signals to recognise English decimal numbers in various scenarios using HMM. The input signal digitising and Mel Frequency Cepstral Coefficients (MFCCs) algorithm was used to generate the feature vector. HMM was used to classify and categorise each frame in the feature vector. Until this level, HMM was used to generate a pattern for each English decimal number. The Viterbi algorithm was used as a decision maker depending on the maximum likelihood. Experimental results fluctuated between 56.25 and 99.5%, depending on the experimental environment.

In the same research area, Foo *et al.* [71] focused on lip reading using HMM. With some modifications and improvements to the Baum-Welch training algorithm, the researchers obtained a lower error rate in English letter recognition when compared with a basic HMM algorithm.

Moving to human motion recognition, this research area is considered one of the closest to our research because it uses sensory data motion recognition. Hara *et al.* [73] focused on detecting unusual human behaviour using HMM. Sensors were installed in different types of accommodations, such as houses and apartments, to test the model. VQ was used as a dimension reduction algorithm to reduce the number of sensors. A sensor network recorded different types of human activity to create a

robust human behaviour model. The likelihood of a user action or the distance between the state transition probabilities were used to detect the unusual behaviour. The authors claimed that this research proved that unusual human behaviour without previous knowledge could be detected by using HMM.

More specifically in human motion recognition, Wan *et al.* [52] addressed human motion real time recognition. Motion feature vectors were collected from a micro inertial measurement unit (μ IMU). A combination of VQ and DHMM was used instead of PCA and CHMM because of its low computational cost, which reached less than 350 ms. VQ was used as a dimension reduction method, and HMM as a classifier for five different categories that were composed of 10 different typical human motions. Sliding window algorithm was used for real time detection purposes. The recognition rate was between 95 and 100 %. Continuing on with this work, the same authors enhanced the classification phase by using a hybrid classifier composed of HMM and a support vector machine (SVM) [74]. The upgraded classifier had a 99% precise recognition, which was more robust than in [52].

To summarise, HMM has been implemented on different types of applications in different methodologies. VQ and PCA have been the most common methods used for the data preparation phase, and the Baum-Welch algorithm has been used in many studies in the training phase [57-61]. HMM was introduced as a robust algorithm that could classify different types of data composed of different categories. HMM features have strongly encouraged researchers who work on this research to consider using it.

2.5 Chapter Summary

In this chapter, we presented the disaster aid research area and the current methods used in gathering building status in the stricken-area mostly by remote sensing images. The SHM system research area shows the current methodology in using WSN in term of building monitoring prior to collapse. In addition, a theoretical background on the three classification methods used in this research work have been presented. Different types of classification algorithms were presented overall algorithms performance under different implementations. This overall performance gave a general idea on how classification algorithms will perform in our novel application.

Chapter 3: Building Collapse Patterns and Simulation

This chapter provides an overview of how building collapse patterns were simulated based on information driven from disaster aid research institutions reports. It describes how research teams collect and classify information in collapsed buildings reconnaissance reports. These survey reports show examples of classification types and the methods used in the classification process, and this information is presented in the introduction (see Section 3.1). Section 3.2 presents various kinds of building collapse patterns, which are included in reconnaissance reports taken from various cities around the world. Section 3.3, presents how building collapse patterns were simulated using the Blender game engine. In addition, an illustration of how Wireless Sensor Networks were simulated inside each building and how the data gathering process operated. Section 3.4 provides a summary of Chapter 3.

3.1 Introduction

Disaster research institutions like the Multidisciplinary Centre for Earthquake Engineering Research (MCEER) and National Centre for Earthquake Engineering Research (NCEER) are interested in studying the effect of natural disasters on houses, buildings, infrastructure, human health conditions and crisis aid. Therefore, after each natural disaster and as soon as it secure, emergency services in cooperation with research institutions begin sending their teams, consisting of rescue workers, researchers and experts, to assess and classify the level of damage caused by these

disasters. These teams normally write a reconnaissance report that contains the level and type of damage in the stricken area with statistics on casualties, injuries, rescues and damaged and undamaged building and infrastructure. These statistics are obtained from observations and the cooperation of police stations and hospitals [17]. Complete reconnaissance reports take approximately three to four months to prepare before they are ready for publication [17], and they are considered useful for civil engineers and crisis aid researchers. However, immediate reconnaissance reports can be used one or two days after the crisis for rescue planning operations [17]. Remote sensing images are used in some cases to save time (see Figure 3.1) [75]. Reconnaissance reporting teams face many challenges during the few days after a crisis, including difficulties with electricity, communication and transportation [17].

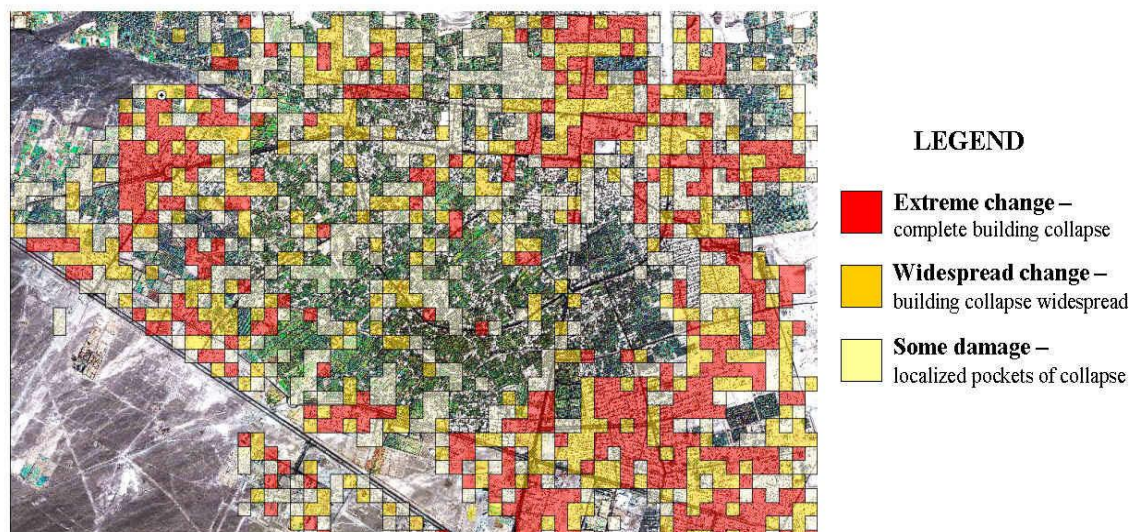


Figure 3.1: Remote sensing image used for building collapse detection by a research institution in Iran during the Bam earthquake in 2003[75].

The main contribution of reconnaissance reports in this research is the surveys on reinforcement concrete building collapse patterns and the causes behind these collapses. In addition, these reports show when the collapsed building was built and

whether it was built based on the country-specific Uniform Building Code (UBC). For example, the reconnaissance report on the 1995 earthquake in Japan shows that buildings built on UBCs current at the time did not show any damage after the earthquake, and most of the collapsed buildings were built on ten year old UBCs [17, 76]. Conversely, during the 1999 earthquake in Taiwan, 24 modern 10- to 15-storey apartment buildings collapsed [77]. These buildings were built based on UBC United States (US) standards from that period. When buildings that were built on the most recent UBCs collapse during a natural disaster, it highlights the possibility that current UBCs may need updating. In most cases, based on reconnaissance reports and civil engineer's reports, UBCs are changed and updated to face new challenges. This means that the existing buildings could still collapse if they are faced with natural disasters that they are not designed to withstand.

The majority of collapses caused by natural disasters are due to either poor design or poor materials [75, 78]. Unfortunately, people in many developing countries, such as Iran [75] and Sumatra [78], are not able to build many buildings based on the latest UBCs, which increases the statistics of casualties and collapsed buildings.

3.2 Building Collapse Pattern Types

The primary objective of this section is to describe the building collapse patterns that have been used as categories in this research. However, it is necessary to provide an explanation on how earthquakes cause building collapse before discussing collapse patterns.

Buildings lose their stability in earthquakes due to lateral movement of the structure [79]. Specifically, the most destructive load effect on a structure is horizontal oscillating movement, which shakes the foundation. The lateral movement can significantly change the basic shape of a structure, which leads to a new shape that has much less capability of carrying the building mass. In addition to this, there is a vertical resistance force generated by the building mass and gravity, which is trying to keep the building in place (see Figure 3.2). As a result, a structure keeps seeking stability by changing shape under the horizontal and vertical forces. If there is no damage to the structure that causes structural failure, then the building will survive. However, if this type of damage is prolonged, then a partial or complete collapse may occur [79].

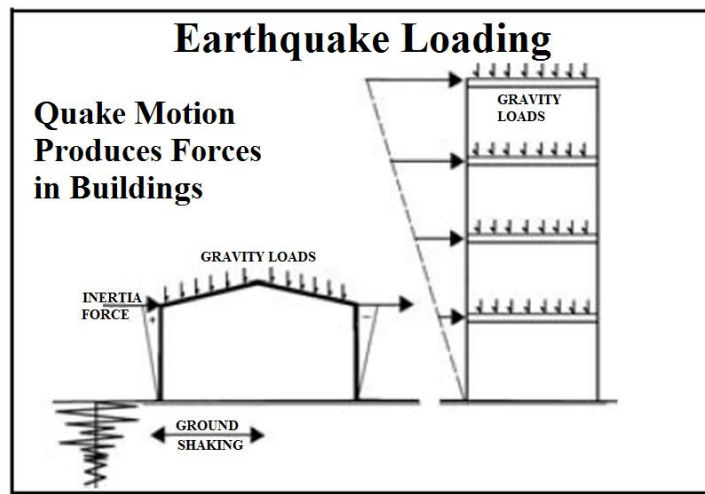


Figure 3.2: Shows the horizontal and vertical forces that can cause building collapse [79].

Different types of building collapse patterns are observed during natural disasters around the world [17, 75–78]. This thesis will cover first column (FC), first-storey (FS), mid-storey (MS) and pancake (PCK) collapse patterns. An explanation on how these collapse patterns occur is presented in the following sections.

3.2.1 First Column Collapse Pattern

Another name for the FC collapse pattern is local column collapse. It is a type of progressive collapse that occurs in part of the building and is caused by loss of stability (see Figure 3.3) [79]. This type of building collapse can be caused by one or a combination of the following failures:

- Inadequate shear strength failure
- Inadequate beam/column joint strength failure
- Tension/compression failure.



Figure 3.3: On the left, Ronan Point building local column collapse in 1968 [63]. On the right, a sketch of local column collapse [79].

The majority of loss of life occurs in the collapsed column, and the percentage of survivals is significantly high in the rest of the building [79]. Opening a new exit could be part of the rescue operation if the collapse occurs at the entrance [79].

3.2.2 First-storey Collapse Pattern

This FS type of collapse is also called a soft, first-storey collapse pattern [75–79], as it occurs in buildings that have first storeys with significantly less stiffness than the other storeys [79]. For example, there could be fewer or no walls in the collapsed storey. This type of collapse occurs in commercial or parking buildings (see Figure 3.4) [79]. This type of collapse can be caused by the following:

- Inadequate shear strength failure
- Inadequate beam/column joint strength failure.



Figure 3.4: First-storey collapse of residential building with shops in Koriyama city Japan after 2011 earthquake [17].

Almost all victims of the collapse will be found within the first storey, and survivors will be found in the second storey and above [79]. The rescue operation should be carried out from the second storey in order to access the collapsed storey.

3.2.3 Mid-Storey Collapse Pattern

The MS collapse is another type of soft-storey collapse (see Figure 3.5). This type of collapse can occur for the following reasons:

- No walls exist in the collapsed storey, while a significant amount of walls exist in the above and below storeys.
- The collapsed storey has shorter columns than the storeys above and below it.



Figure 3.5: Sixth-storey collapse in the eight-storey Kobe city hall building in the 1995 Japan earthquake [81].

The survival rate for the above and below storeys is high, and the death rate will be high in the collapsed storey [79]. In order to rescue the victims in the collapsed storey, an operation should be carried out from the above storey after securing access into the below storey [79].

3.2.4 Pancake Collapse Pattern

The PCK collapse pattern is also referred to as the catastrophic collapse pattern (see Figure 3.6). It can occur due to the same factors that cause the FS and MS collapse

patterns. The survival rate is very low and rescue dogs can be very useful in this type of collapse [79].



Figure 3.6: A collapsed four-storey building in the Turkey earthquake [82].

3.3 Building Collapse Pattern Simulations

Experimental environments can be represented in various ways, such as in the field, in laboratories or by computer simulation. The best environment is in the field; however, experiments in this area of research cannot be implemented in the field unless there is earthquake. A laboratory earthquake simulator was unavailable. In simulating building collapses, we chose the Blender game engine, which is open source software, for its capability and flexibility in simulating buildings and the environment that surrounds them [83-86]. We chose not to use civil engineering simulation programs (such as Strand, Staad, and ANSYS) in order to avoid material calculation complexity, which is not within the scope of this research. In addition, most civil engineering simulations do not prepare for complete physical building collapse; they are used to predict building collapse probability based on vibration, strength and material type. The Blender physics game engine, has been used in

previous research studies and shows the capability of simulating different environments, such as indoor robot vision [87, 88], and simulating a complete outdoor environment for simulated robots [89]. Despite Blender being designed for animation and not for simulations, researchers were generally satisfied with this software [87–89]. However, researchers complained about the rendering quality on corners but agreed it does not affect the experimental results [89]. Blender provides a wide space for simulating various collapse patterns. In particular, blender’s ability to track a block through out the entire collapse is crucial to the study of collapse patterns. Moreover, using the Python programming language with Blender gives an opportunity to observe the building collapse behaviour during an earthquake’s simulated force using simulated WSNs. This flexibility helped implement different types of collapse patterns. The four building collapse patterns that were presented in section 3.2 were simulated using the Blender game engine. The following sections will discuss the details.

3.3.1 Building Dimensions

Our prototype represents a very basic building with three storeys and four columns. Building dimensions were set to reduce the complexity and increase the building generality. As shown in Figure 3.7, the building dimensions are 20×5×12 m. The life and death load was included by considering materials used in our simulation. The main building structure, including the building columns and roof for each storey, is constructed from concrete, which weighs 2500 kn/m² [90] for a death load and 100 kn/m² for a life load in an office building [90]. The walls are constructed from masonry and are 26×13×10 cm in dimension and a weight equal to one kilogram.

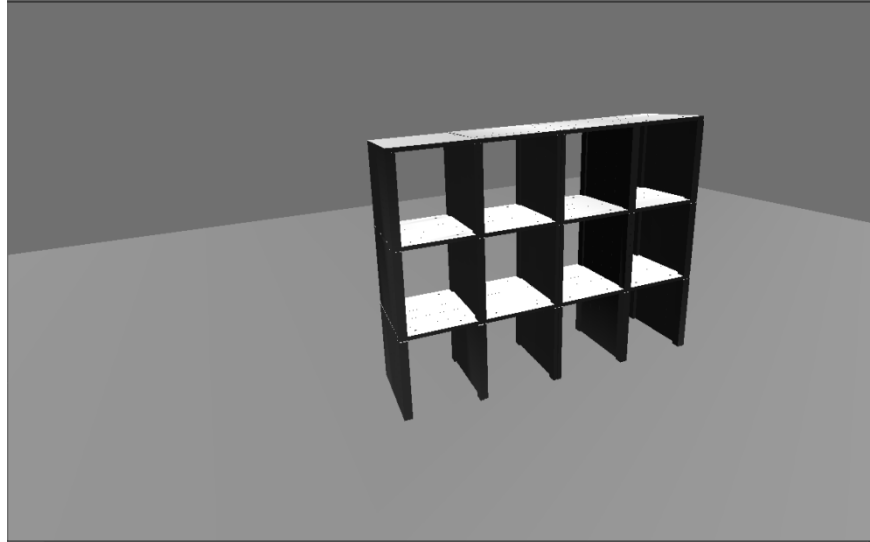


Figure 3.7: The basic building prototype that was used in this research.

3.3.2 Earthquake Simulation

To design an anti-seismic structural building, the horizontal forces that are equivalent to an earthquake must be calculated. We used the same methodology that is used in seismic engineering to find the equivalent horizontal forces and then implemented these in our simulation. The following steps show how we calculated the horizontal forces:

1. The most important storey of the building needs be determined. This estimation is based on the designer's point-of-view and the main use of the building [see Appendix A]. In the prototype, we determined that the second storey was the most important because the building was an office building.
2. Finding the probability factor (K_p) and hazard factor (Z) is necessary to find the estimated work life of the building. The hazard factor is based on the Sydney CBD area. We estimated that the average work life was equal to 50 years. Based on this work life, we found the following:
 - Annual probability of exceedance = $1/500$ [see Appendix A]

- $K_p = 1.0$ [Table 3.1; AS 1170.4-2007] [see Appendix A]
- $Z = 0.08$ [Table 3.2; AS 1170.4-2007] [see Appendix A]

3. The site's sub-soil class has an effect on the equivalent earthquake force.

There are many types of sub-soil class, such as strong rock, rock, shallow soil, deep or soft soil and very soft soil. The site's sub-soil class is determined by estimating the soil profile of the ground under the building. Figure 3.8 shows the estimated soil profile for our simulation. The next step is calculating the travelling time (T) based on the soil profile, which is shown below [see Appendix A].

$$T = \sum \left(\frac{\text{Layer thickness}}{\text{Max. depth of soil}} \right) \times \text{ratio} \quad (\text{Equation 1})$$

where $\text{ratio} = 0.6$

$$T = \left(\frac{10}{40} \right) \times 0.6 + \left(\frac{10}{45} \right) \times 0.6 + \left(\frac{10}{55} \right) \times 0.6 = 0.39 \text{ sec} \quad (\text{Equation 2})$$

$T < 0.6$, so site sub-soil class is shallow soil (C_e)

4. From the calculations so far, we can determine the earthquake design category (EDC). Using the site's sub-soil class, building height, Z and K_p [see Appendix A], the EDC type is calculated as equal to II. This type means that earthquakes have a critical effect on buildings in the area.
5. The last step is to calculate the horizontal equivalent static forces on each storey. V in equation 3 represents the static force [73].

$$V = [K_p Z C_h(T_1) S_p / \mu] W_t \quad (\text{Equation 3})$$

Where $C_h(T_1)$ = value of the spectral shape factor for the fundamental natural period of the structure.

W_t = seismic weight of the structure taken as the sum of W_i for all storeys.

S_p = structural performance factor.

μ = structural ductility factor.

By implementing equation 3, the equivalent forces were as follows:

- First-storey force = 144.8 N
- Second-storey force = 332.7 N
- Third-storey force = 541.2 N

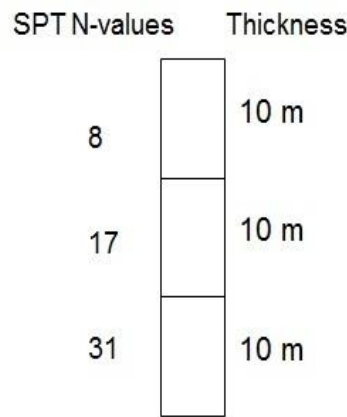


Figure 3.8: Soil profile for the simulated building [90].

3.3.3 Building Collapse Implementation

Sensors and controls are used in Blender to implement the earthquake forces. The sensors (which are not the WSNs in the building; they are controllers in the Blender game engine) and controls are used to execute the Python code that implements forces. Python and Blender integration helps to make the simulation more stable and reliable in generating desirable forces. Based on the building collapse type, part of the building is joined together so it will not be affected by the earthquake forces. For example, in an FC collapse, three columns were joined together to prevent collapse in those columns. Figure 3.9 shows an example of the four collapse types that have

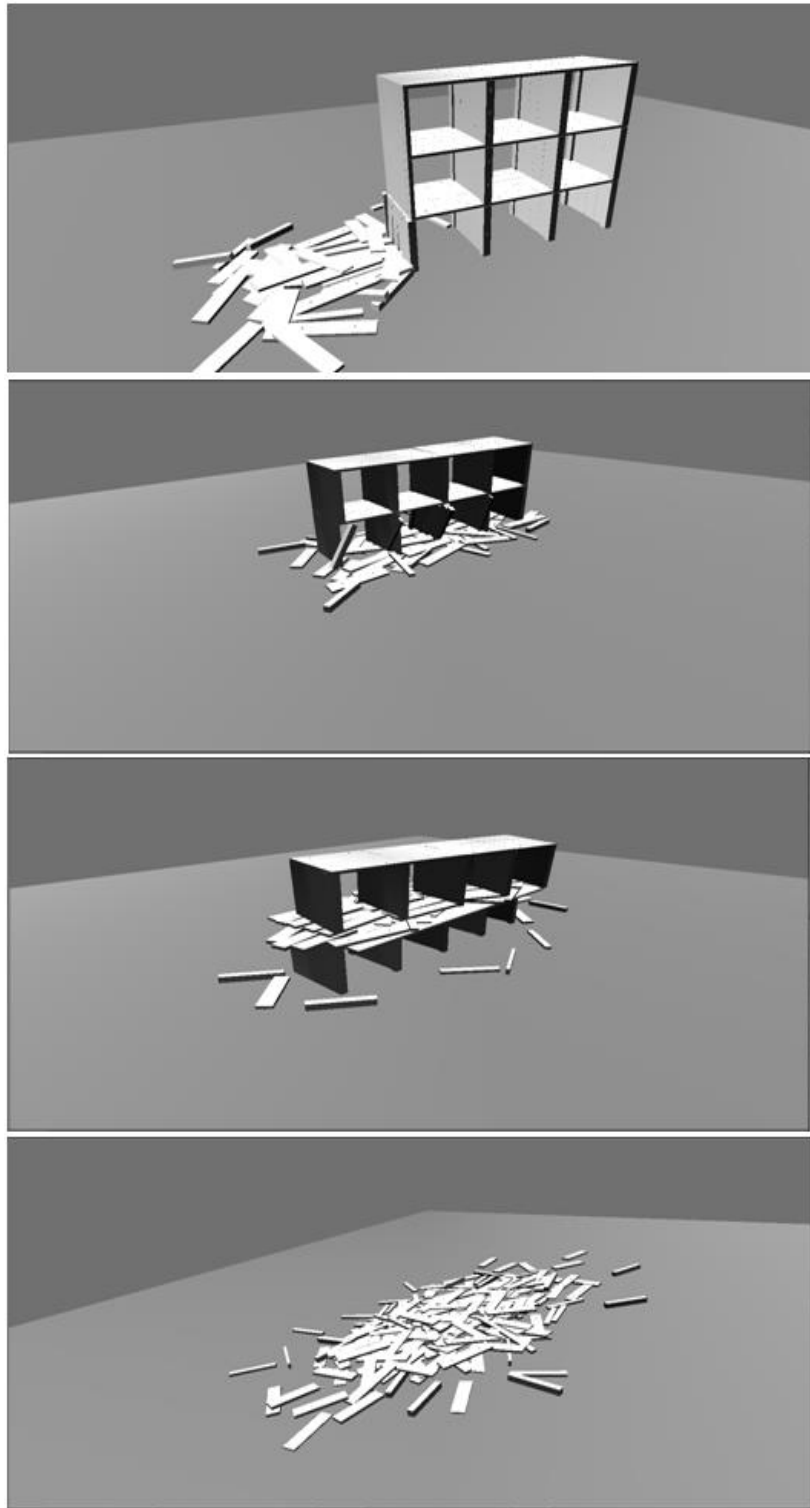


Figure 3.9: Building collapse patterns. From top to bottom: first column (FC), first-storey (FS), mid-storey (MS) and pancake (PCK) collapse patterns.

been investigated and presented in this thesis. This is because the main purpose is to generate many finger prints for each type of collapse.

3.3.4 Wireless Sensor Networks (WSNs) Simulation

As shown in Figure 3.7, the building prototype has 12 blocks. Each block has four wireless sensors installed in the block roof (see Figure 3.10). Therefore each building consists of 48 sensors. The sensor itself is a blank cube in Blender and is not functional. However, each sensor (cube) is linked with a Python program [see Appendix C] that can record sensor velocity in three directions in real time, using Blender controllers. The Python program will record the sensor's velocity during and after building collapse and save the record in .csv files for post-processing in MATLAB [see Appendix C].

It is worth to mention that in early experiments of this work, acceleration was used instead of velocity. We used sensors acceleration based on simulating real world acceleration sensors as an input data to classification algorithms. However, sensors acceleration database created major confusion for classification algorithms because of the similarity between unmoved sensors and moving sensors. In unmoved sensors, simulated earthquake forces cause a small amount of shaking in the embedded sensors that leads to a change in acceleration. These small changes in acceleration matched moved sensors in two scenarios. The first scenario appears when the building collapse starts and sensors start falling. In this window of time, moving sensors record an acceleration similar to the unmoved sensors during the earthquake vibration. The second scenario appears when moving sensors travel with high speed and slight changes in velocity. This kind of similarity either shows moving sensors as

unmoved or vice versa. Therefore, investigating velocity signatures instead of acceleration signatures removed all disadvantages above (see the outcome of chapter 4 and 5).

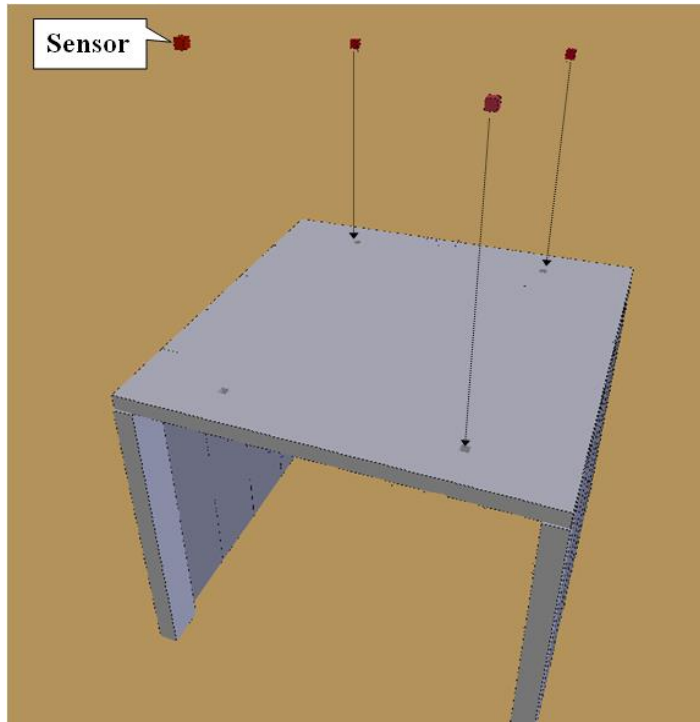


Figure 3.10: Illustrates the position of wireless sensors in each block.

3.4 Training and Testing Databases

Five building patterns were investigated: four collapse patterns plus one pattern for uncollapsed building. For training purposes, we generated 15 datasets for each pattern, while 50 datasets were generated for each pattern to use in the testing phase. In total, 75 datasets for training phase and 250 datasets for testing phase were generated. Data was normalised with respect to number of samples to make all the pattern datasets even. The prime purpose behind generating many datasets is to train

and test classification algorithms with different collapse types to determine feasibility.

3.5 Summary

A novel method to capture building collapse motion is presented in this chapter. The Blender game engine, Python code and MATLAB were used to create a database for each collapse pattern that reflected building behaviour during the collapse. Collapse-pattern databases are beneficial for classification purposes. This research aims to help rescue teams by providing information that can create a primary report on buildings condition after the earthquake and a pattern of building collapse.

Chapter 4: Experimental Results

In this chapter, we present the experimental results of the building collapse pattern classification algorithms. We highlight the important trends that helped to assemble the research process into a clear picture, which in turn answer the research questions regarding the efficiency of the proposed pattern identification algorithms for situation awareness in disaster events. The goal of each of the classification algorithms presented here is to classify collapsed buildings under five categories: first column (FC) collapse, first storey (FS) collapse, mid-storey (MS) collapse, pancake (PCK) collapse and uncollapsed (UC). Section 4.1 presents an overview of sensor behaviour during building collapse. In Section 4.2, a general experimental setup for all classification algorithms is presented. Each algorithm has its own additional experimental setup, presented in the same algorithm section. Section 4.3 introduces the first classification algorithm, PCA; while Sections 4.4 and 4.5 present the VQ and HMM techniques respectively. A hybrid algorithm is presented in Section 4.6. Results, statistical analysis and interpretation are discussed in Section 4.7.

4.1 Sensor Behaviour During Building Collapse

As mentioned in Chapter 3, each simulation is based on a building equipped with 48 sensors, distributed over four columns in three storeys. In this section, we outline sensor behaviour during building collapse. As shown in Table 4.1 (see Appendix D), four different sensor readings were taken from various positions in various collapse

patterns. Each sensor (in Table 4.1 (Appendix D)) reflects the sensor behaviour fallen from a specific storey. We classify sensor behaviour in four categories (see Table 4.1 (Appendix D)).

The first three categories are first-, second- and third-storey sensors. Three main variables diagnose sensors in these three categories:

- Sensor maximum fall velocity
- Travelling time

In regards to fall velocity, as expected, sensors that fell from the third storey to the first storey recorded higher velocities than did sensors that fell only one or two storeys. For instance, third-storey sensors in FC and PCK had a higher velocity than did sensors in any other collapse pattern.

On the topic of travelling time, some sensors fell directly from the third or second storey to the ground, without hitting any objects on the way. Conversely, other sensors took more time to reach the ground because they hit objects during their journey. For instance, a sensor located on the third storey of an FC collapse pattern that hit an object had a slower rate of fall than did a sensor falling from the same position without interference.

The last category is unmoved sensors. There are many similarities between FC and MS sensors. The vibration caused by an earthquake's equivalent horizontal force and building collapse could cause a small movement along three axes and make sensor recordings differ slightly. However, we believe that the fourth category of sensors can record more difficult vibrations in reality because all buildings shake during earthquakes. However, this is not the case with UC patterns. UC sensors should not

record any major vibration during normal operation when there are no major earthquakes. Moreover, buildings with three storeys did not record vibrations from wind force as this is beyond our research scope; although, because we used a game engine as a simulator, some vibrations did occur during simulation.

In summary, the time a sensor takes to reach the ground, the vertical velocity, and the number of fallen sensors can play a major part in pattern recognition algorithms.

4.2 The Experimental Results using PCA Algorithm

In this section, we present the PCA algorithm in three ways: basic PCA, two-level PCA and sensor-by-sensor based PCA. The main difference between these methods lies in their manner of feeding the PCA algorithm with data, so that each method gathers various features.

4.2.1 Basic PCA

After preparing the training data, the organised dataset containing 75 patterns was input to the PCA algorithm to extract the most crucial features as represented by the principal components (see Figure 4.1). We chose 19 principal components from

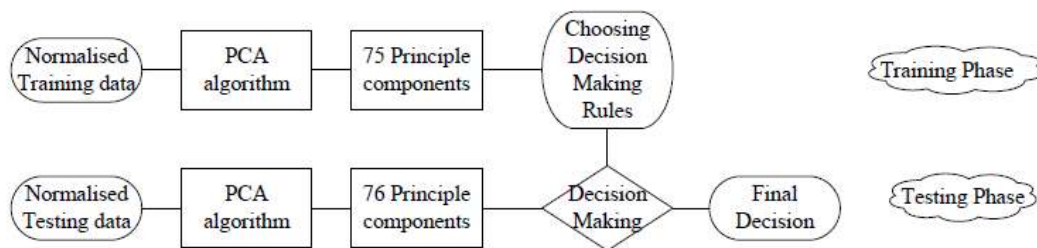


Figure 4.1: Process steps of basic PCA.

overall 75 principal components, representing a 90% data variance. We could have chosen 33 principal components to achieve a 98% variance. However, the additional 14 principal components did not add any useful information in regard to creating a trend for each collapse pattern.

The decision-making process was strongly dependent on how the principal components translated sensor readings into features able to distinguish between each category. Below are the decision rules driven from the 19 principal components:

If principal component 1 is negative, **then** the pattern is UC.

If principal component 2 is bigger than 0.1, principal component 3 is negative, and principal component 4 is negative, **then** the pattern is MS.

If principal component 3 is bigger **than** 0.16, then the pattern is FC.

The PCA algorithm could not provide a clear boundary between FS and PCK. For this reason, we built Table 4.2, and combined rules to assist in the final decision between FS and PCK collapse patterns. A pattern should achieve 60% or higher score in Table 4.2 to classify either FS or PCK collapse pattern. Otherwise, the pattern is considered unclassified.

Table 4.2: Decision rules for FS and PCK collapse patterns.

Principle components	FS weights		PCK weights	
	+	-	+	-
PC5	1	14	7	8
PC10	11	4	6	9
PC11	5	10	12	3

During testing phase, unknown building collapse patterns added to the 75 training data for classification purposes, based on decision rules. Figure 4.1 shows the process undertaken, while Table 4.3 details the classification results.

It is clear from (Table 4.3) that the combined rules in Table 4.2 are biased toward FS. As all sensors in both FS and PCK are moving during building collapse, the PCA algorithm could not distinguish features between them. For this reason, the algorithm failed to classify PCK accurately (42%).

Table 4.3: Basic PCA results and Confusion matrix.

Basic PCA results							
				PCA (theoretical)			
Overall classification rate				82.4%			
First column (FC)				98%			
First storey (FS)				88%			
Mid-storey (MS)				86%			
Pancake (PCK)				42%			
Uncollapsed (UC)				98%			
Processing time in seconds				191/ 250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	49	1	0	0	0	0	50
FS	0	44	0	4	0	2	50
MS	0	4	43	3	0	0	50
PCK	0	29	0	21	0	0	50
UC	0	1	0	0	49	0	50

With MS, seven patterns were misclassified from this category. To explain this situation, we reviewed the decision rule for MS. The combination of three principal components indicated that the PCA algorithm could not find specific features between MS patterns and other collapse patterns. These results indicate that more

sensors moving during the building collapse results in more confusion for the basic PCA algorithm.

4.2.2 Two-level PCA

To improve the outcome of the basic PCA algorithm, a two-level PCA algorithm was implemented. In this two-level PCA, the first level is similar to the basic PCA without FS and PCK collapse pattern decision rules. In other words, the first level classifies FC, MS and UC collapse patterns only. If the unknown collapse pattern is not one of the three collapse patterns at the first level, the second level takes place by implementing PCA with FS and PCK collapse pattern training data, extracting features that classify each collapse pattern (see Figure 4.2). Fourteen principal components were required to achieve 90% from the total 30 principal components. However, only the first principal component was needed to distinguish between FS and PCK collapse patterns, as shown below:

If principal component 1 is bigger than 0.15, **then** the pattern is PCK.
Otherwise, the pattern is FS.

It is important to mention that the previous rule covered 99.5% of all cases. Table 4.4 shows the two-level PCA final results and presents the confusion matrix. As shown in Table 4.4, the two-level PCA algorithm did well in demarcating the FS and PCK collapse patterns. In addition, this method clarified that the primary confusion was between FS and MS. There are two possible reasons for this: the decision rule for MS, or the similarity of building behaviour during the collapse between FS and MS collapse patterns.

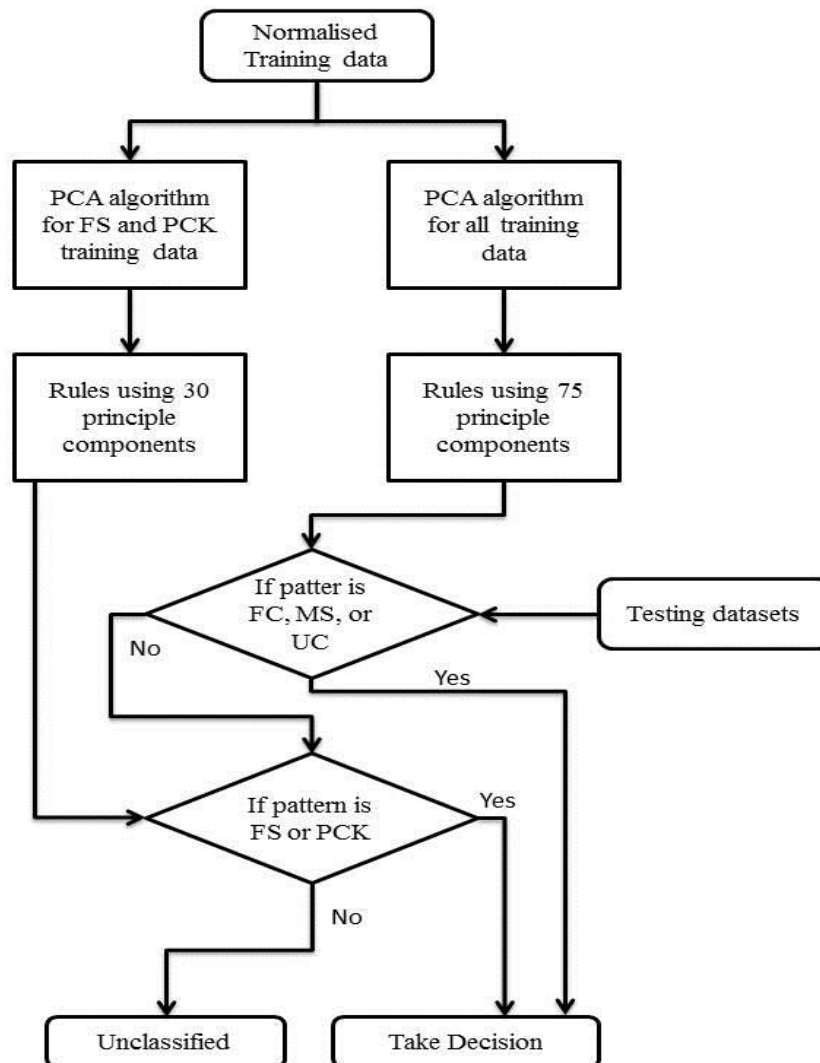


Figure 4.2: shows two-level PCA decision making process.

Table 4.4: Two-level PCA results and Confusion matrix.

Two-level PCA results							
				PCA (theoretical)			
Overall classification rate				96.4%			
First column (FC)				98%			
First storey (FS)				100%			
Mid-storey (MS)				86%			
Pancake (PCK)				100%			
Uncollapsed (UC)				98%			
Processing time in seconds				237 / 250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	49	1	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	7	43	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	1	0	0	49	0	50

In the first case, the rule covers 99.5% of MS collapse patterns. This means that only one pattern out of 15 training patterns does not match the rule. This could indicate that the MS collapse pattern has various scenarios not covered in the training data. Using Blender, it is impossible to generate 100% matching patterns, but a high level of similarity between collapse patterns under the same class is guaranteed.

In the second case, sensors in both the MS and FS collapse patterns travelled one storey during the collapse. This means that the velocity recorded can be close in the two patterns (see Table 4.1). However, the first storey in the MS collapse pattern did not move, and should give a major difference when compared to the first storey in the FS collapse pattern. The way the PCA algorithms were input into this method might explain the loss of demarcation between the FS and MS collapse patterns. As a result, we can conclude that MS behaves similarly to FS in some cases, based on

PCA features analysis. Sensor-by-sensor based analysis is needed to solve the confusion between the FS and MS collapse patterns, and to more fully recognise unmoved sensors and high velocity sensors (sensors fallen through three storeys).

4.2.3 Sensor-by-sensor based PCA

To increase PCA algorithm accuracy, each sensor can be treated individually. This means that PCA is implemented 48 times, based on sensor numbers in the simulated building. As a result, the 48 training datasets of principal components in each dataset contained 75 principal components. Optimum decision rules were derived from 48 training datasets (see Figure 4.3).

In this method, the PCA algorithm was fed one sensor from each set of training data, considering sensor position matching. Getting closer to each sensor reading by extracting features based on sensor location affects the way the classification is considered, using PCA from different viewpoints. This is confirmed by results and decision rules that reflect a contrast between sensors in different collapse patterns (see Figure 4.3). A clear classification of the FC, PCK and UC collapse patterns and pure decision rules reflects this method. Despite the confusion between FS and MS, we consider the results acceptable (see Table 4.5), taking into account the variability in datasets with each category. In addition, it is worth mentioning that two patterns were classified as UC, but were FS. This indicates that sensor-by-sensor based classification can sometimes erase pattern identity by relying on sensors features and ignoring pattern features.

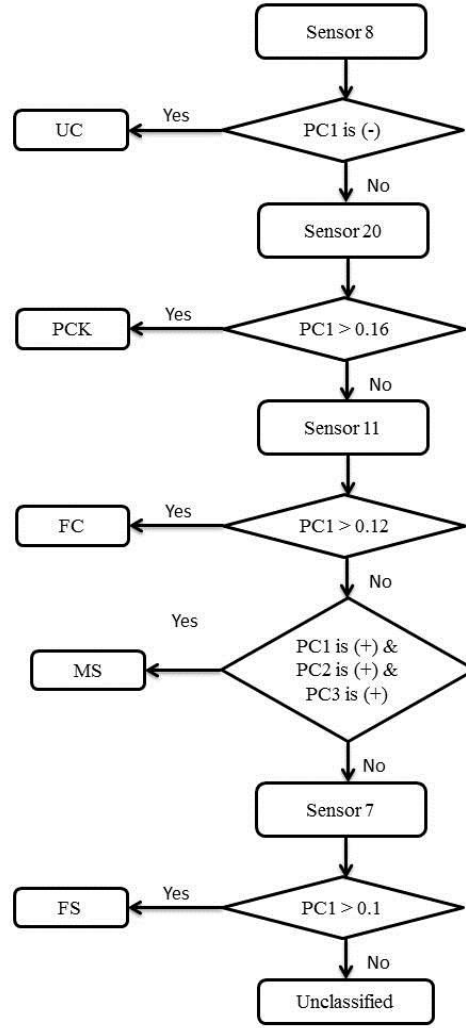


Figure 4.3: Decision-making rules for sensor-by-sensor based PCA.

The PCA algorithm results show high accuracy, considering the nature of implementation and the similarity in the sensor readings. The main advantage of this algorithm is its ability to extract features that create a boundary between categories, with low computation costs. However, incorrect classification forces a review of PCA algorithm performance when different scenarios of the same building collapse pattern are implemented (which are not part of this research). In Chapter 5, questions are raised regarding PCA algorithm performance when we implement sensor failure scenarios during a building collapse.

Table 4.5: Sensor-by-sensor based PCA results and Confusion matrix.

Sensor-by-sensor based PCA results							
				PCA (theoretical)			
Overall classification rate				97.2%			
First column (FC)				98%			
First storey (FS)				94%			
Mid-storey (MS)				94%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				65 / 250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	49	0	1	0	0	0	50
FS	0	47	1	0	2	0	50
MS	0	3	47	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Table 4.6: Final results of the PCA algorithm.

	PCA	Two-level PCA	Sensor-by-sensor based PCA
Overall classification rate	82.4%	96.4%	97.2%
First column (FC)	98%	98%	98%
First storey (FS)	88%	100%	94%
Mid-storey (MS)	86%	86%	94%
Pancake (PCK)	42%	100%	100%
Uncollapsed (UC)	98%	98%	100%
Processing time in seconds	191 / 250	237 / 250	65 / 250

4.3 Vector Quantisation Histogram

As mentioned previously, we have five building collapse patterns. To classify them using VQH, we used five codebooks. One dataset was used as an initial codebook, while another dataset was used for the training phase of each collapse pattern, using the VQ design toolbox in MATLAB. To increase methodological robustness, we

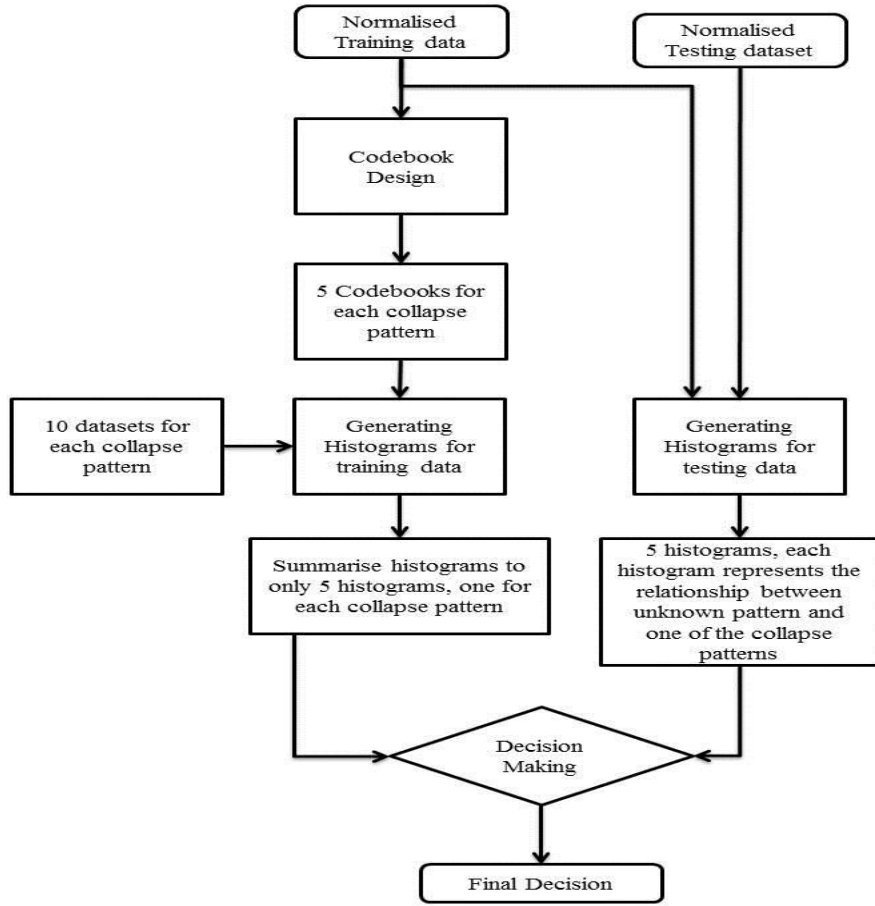


Figure 4.4: VQH implementation process.

used five codebooks per pattern instead of one. Codebooks were still considered as local, but we tried to cover more collapse scenarios in the same category.

Next, we created known histograms for each known collapse pattern using the training data with the codebooks from the first step (see Figure 4.4). We reduced these histograms to five by averaging the histograms belonging to the same category as representing a signature for each collapse pattern. In the third step, we generated unknown histograms using testing data and codebooks from the training data in step one. Each unknown pattern had five histograms that represented the relationship between the unknown pattern and pattern categories.

Table 4.7: VQH algorithm result and Confusion matrix.

VQ results							
				VQ (theoretical)			
Overall classification rate				88.4%			
First column (FC)				100%			
First storey (FS)				78%			
Mid-storey (MS)				88%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				281/ 250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	39	10	1	0	0	50
MS	1	3	44	2	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	08	0	0	38	0	50

During our decision-making phase, the Euclidean distance was used to classify the test patterns. To better demonstrate the results in Table 4.7, examples of histograms are highlighted in Table 4.8 (see Appendix D). In Table 4.8 (see Appendix D), single pattern histogram is generated using a single dataset, general histogram is the summarise of 50 histograms under the same category (Training phase), while misclassified row represents an example of misclassified UC collapse pattern during testing phase.

The FC collapse pattern displayed a clear signature in the histograms. For instance, fallen sensors were distributed between codewords 1 to 12 that represent the fallen sensors from different storeys, while the unmoved sensors were generally distributed between codewords 25, 37 and 40 which are part of unmoved sensors range. The FC trend was very clear in the pattern histograms, by creating groups to represent fallen and unmoved sensors. Unmoved sensors gave FC patterns a high weight for classification and distinguished them from other collapse patterns.

The MS collapse pattern displayed a similar trend, scoring high occurrences in codeword 2, which represents one of the unmoved sensors codewords. The fallen sensors were distributed among many codewords because all the fallen sensors travelled one storey during the collapse. This created similarities between sensors, regardless of their positions.

Like FC and MS, PCK collapse patterns achieved a high classification rate. Sensors that fell one storey played a major part in drawing a clear signature in the PCK histograms. These sensors were distributed across three or four codewords; while the remaining sensors covered a large number of codewords. In contrast, FS and UC classification rates were lower than for other collapse patterns. All sensors in the FS collapse pattern travelled one storey, which caused confusion with the MS histogram, despite some codewords showing a clear trend. Moreover, because unmoved sensors showed a high trend in FC, this created confusion with UC. All sensors were unmoved in UC and they were distributed highly through the first 12 codewords. This makes UC closer to the FC histogram, as shown in the example of incorrectly classified patterns in Table 4.8 (see Appendix D).

Some patterns thus benefited from having unmoved sensors, distinguishing them from patterns without unmoved sensors. Some patterns used the fallen sensors as a clear signature in their histograms, while others used both types of sensors to create their signature. Despite some collapse patterns having a high confusion rate as compared to other collapse patterns, they did not lack a signature. This confusion can be attributed to the nature of the VQ technique in assigning vectors to the closest codeword, as this can cause an undesirable reshaping of collapse patterns.

4.4 Hidden Markov Model Results

As part of experimental preparations, a Piecewise Aggregate Approximation algorithm (PAA) was used to convert sensor readings to a specific integer number to interpret the data using three different break point distributions (see Figure 4.5).

We required five models to represent the building collapse patterns. To increase methodological robustness, we generated a HMM of each training dataset. The model with the maximum likelihood for the testing data was the classification of unknown building collapse patterns (see Tables 4.9). The Baum-Welch training algorithm was applied to the HMMs.

In experiment one (see Table 4.9 and Table 4.10(a)), we designed the PAA algorithm to focus on positive and negative velocities and vibration areas (see Figure 4.5(a)). We aimed to detail general trends of sensor behaviour during building collapse.

Despite excellent results in most collapse patterns, the MS collapse pattern category failed completely. The manner in which PAA represented high velocity took no account of any high velocity that appeared only in three-storey fallen sensors. This caused great confusion between the MS and PCK collapse patterns.

To reduce confusion, we redesigned the PAA algorithm in a more logical way (see Figure 4.5(b)). Using two breaking points, sensor readings were classified into two categories: fallen and unmoved. In this way, we aimed to clarify HMM so that the algorithm produced less confusion between MS and PCK, as MS had 16 unmoved sensors.

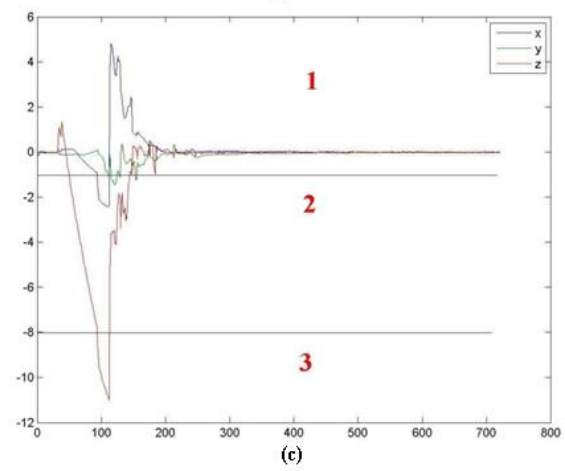
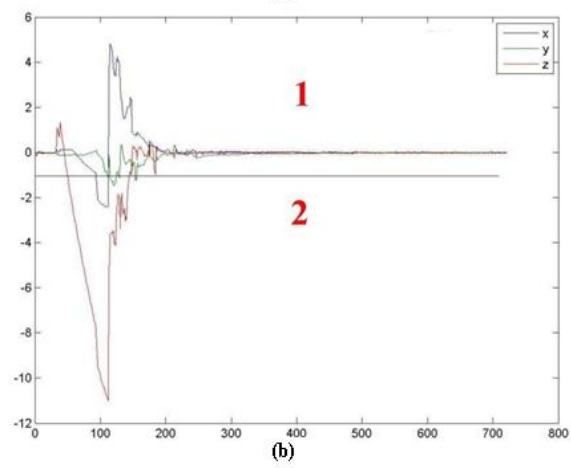
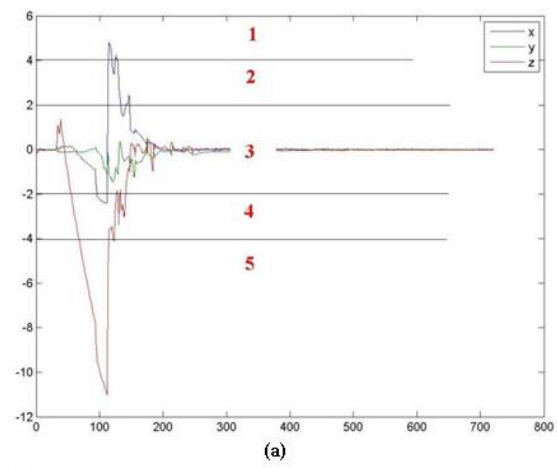


Figure 4.5: PAA breaking points.

Table 4.9: HMM results.

	Experiment 1	Experiment 2	Experiment 3
Overall classification rate	86%	92.8%	79.6%
First Column (FC)	100%	100%	100%
First Storey (FS)	88%	98%	98%
Mid-Storey (MS)	42%	76%	0%
Pancake (PCK)	100%	90%	100%
Uncollapsed (UC)	100%	100%	100%
Processing time in seconds	32400 / 250	22000 /250	23000 / 250
Breaking points	5	2	3
Window size	1	1	1

The experimental results displayed an improvement in MS classification accuracy. However, confusion remained regarding the FS collapse pattern, which reduced PCK and MS accuracy. MS and FS can be confused when sensors in both patterns fall one storey. Moreover, the PCK and FS collapse patterns have the same number of fallen sensors. This reading of number of fallen sensors creates a similarity between the two patterns, as does the breaking point design because the speeds of fall for the high fallen sensors were undistinguishable for the two pattern types.

We redesigned the PAA to be able to clearly separate high speed fallen sensors from other fallen sensors (see Figure 4.5(c)). We focused completely on clarifying any confusion regarding PCK collapse patterns. The result in Table 4.10(c) clearly shows that the new very high speed state ended any confusion with the PCK collapse pattern. However, it created a major confusion between MS and FS. The new state created a major similarity between FS and MS because no sensors in either of these patterns fell through three storeys, as was necessary for high velocities to occur.

**Table 4.10: Confusion Matrix of HMM (a) using PAA (five breaking points)
(b) using PAA (two breaking points) (c) using PAA (three breaking points).**

(a)

	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	44	0	6	0	0	50
MS	0	1	21	28	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

(b)

	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	0	1	0	0	50
MS	0	4	38	8	0	0	50
PCK	0	5	0	45	0	0	50
UC	0	0	0	0	50	0	50

(c)

	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	0	1	0	0	50
MS	0	50	0	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Thus, breaking points represent states in HMM. Adding or deleting breaking points had both advantages and disadvantages, creating confusion between patterns and affecting the overall results. The simplicity of data representation achieved the highest results with HMM. This could be the basis of HMM performance when dealing with sensor failure during building collapse, as detailed in Chapter 5. However, such simplicity can result in confusion between patterns that the HMM algorithm alone cannot improve.

4.5 HMM-PCA Hybrid Algorithm

The primary method behind the hybrid algorithm is to use the advantage of each algorithm in classifying building collapse patterns. In the previous section, HMM in

experiment 3 achieved 100% accuracy in classifying FC, PCK and UC collapse patterns, but with major confusion remaining between FS and MS collapse patterns. Attempting to improve the HMM results led to confusion between MS and PCK collapse patterns.

At this stage, we decided to use the sensor-by-sensor based PCA algorithm for only FS and MS collapse patterns as the second stage for HMM experiment 3 results. HMM was responsible for classifying FC, PCK and UC patterns only (see Table 4.11). If the unknown pattern were not one of the previous three patterns, the second stage would classify the pattern as either FS or MS. Sensors from various positions in the building showed a clear variance represented by principal components. Sensor 15 was the most robust sensor to extract the rule for pattern classification, as shown below:

If $PC(1) > 0.02$, then the collapse is FS otherwise it is MS.

Table 4.11: HMM_PCA results.

Hybrid method results							
				HMM_PCA (theoretical)			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				23005 / 250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

The results (see Table 4.11) affirmed the successful combination of these two algorithms. Using HMM features in taking in the count from each observation, and using PCA features to find dissimilarities between patterns, 100% accuracy was achieved.

4.6 Discussion

Building collapse pattern datasets can be analysed in many different ways, including sensor-by-sensor based analysis, fallen speed analysis, column- or row-based analysis, block-based analysis and whole pattern-based analysis. In this research, we covered pattern-based analysis using the PCA algorithm, sensor-by-sensor based analysis using the PCA algorithm and VQ technique, and fallen speed analysis using HMM. Each algorithm and technique achieved acceptable results in building collapse pattern classification, with optimum results achieved by combining the HMM and PCA algorithms. The HMM method confirmed its capacity to classify collapse patterns in the case of major dissimilarities by focusing on speed of fall patterns. Further, the PCA algorithm encountered difficulties in pattern classification when using whole patterns, but proved a robust algorithm for finding variance between patterns based on sensor features working as principal components.

Although VQ did not score the best results among other methods due to sensor distribution through the codebook, it could reduce the impact of sensor failure on the overall accuracy of VQ results. Before we make a decision on identifying which algorithm, or combination of algorithms, is best for this implementation, we must consider data used in this chapter is “theoretical” data. Theoretical data does not

include any real collapse scenarios such as sensor failure due to crushed or error manufacturing that can affect classification algorithm performance.

Chapter 5: Sensor Failure Scenarios

The aim of this chapter is to examine building collapse pattern classification algorithm reliability in more realistic scenarios, which would include sensor failure before or during building collapse. The consequences of sensor failure are varied, based on sensor position, collapse pattern and classification algorithm. There are many different causes for sensor failure, including being crushed during building collapse, power source failure, faults manufacturing and disconnection from the WSNs.

Two main scenarios are examined in this chapter. The first scenario, which is represented in Section 5.1, considers sensor failure on impact. In this scenario, sensor failure is caused by a sensor collision with an object at a certain speed or a sensor hitting the ground. The second scenario, which is examined in Section 5.2, is added to the first scenario by considering random sensor failure throughout the building collapse. The cause for this type of sensor failure can be due to the power source or a manufacturing error. This scenario could be more challenging for classification algorithms because sensor data is missing from the outset that can impact the efficiency of classification algorithms.

Each section includes a comparison between the sensor failure experiment and the experiment in Chapter 4 for the same classification algorithms. To distinguish between the results, results from chapter 4 are labelled theoretical results, as all sensors function during and after the building collapse, regardless of impacts.

5.1 Scenario 1: Sensor Failure on Impact (SFoI)

During a building collapse, sensor functionality is a major challenge due to debris that is caused by tremendous destruction during earthquakes. In comparison with theoretical sensor behaviour as mentioned in Chapter 4 (see Figure 5.1 (a)), sensor failure during building collapse has its advantages and disadvantages.

For example, if a sensor stops sending data after hitting the ground, it is considered an advantage for the classification algorithms because the sensor will not record any collapse noises (see Figure 5.1 (b)). Collapse noises can play a major part in creating similarity between collapse patterns by recording low vibrations, similar to uncollapsed patterns. Conversely, if a sensor hits an object before hitting the ground and then stops working (see Figure 5.1 (c)), it is considered a disadvantage for classification algorithms because it did not record the entire journey. This type of sensor failure could cause confusion between sensors fallen from different storeys.

The following sections present a reflection of the classification algorithms performance, including PCA, VQH, HMM and the hybrid HMM/PCA method. The sections focus on the performance and reliability of each classification algorithm under two types of sensor failure during building collapse.

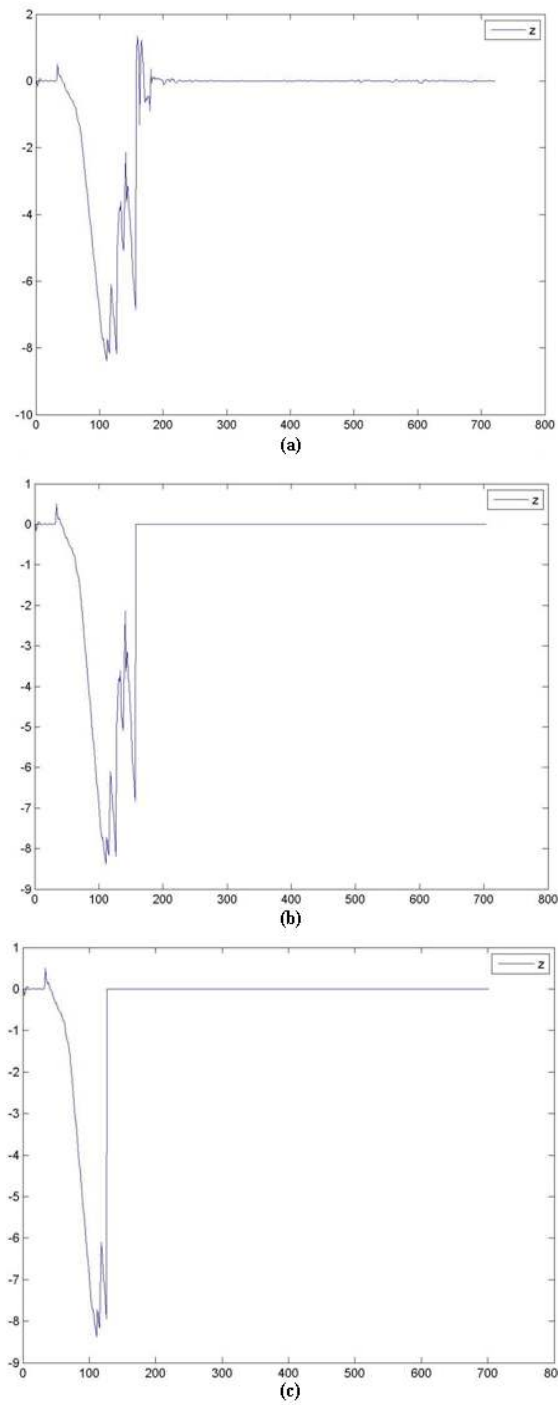


Figure 5.1: Sensor recording in three different cases: (a) theoretical case, (b) hitting the ground and (c) hitting an object during a fall.

5.1.1 Experimental Setup

All sensors in the training and testing patterns were subjected to a sensor failure check process to diagnose possibly damaged sensors during a building collapse. The diagnosing process is based on the changes in sensor velocity. A threshold was set for the level of velocity that can cause sensor damage after hitting an object or the ground. After locating the collision time, all subsequent sensor recording were changed to zero. Normalisation takes place after the sensor failure check process is finished.

5.1.2 Basic PCA Algorithm Performance under SFoI

In this section, each pattern represents a one-row vector, which is the same methodology used in the basic PCA algorithm in Chapter 4. The only difference is that the training and testing data were pre-processed by the sensor failure check. In this way, a new database is created for each pattern (testing and training datasets). In addition, a new set of rules is derived from the principal components using the new testing datasets to classify the collapse patterns (see Figure 5.2).

From the decision rules, it is clear that more than one principal component was used to assign one decision rule in most categories. This means that there is a high probability of confusion between patterns, especially if the number of sensors that malfunctioning before the end of the collapse are increased in the testing data. The outcome of the basic PCA algorithm implementation in this scenario (see Table 5.1) shows that algorithm accuracy drops by around 11% in comparison with the theoretical results presented in Chapter 4.

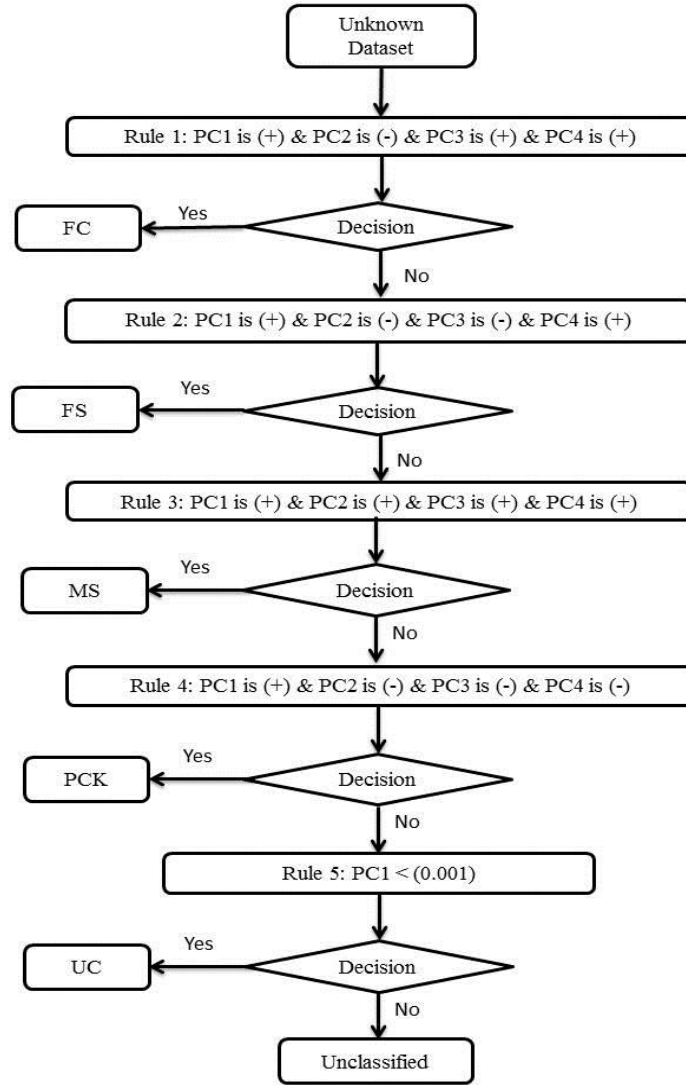


Figure 5.2: Basic PCA decision rules.

In addition, the confusion matrix in Table 5.1 shows a sharp decrease in FC collapse pattern accuracy and a significant rise in PCK collapse pattern accuracy. The FC results open the door for questions regarding basic PCA algorithm implementation reliability for building collapse pattern classifications. However, the basic PCA algorithm scored the lowest accuracy among the three ways to implement the PCA algorithm, as stated in Chapter 4. This leads to a conclusion that PCA by itself is not effective enough to be considered a suitable classification algorithm for a building collapse pattern.

Table 5.1: Basic PCA results and confusion matrix under SFoI.

Basic PCA results under SFoI							
					PCA (theoretical)	PCA (SFoI)	
Overall classification rate					82.4%	71.6%	
First column (FC)					98%	0%	
First storey (FS)					88%	100%	
Mid-storey (MS)					86%	100%	
Pancake (PCK)					42%	92%	
Uncollapsed (UC)					98%	66%	
Processing time (seconds)					191/250	200/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	46	0	4	50
UC	0	0	0	17	33	0	50

It is worth mentioning that the unclassified pattern jumped from two patterns in basic PCA in ideal result to 21 patterns in the current experiment. This indicates that sensor failure can mislead the classification algorithm by creating another category (unclassified patterns).

5.1.3 Two-level PCA under SFoI

To improve basic PCA algorithm outcome, two main issues need to be solved: (1) confusion between FC and MS and (2) confusion between PCK and UC collapse patterns (see Table 5.1). This means that the basic PCA algorithm can only correctly classify FS collapse patterns, and the remaining collapse patterns can be escalated to two-level PCA algorithms (see Table 5.1). New two-decision rule sets are used a set for each confusion (see Figure 5.3).

It is clear from the decision rules in the second level of the two-level PCA algorithm that the PCA algorithm clearly distinguishes between each collapse pattern by using

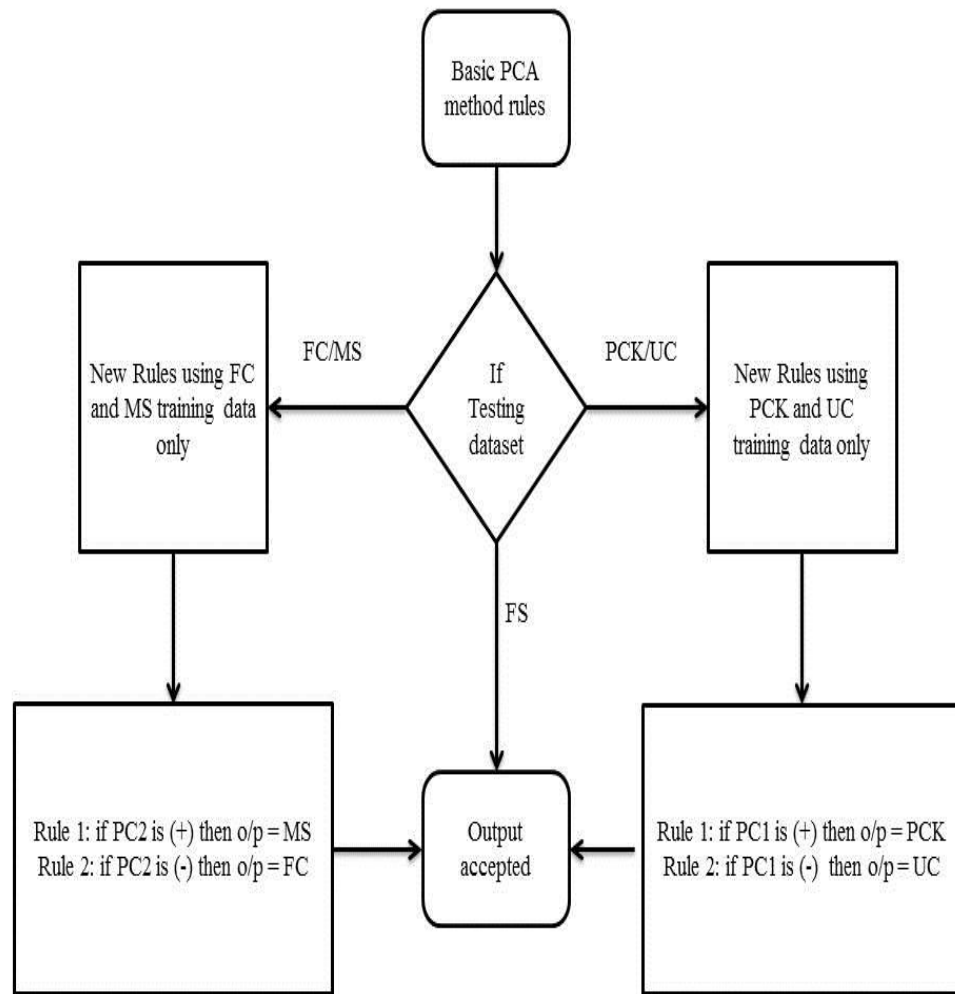


Figure 5.3: Two-level PCA methodology.

only one principal component for each rule. This will reflect positively on the algorithm outcome (see Table 5.2).

There is an increase of 20% in accuracy in this method when compared with the basic PCA, while a decrease of around 5% is in comparison with the theoretical results for a two-level PCA algorithm (see Table 5.2). A major improvement in the FC pattern in the two-level method appeared in comparison with the previous method (see Table 5.1). In addition, Table 5.2 reflects a clear picture of how the two-

Table 5.2: Two-level PCA results and confusion matrix under SFoI.

Two-level PCA results under SFoI							
					PCA(theoretical)	PCA (SFoI)	
Overall classification rate					96.4%	91.6%	
First column (FC)					98%	66%	
First storey (FS)					100%	100%	
Mid-storey (MS)					86%	100%	
Pancake (PCK)					100%	92%	
Uncollapsed (UC)					98%	100%	
Processing time (seconds)					237/250	250/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	33	0	0	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	46	0	4	50
UC	0	0	0	0	50	0	50

level PCA solved the confusion problem between patterns such as FC and MS (see Table 5.1). However, there is no indication of any improvement in unclassified when a two-level PCA was used to improve the outcome of the basic PCA algorithm results in Chapter 4, it solved the confusion between patterns and classified the unclassified patterns that appeared in the basic PCA method. Conversely, in this experiment, a two-level PCA could not place unclassified patterns into any class. This reinforces the idea that sensor failure on impact can significantly affect classification using PCA, by creating an unclassified class.

5.1.4 Sensor-by-sensor PCA under SFoI

As mentioned in Chapter 4, this method is heavily sensor-based rather than pattern-based. Therefore, in this scenario, sensor-by-sensor based analysis can be a high advantage for a PCA algorithm if the majority of sensors stop sending data after hitting the ground. From our observation, sensors collision with objects during a building collapse were most likely to occur short time before the sensors hit the

ground, but other sensors malfunctioning earlier in a building collapse based on the nature of a progressive collapse. This means that the amount of missing information is not significant. In addition, sensor-by-sensor based analysis can ignore early malfunctioning sensors by not including them in the decision-making process (see Figure 5.4). This process of ignoring early malfunctioning sensors improved the final results in regards to the confusion issue created by the similarity between sensors.

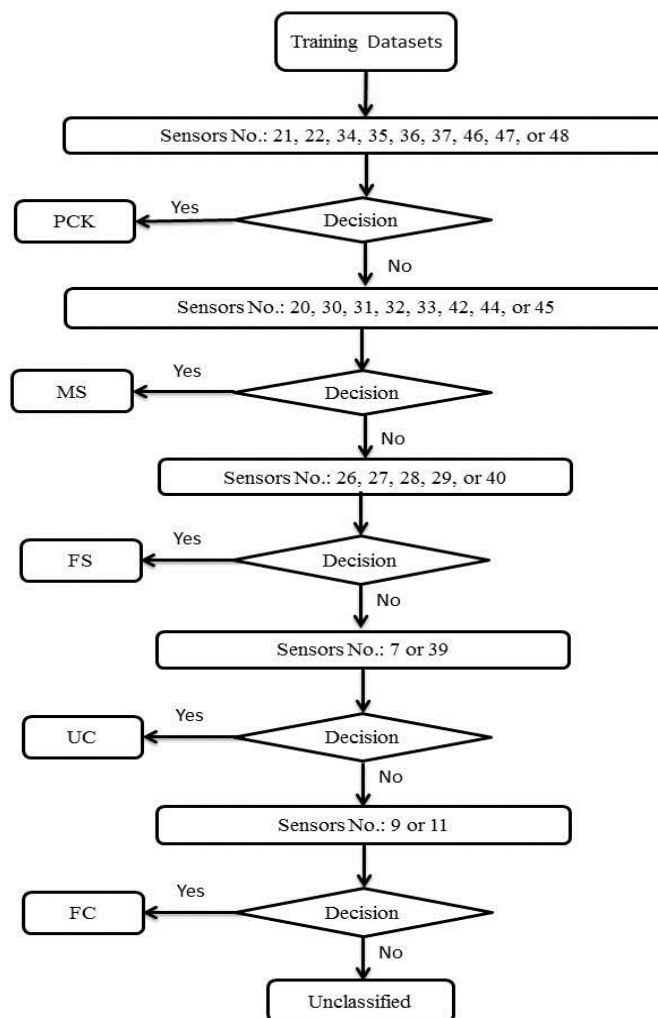


Figure 5.4: Number of sensors that can generate a reliable rule for each collapse pattern under SFoI.

Only one principal component was needed from each sensor involved in the decision rules. Figure 5.4 shows the decision rules in terms of the sensor number used to classify the collapse pattern type. This method shows the ability to generate a clear rule from more than one sensor for each collapse pattern class. An increase in the number of rules for each collapse pattern led to an increase in the flexibility and robustness of the decision-making process by offering a range of rules that fit many cases in the same collapse category. As a result, a sensor-by-sensor PCA algorithm scored around a 7% improvement compared with a two-level PCA algorithm at 1.6% in comparison with theoretical sensor-by-sensor PCA (see Table 5.3).

Table 5.3: Sensor-by-sensor PCA results and confusion matrix under SFoI.

Sensor-by-sensor based PCA results under SFoI							
				Sensor-by-sensor PCA (theoretical)		Sensor-by-sensor PCA (SFoI)	
Overall classification rate				97.2%		98.8%	
First column (FC)				98%		100%	
First storey (FS)				94%		94%	
Mid-storey (MS)				94%		100%	
Pancake (PCK)				100%		100%	
Uncollapsed (UC)				100%		100%	
Processing time in seconds				65/250		211/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	47	1	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

From the number of sensors used to create the decision rules and the outcome, we conclude that, based on a sensor-by-sensor analysis, sensor failure on impact actually improved the PCA algorithm. In addition, the sensor-by-sensor based method did not record any unclassified patterns. Unclassified patterns class is a feature of sensor-by-

sensor PCA that caused by the large number and clear decision rules involved in the decision-making process that created no place for unwanted classes. Nevertheless, the complete failure of random sensors (CFoRS) holds us back from giving sensor-by-sensor based analysis high credit in reliability, despite high accuracy results. A PCA algorithm using sensor-by-sensor based analysis results under CFoRS will clarify our final judgement.

To summarise, the sensor failure on impact scenario positively affects the sensor-by-sensor PCA algorithm and negatively affects both the basic and two-level PCA algorithms (see Table 5.4).

Some trends can drive the PCA algorithm's performance under the SFoI scenario:

- The complete failure to classify FC patterns using a basic PCA algorithm puts the method out of competition with other classification algorithms.
- Despite its inability to solve unclassified pattern issues with a basic PCA algorithm, a two-level PCA algorithm is still considered an effective solution for class confusion, and it can still be used in the hybrid method.
- Avoiding sensor data that create confusion is still a powerful tool for the sensor-based PCA algorithm.

Table 5.4: Final results of the PCA algorithm under SFoI.

	PCA		Two-level PCA		Sensor-by-sensor PCA	
	Theor.	SFoI	Theor.	SFoI	Theor.	SFoI
Overall classification rate	82.4%	71.6%	96.4%	91.6	97.2%	98.8%
First column (FC)	98%	0%	98%	66%	98%	100%
First storey (FS)	88%	100%	100%	100%	94%	94%
Mid-storey (MS)	86%	100%	86%	100%	94%	100%
Pancake (PCK)	42%	92%	100%	92%	100%	100%
Uncollapsed (UC)	98%	66%	98%	100%	100%	100%
Processing time in seconds	191/250	200/250	237/250	250/250	165/250	211/250

5.1.5 Vector Quantisation Histogram under SFoI

As mentioned previously, sensor failure caused by collision with an object can have positive and negative effects regarding noise reduction and data loss, respectively. In addition, it can affect sensor location in VQH by improving the histograms of the majority of sensors reach the ground or by negatively affecting the histograms if a reasonable number of sensors are early malfunctioning sensors.

Table 5.5 shows that VQH results improved by 0.6% in comparison with ideal VQH results. Moreover, slightly positive improvements were presented by removing confusion between FS and PCK and between MS and PCK collapse patterns (see Table 5.5 and Table 4.7). However, there appeared to be no change in the confusion between the UC and FC collapse patterns. Despite fallen sensors having a high

Table 5.5: VQH results and confusion matrix under SFoI

VQH results under SFoI							
					VQH (theoretical)	VQH (SFoI)	
Overall classification rate					88.4%	90%	
First column (FC)					100%	100%	
First storey (FS)					78%	82%	
Mid-storey (MS)					88%	92%	
Pancake (PCK)					100%	100%	
Uncollapsed (UC)					76%	76%	
Processing time in seconds					281/250	290/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	6	0	2	0	50
MS	1	3	46	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

impact (in regards to distinguish between patterns signatures) in this scenario compared to unmoved sensors, the majority of sensors in both collapse patterns were unmoved, which creates a similarity between the FC and UC collapse patterns.

In summary, the VQH technique produces a slight improvement under sensor failure during a building collapse scenario. Therefore, VQH can be considered another step towards improving method robustness under natural catastrophe challenges. However, the VQH technique still suffers from the confusion between UC and FC collapse patterns, which has high costs in implementation during normal operation, causing a significant amount of false-positives.

5.1.6 Hidden Markov Model under SFoI

As mentioned in Chapter 4, the PAA algorithm is used in three different settings based on a number of breaking points (see Figure 4.6). The PAA algorithm with two breaking points achieved the best results with the HMM. As a result, a PAA algorithm with two breaking points and a HMM were examined under sensor failure during a building collapse. The experimental results show that the HMM performed with 100% accuracy (see Table 5.6), which is considered the highest classification accuracy for an individual algorithm thus far.

To understand how this optimum result was achieved, a clear explanation of how PAA and HMM work together is needed. The main idea behind using only two breaking points in a PAA algorithm is to create two states in HMM: one for a low-velocity range, which is represented by unmoved sensors, vibration and noise, and one for a high-velocity range, which represents the fallen sensors. For example,

sensor failure in scenario one (SFoI) reduced noise in the sensor data, which gave more credit to the low-velocity state in each sensor and reduced credit in the high-velocity state. The effect of noise created a level of similarity that affected the classification results of the HMM in Chapter 4. For instance, major confusion in the

Table 5.6: HMM results and confusion matrix under SFoI.

HMM results under SFoI							
				HMM (SFoI)			
Overall classification rate				100 %			
First column (FC)				100 %			
First storey (FS)				100 %			
Mid-storey (MS)				100 %			
Pancake (PCK)				100 %			
Uncollapsed (UC)				100 %			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM ideal results was between patterns that had a close number of fallen sensors, such as the confusion between the MS and PCK collapse patterns. Noise level dropped sharply in the sensor failure on impact scenario, which gave the HMM method a greater chance to draw a sharp border between collapse patterns.

In summary, most classification algorithms' performance improved under the sensor failure on impact scenario, primarily due to noise reduction and declining data that would not exist in real-life. However, not every sensor failure on impact can have a positive effect on algorithms' performance. As mentioned, early malfunctioning sensors can cause performance degradation due to an amount of missing data. The hybrid method could be used to improve the HMM results. However, in this

scenario, there is no need to use it because the HMM method achieved 100% accuracy.

5.2 Scenario 2: Complete Failure of Random Sensors (CFoRS)

This sensor failure scenario is applied upon the SFoI scenario. The sensor failure in this scenario is complete throughout and can be caused by a manufacturer's error, failure in power supply, communication issue or any sudden interruption caused by an earthquake. This chapter investigates the robustness of classification algorithms when presented with random missing data.

It is helpful to define least and most dependent sensors or cases terminology that used often in this section.

- Least dependent sensor means the sensor has lowest impact on classification algorithm performance when that sensor's data is missing. Least impact on accuracy (LloA) case (or result) is a combination of number of least dependent sensors that has the least impact on the classification algorithm performance when sensors' data are missing.
- Most dependent sensor means the sensor has high impact on the classification algorithm performance when that sensor's data is missing, while most impact on accuracy (MloA) case (or result) is a combination of most dependent sensors that has one of the highest impacts on classification algorithm performance.

The main objective of this scenario is to examine the limit of each classification algorithm. For this reason, two levels stress-testing were examined. The first level

examines each classification algorithm with a maximum of six sensors failure. Each sensor from the 48 sensors in the training datasets was examined to find the six LIoA and six MIoA failed sensors. Then, from the six LIoA and MIoA sensors, each combination was examined to find the LIoA and MIoA case of each classification algorithm when the sensors failure were between one and six. However, in the HMM algorithm, a random number of sensor numbers was chosen to assemble the combinations starting from two sensors to six sensors for the LIoA and MIoA cases. This methodology was used due to time limitations and because almost all sensors gave the same outcome when tested individually.

If the classification algorithms passed the first level of stress-testing with greater than, or equal to 60%, the algorithm will input to the second level of stress-testing. Level two was the row and column failure test, which meant that all sensors in a tested column or row would fail completely during a building collapse. It was considered the most difficult test for classification algorithms used in this research. Each section contained the LIoA and MIoA cases from the level one test and the MIoA case of the level two test. The remainder of the results can be found in Appendix B.

5.2.1 Basic PCA under CFoRS

This method scored 74.8% with least impact on accuracy, regardless of the number of failed sensors (see Table 5.7). It is clear that the major effect appeared in the FC collapse pattern. The majority of FC patterns were classified as MS collapse patterns, while the PCA algorithm could not classify the rest of the FC testing patterns (see

table 5.7). In the MIoA case, the results were 53.2–60%; a result of 53.2% was obtained when four sensors failed.

Table 5.7: Basic PCA results (LIoA) and confusion matrix under CFoRS.

LIoA Basic PCA results under CFoRS							
				Basic PCA (CFoRS)			
Overall classification rate				74.8 %			
First column (FC)				0 %			
First storey (FS)				100 %			
Mid-storey (MS)				100 %			
Pancake (PCK)				100 %			
Uncollapsed (UC)				74 %			
Processing time in seconds				300/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	13	37	0	50

By achieving this level of accuracy (see Table 5.8), the basic PCA algorithm failed in the level one test, thus there was no need for a level two test. In addition, this method failed to classify the FC collapse pattern from the LIoA results and to classify PCK collapse patterns in the MIoA case. This failure could be because the four sensors were based in the third storey, which is the main feature of the PCK collapse pattern.

It is worth mentioning that the basic PCA LIoA and MIoA results indicate that a complete failure of some sensors can improve the results by removing some confusion. This method scored 74.8 % when six sensors failed, and it achieved 71.6% under the SFoI scenario.

Table 5.8: Basic PCA results (MIoA) and confusion matrix under CFoRS.

MIoA results of Basic PCA under CFoRS							
				Basic PCA (CFoRS)			
Overall classification rate				53.2 %			
First column (FC)				0 %			
First storey (FS)				100 %			
Mid-storey (MS)				100 %			
Pancake (PCK)				0 %			
Uncollapsed (UC)				66 %			
Processing time in seconds				300/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	3	0	0	47	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	4	0	0	0	46	50
UC	0	2	0	15	33	0	50

5.2.2 Two-level PCA under CFoRS

This method scored 93.2% as LIoA result, regardless of the number of failed sensors (see Table 5.9). The main effect appears to be in relation to the FC collapse pattern. In the MIoA case, the results were between 60% and 84.8%, where 60% appears when four to six sensors failed (see Table 5.10). The main failure in this method appears in classifying the FC and PCK collapse patterns.

The majority of miss-classified patterns appear under the unclassified class, which highlights the link between the flexibility of the PCA decision rules and the number of unclassified patterns. Scoring 60% in a level one test means that the method is not qualified to progress through to the second test.

A sign of improvement caused by complete sensor failure is shown in this method in comparison with LIoA case (93.2%) of this method and the result of this method under the SFoI scenario (91.6%). Therefore, randomised sensor failure has less of an impact on the results.

Table 5.9: Two-level PCA results (LIoA) and confusion matrix under CFoRS.

LloA result of 2-level PCA under CFoRS							
				2-level PCA (CFoRS)			
Overall classification rate				93.2 %			
First column (FC)				66 %			
First storey (FS)				100 %			
Mid-storey (MS)				100 %			
Pancake (PCK)				100 %			
Uncollapsed (UC)				100 %			
Processing time in seconds				320/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	33	0	0	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Table 5.10: Two-level PCA results (MIoA) and confusion matrix under CFoRS.

MIoA result of 2-level PCA under CFoRS							
				2-level PCA (CFoRS)			
Overall classification rate				60 %			
First column (FC)				4 %			
First storey (FS)				100 %			
Mid-storey (MS)				100 %			
Pancake (PCK)				0 %			
Uncollapsed (UC)				96 %			
Processing time in seconds				320/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	2	0	0	0	0	48	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	0	0	50	50
UC	0	2	0	0	48	0	50

5.2.3 Sensor-by-sensor PCA under CFoRS

Thus far, the sensor-by-sensor based PCA algorithm was the highly accurate method using PCA algorithm in relation to overall PCA algorithm results. However, relying completely on sensors without considering the dataset as a pattern in general was always under debate of this work researches. In a LIoA case, this method shows almost no effect on accuracy, regardless of the number of failed sensors. It scored 99.6% accuracy with only one dataset of FS collapse pattern misclassified as MS collapse pattern. Conversely, in the MLoA case, the results ranged from 55.8% to 99.2%, where 55.8% occurred when four to six sensors failed (see Table 5.11). The complete failure in classifying FC and UC patterns appears when the sensor failure occurs in sensors that are involved in the decision-making rules. In addition, based on building prototype and sensor distribution in the building, only two sensors that

Table 5.11: Sensor-by-sensor based PCA results (MLoA) and confusion matrix under CFoRS.

MIoA result of sensor-by-sensor based PCA under CFoRS							
				Sensor-by-sensor PCA (CFoRS)			
Overall classification rate				58.8 %			
First column (FC)				0 %			
First storey (FS)				94 %			
Mid-storey (MS)				100 %			
Pancake (PCK)				100 %			
Uncollapsed (UC)				0 %			
Processing time in seconds				250/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	0	50	50
FS	0	47	1	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	0	50	50

are involved in decision making can degrade the results from 99.2% to 78.8% using a sensor-by-sensor based PCA algorithm. The 58.8% accuracy terminated the last hope on one of the PCA algorithm methods can make it to the level two test.

5.2.4 VQH under CFoRS

In the VQH technique, complete sensor failure is interpreted as an unmoved sensor, as VQH assigns the sensor failure record to the closest vector word in a codebook. When the failure is recorded as zeros, this is similar to the static sensor's vector word. In addition, all failed sensor records will go under only one vector word in the codebook, which can change unmoved sensor distribution in a histogram and create confusion between collapse patterns.

The LIoA case in this scenario is when one or two sensors failed. VQH achieved 90.4%, which is classify one unknown pattern correctly more than the SFoI scenario. In the MLoA case, the lowest accuracy appeared when a combination of five sensors failed, with all of them located in the second storey (see Table 5.12).

Table 5.12: VQH results (MLoA) and confusion matrix under CFoRS.

MlOA result of VQH under CFoRS							
				VQH (CFoRS)			
Overall classification rate				86.8 %			
First column (FC)				100 %			
First storey (FS)				82 %			
Mid-storey (MS)				76 %			
Pancake (PCK)				100 %			
Uncollapsed (UC)				76 %			
Processing time in seconds				300/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	6	0	2	0	50
MS	11	1	38	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

From the MIoA case result, the VQH technique still showed an acceptable result, regardless of the fact that six sensors malfunctioning. This result was encouraging to put the VQH technique through the level-two test, which was a row-column failure. Moreover, the second test will explain in more detail why the main confusion appears between the FC and MS collapse patterns when five sensors failed.

Surprisingly, the results fluctuated between 76% and 80.4% when all sensors in a column failed at the same time. However, it was not the case when all sensors were malfunctioning in a storey at the same time. The VQH technique obtained 36.4% accuracy when all sensors in the mid-storey were malfunctioning at the same time (see Table 5.13), demonstrating a difficult degradation in the classification algorithm in response to storey-wide sensor failure.

Table 5.13: VQH results (MS sensors failure) and confusion matrix under CFoRS.

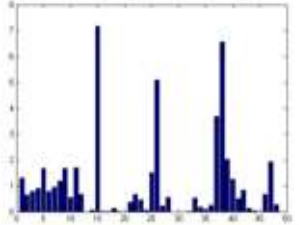
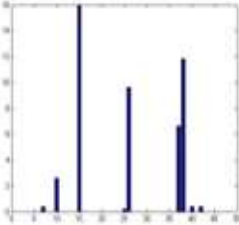
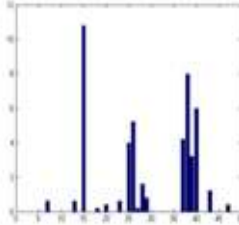
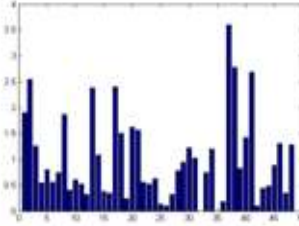
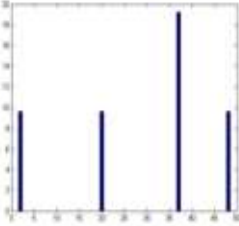
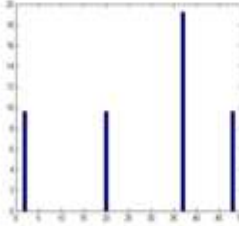
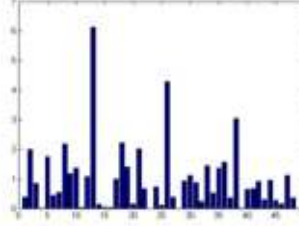
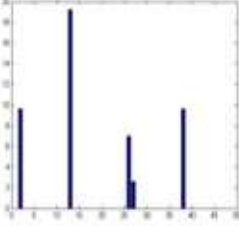
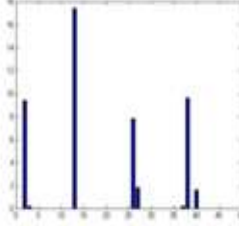
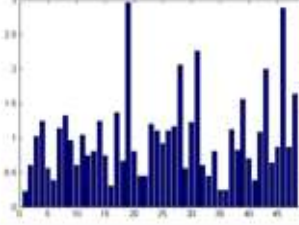
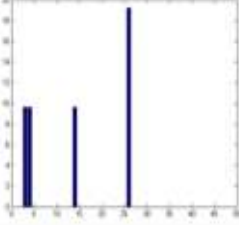
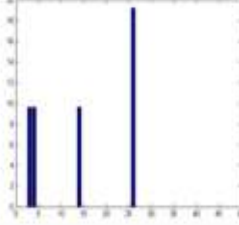
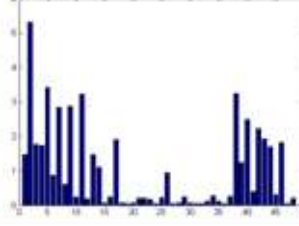
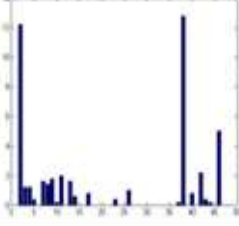
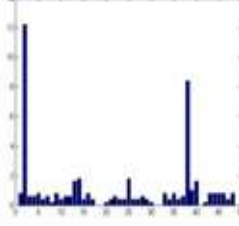
The failure of VQH algorithm under CFoRS							
				VQH (CFoRS)			
Overall classification rate				36.4 %			
First column (FC)				100 %			
First storey (FS)				54 %			
Mid-storey (MS)				0 %			
Pancake (PCK)				28 %			
Uncollapsed (UC)				0 %			
Processing time in seconds				320/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	10	27	13	0	0	0	50
MS	50	0	0	0	0	0	50
PCK	1	0	35	14	0	0	50
UC	50	0	0	0	0	0	50

As shown in Table 5.13, all collapse pattern results were affected by the mid-storey sensor malfunctioning, except for the FC collapse pattern. Two collapse patterns created this similarity, which led to catastrophic results in the MS and FC collapse patterns. Major confusion with MS comes from the FS and PCK collapse patterns. FS and PCK have a common feature, which is that all sensors are moved during a building collapse, and MS came second after FS and PCK in terms of the number of moved sensors. This means that mid-storey malfunctioning sensors in FS and PCK created a similarity by creating an unmoved storey, and the MS collapse pattern had one. It is the same case concerning the confusion between the FC and MS collapse patterns. The number of unmoved sensors increases in MS datasets because of mid-storey malfunctioning sensors that create a similarity with FC patterns with the closest number of unmoved sensors.

To understand why all UC testing datasets were misclassified as an FC collapse pattern, an investigation is required into the histograms' features of both UC and FC patterns (see Table 5.14). Earthquakes cause vibrations for all buildings, regardless of whether they collapse. This vibration created a reading in all UC pattern sensors. This type of record caused wide sensor distribution in the UC pattern's histogram (see Table 5.14). Conversely, the moved sensors in the FC collapse pattern were distributed from number 1 to 12 codeword in the histogram, while the unmoved were distributed through the rest with an accumulation of two or three codewords in the FC histogram (see Table 5.14). After understanding how the histograms were organised in both the FC and UC patterns, another investigation was carried out on how mid-storey malfunctioning sensors changed the distribution of codewords in UC test patterns' histograms. Firstly, an investigation of codewords distribution in the

UC histogram that generated by the trained codebook was carried out. Three to four codewords in the generated histogram were accumulating most of the sensors' records, which created an increase in the distance between histogram from the test data and histogram from the training data (see Table 5.14).

Table 5.14: An example of two misclassified unknown patterns.

Collapse Pattern Type	Trained Histograms	Histograms of Unknown Collapse Pattern (UC)	Histograms of Unknown Collapse Pattern (MS)
FC			
FS			
MS			
PCK			
UC			

Secondly, when UC test datasets went through FC trained codebook, the generated histograms carried a major similarity with the major areas in histograms from the training FC patterns. Despite there being no match between the FC histogram and UC–FC histograms in regard to moved sensors codewords in FC histogram, the distance between them was still less than the distance between the UC training data histogram and the UC–UC histograms (see Table 5.14).

Similar results were obtained when malfunctioning sensors appeared in the third storey. However, it was not the case with the first storey. An accuracy of 74.8% was obtained when malfunctioning sensors appeared in the first storey using the VQH technique. The major improvement was in the disappearance of the MS–FC confusion and a reduction in the UC–FC confusion. Moreover, there was a slight enhancement in the PCK–MS confusion results.

From the overall results of the VQH technique, VQH shows the ability to survive when 25% of sensors have failed, when those sensors are arranged in columns. However, when a storey of sensors failed, the impact was more devastating. In addition, VQH shows that some sensors' combinations have higher classification value than the others. That is, VQH shows that storeys two and three are more valuable than the ground storey, which could relate to the building design (not part of this research). In general, if one-third of sensors are malfunctioning, VQH will fail to classify building collapse patterns.

5.2.5 HMM under CFoRS

As mentioned in scenario one, the HMM algorithm results show an improvement compared to the HMM results in Chapter 4. In the LIoA case scenario, the HMM is not affected by the failure of six sensors and still scored 100% accuracy. Conversely, in the MLoA case scenario, the HMM results accuracy decreased to 80.8% when five to six sensors failed (see Table 5.15). This primarily occurred when most of the MS collapse patterns were classified as FS collapse patterns. This type of confusion occurs in the HMM when three breaking points were used in PAA algorithm, but it does not mean that they share the same cause. However, the overall results are acceptable and this type of confusion can be resolved by using the hybrid method (see Section 5.2.5). By scoring over 60% accuracy, the second level of stress-test was applied. In the column collapse case, the MLoA result (65.2%) was achieved when

Table 5.15: HMM results and confusion matrix under CFoRS.

MIoA results of HMM under CFoRS							
				HMM (CFoRS)			
Overall classification rate				80.8 %			
First column (FC)				100 %			
First storey (FS)				100 %			
Mid-storey (MS)				4 %			
Pancake (PCK)				100 %			
Uncollapsed (UC)				100 %			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	48	2	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

the first column sensor malfunctioning and the majority of the FC collapse patterns were classified as UC patterns (see Appendix B). when storey of sensors malfunctioning, the PCK collapse patterns were misclassified when any of the storey sensors failed (see Appendix B). In addition, the HMM algorithm reached 60% with the failure of either the first- or second-storey sensors. The common feature between the first- and second-storey sensors' results was that the algorithm misclassified two classes completely (see Table 5.16). For instance, the first-storey case missed the FS and PCK collapse patterns completely, while the MS and PCK collapse patterns were misclassified in the second-storey case. However, the confusions between patterns

Table 5.16: Confusion matrices of first- and second-storey cases using the HMM algorithm.

(a) Confusion matrix of first-storey case							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	0	0	49	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	50	0	0	0	50
UC	0	0	0	0	50	0	50
(b) Confusion matrix of second-storey case							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	50	0	0	0	0	0	50
PCK	0	50	0	0	0	0	50
UC	0	0	0	0	50	0	50

occurred between different patterns in the first- and second-storey case. For example, the PCK collapse patterns were classified as MS collapse patterns in the first-storey case, while they were classified as FS collapse patterns in the second-storey case.

In general, confusion between patterns that appeared in the first- and second-storey cases is different to the confusion between the patterns created by the columns'

cases. This point can be considered as a weakness in the HMM's performance. Nevertheless, uncollapsed patterns were classified as 100% accurate in both the LIoA and MIoA cases of the HMM algorithm.

In summary, the HMM algorithm's performance was reasonable, considering the algorithm failure that occurred in the level two stress test. Algorithm failure and the randomness confusion between different types of patterns were caused by the large number of early malfunctioning sensors, which reached one-third of the total sensors in some cases. A highly accurate classification rate of UC patterns can be a useful feature of the HMM algorithm in normal operation when no natural disaster strikes.

5.2.6 HMM-PCA Hybrid Algorithm under CFoRS

The main trend in the HMM results shows that the MS-FS collapse patterns' confusion appears in many tested scenarios, such as the MIoA six malfunctioning sensors and first-column malfunctioning sensors. For this reason, we chose the same methodology that was used in Section 4.6, which was represented by two steps in the classification process: the HMM algorithm applied in the first step to classify FC, PCK and UC collapse patterns, and the PCA sensor-by-sensor based method to classify FS and MS collapse patterns.

The hybrid method improved the MIoA six sensors failed case from 80.8% to 100% and the MIoA column case from 65.2% to 80% (see Appendix B). However, this method did not have any influence on the MIoA row case, which was represented by storey one and two. The main reason was that each row case had its own confusion

type, which was considered different to other tested case scenarios, and it was not efficient to design a rule for every confusion. For example, in the first-storey case, the main confusion was between the FS–FC and the PCK–MS collapse patterns, while the MS–FC and PCK–FS collapse patterns were the main confusion of the second-storey case. The hybrid method still has a positive effect on the classification algorithm despite having no effect on the MIOA case scenario.

5.3 Chapter Summary

Real-case scenarios were simulated in two main scenarios; sensor failure on impact and random sensor failure. Each scenario reflected the classification algorithms' performance in different ways. The sensor failure on impact scenario indicated how sensor failure could contribute positively if failure occurred when the sensor hit the ground. These are the datasets that more accurately mimic real-life (destruction of sensor upon impact). This was clearly represented in the VQ and HMM results. The second scenario was designed to target the classification algorithms' capabilities and limitations. The VQ and HMM algorithms went further than the PCA algorithm in this test; one-third of total sensors was the limit of both the VQ and HMM algorithms with respect to performance diversity. The hybrid method showed some improvements in the overall results, but it failed to extend the limitation of the HMM algorithm.

Chapter 6: Conclusion

Three classification algorithms were used to classify five building patterns: four collapse patterns and an uncollapsed building pattern. The algorithms were used in various scenarios and tests to examine their robustness and reliability under various simulated cases, inspired by real-world scenarios.

A summary of the ideas and results of this thesis will be presented in Section 6.1. Section 6.2 provides the discussion and conclusion. The discussion will present the most important trends that appeared during the research, supported by the experimental statistical results. Conclusion is presented and supported by evidence derived from the results. Finally, Section 6.3 will discuss the next stage for this research and other ideas that could be implemented to improve the classification algorithms' performance.

6.1 Thesis Summary

Extensive research has been conducted to understand information gathering in the field of disaster aid, as well as rescue teams' operations and procedures during earthquakes. Aerial and satellite images represent one of the main research areas that provide rescue teams with useful information regarding stricken-zone situations. These research areas covered building collapse classification in various manners. Some researchers classified buildings as damaged or undamaged, while others classified them as collapsed or uncollapsed. This type of crisis aid has its costs and challenges. Processing time, which is a crucial factor in a time of crisis, is considered

one of the main challenges for most of the aerial or satellite image-processing approaches, while weather conditions can be a major challenge to collecting high-quality images for the stricken-zone. Conversely, there is no need for robust infrastructure to implement this type of research, as the information resource will not be directly affected by the crisis or the damaged infrastructure in the stricken area.

We derived a novel approach to classifying building collapse patterns by using WSNs and classification algorithms. The aim of this approach was to classify building collapse patterns in more detail than just “collapsed” or “uncollapsed”, so that additional useful information could be obtained in order to help emergency management teams during rescue operations. The main achievement of this research was successfully obtaining highly accurate classifications in the shortest time for rescue operations. In addition, datasets that can simulate real-life situations were designed, to examine the classification algorithms’ performance and reliability. The following steps were conducted during this research:

- We simulated building collapse patterns using the Blender Game Engine. Four collapse patterns were the subject of this work, as well as an uncollapsed building pattern. Capturing behaviour pattern during an earthquake was conducted using simulated WSNs with the assistance of Python and Blender Game Engine. In addition, simulated earthquake forces were used according to the literature.
- Three classification algorithms (PCA, VQH and HMM) were chosen to extract the signatures of each collapse pattern type from a training dataset and to classify a larger testing dataset. The first test for the classification algorithms was classifying the test data without any external or

environmental effects, which is called the theoretical case. In this stage, the hybrid method from the HMM–PCA was used to achieve 100% classification accuracy.

- To be more realistic, a new test scenario was simulated to deal with real conditions that WSNs could face during a building collapse. This test was called sensor failure on impact (SFoI) and it targeted sensor failure when collision with an object or hitting the ground during a building collapse. Highly accurate results were achieved, with the HMM algorithm reaching 100%.
- To test the performance limit of each classification algorithm, another test was designed to find the highest number of early sensor failures before the classification algorithm fell below 60% accuracy. The first stage of this test chose the MIoA combination from six sensors with MIoA. The HMM and VQ algorithms passed this stage, while the PCA algorithm achieved lower than 60% accuracy. The second stage was the row–storey early sensor malfunctioning case. The HMM and VQ algorithms' performance was tested when 25% to 33.3% of total sensors failed at the same time. Both algorithms achieved 60% or lower when 33.3% of sensors failed to function during a building collapse.

6.2 Conclusion

The three classification algorithms were subjected to various types of experiments and tests, and each algorithm displayed a trend (see Figure 6.1). The PCA algorithm was implemented in three different ways to achieve the best results. There are two main trends in a PCA algorithm: a pattern-based trend, which is represented by basic

PCA and two-level PCA methods; and a sensor-by-sensor based trend, which is represented by sensor-by-sensor based PCA method. The best results achieved using the pattern trend in the PCA algorithm by using a two-level PCA, which obtained 91.4% accuracy in the SFoI scenario. Nevertheless, the method failed in the first-stage test in the CFoRS scenario, which indicated that the pattern recognition in the PCA algorithm relied heavily on rich information coming from a collapsed building, and any missing information could result in crucial degradation to the algorithm's performance.

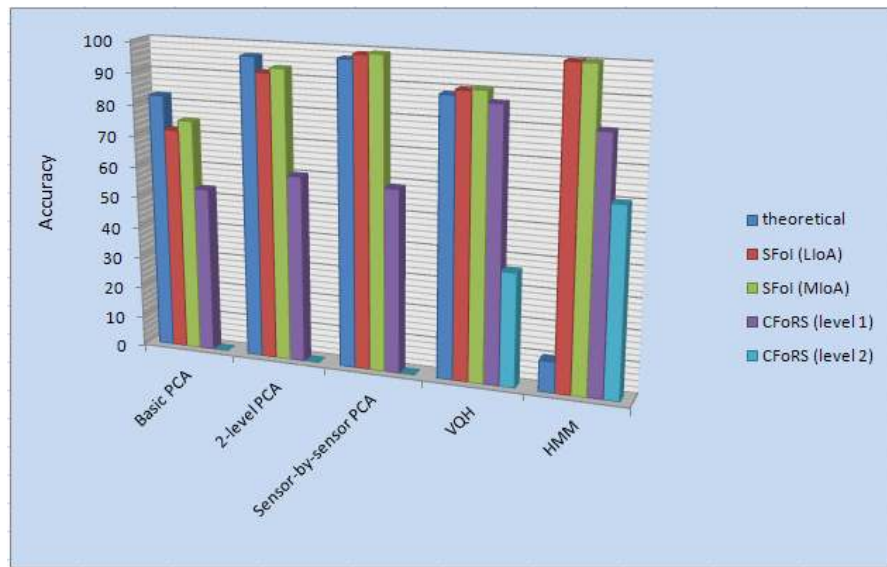


Figure 6.1 shows classification algorithms results under different case scenarios.

The sensor-by-sensor based trend in the PCA algorithm was the only method among the PCA implementation methods that scored an improvement (1.6%) when the SFoI scenario was tested. Less noise improved this method in terms of extracting clearer rules from principal components (PCs). In addition, this method was the most accurate method among the other PCA methods. However, it scored the lowest accuracy (36.4%) among other PCA methods when the first-stage test (CFoRS) was

implemented (see Figure 6.1). The sensor-by-sensor based PCA result depends on individual sensors to recognise the building collapse pattern, which can include a high risk in terms of classification accuracy, as any sensor malfunctioning at any time during a building collapse.

The VQH algorithm was tested using sensor-by-sensor based analysis. Histograms represent a signature for each building collapse pattern, and sensors distributed through histograms are based on trained codebooks. VQH suffered from low classification accuracy in the uncollapsed building class due to the high similarity between unmoved sensors in both the FC and UC pattern histograms. This type of confusion can be a major weakness in algorithm reliability in terms of day-to-day function, regardless of the overall accuracy results. Conversely, regardless of the UC pattern issue, VQH passed the first stage (CFoRS) with a MIoA case of 86.8%, which means that failure in six sensors had little affect on the algorithm's performance (see Figure 6.1). Moreover, the algorithm passed column sensor failure when 25% of the total sensors had early malfunctioning (see Appendix B) and scored 36.4% when one-third of the total sensors in one storey failed to function during a building collapse (see Table 5.14). The similarity issue that caused the UC–FC confusion in the theoretical test was escalated in the CFoRS scenario and created wider confusion with other collapse patterns, resulting in classification accuracy reaching less than 60%.

The HMM algorithm extracted building collapse pattern signatures from the time-series of each class. Based on sensor velocity in each collapse pattern, the HMM reached the optimum results by achieving 100% in the SFoI scenario. In addition, the HMM passed the first stage in the CFoRS scenario and scored just 60% accuracy in

the stage two (CFoRS) scenario. Despite confusion appearing between different patterns in the stage two test, the HMM managed to keep the UC pattern classification rate at 100% all the time and, as mentioned previously, this point can be considered an advantage for this algorithm. Conversely, the HMM had the highest processing time compared with the PCA and VQH algorithms, but it is still acceptable when compared with the crisis aid systems that are currently used (classifying building collapse only) and image processing approaches that investigate the collapse patterns hours or even days after the crisis.

In conclusion, every pattern recognition algorithm has limitations, based on how much information is missing compared with the theoretical pattern. The HMM scored the highest classification accuracy in terms of LIoA- and MLoA-case scenarios. Additionally, in comparison with other algorithms tested in this thesis, the classification rate of the UC pattern in all scenarios gave the HMM high credibility in terms of reliability. The hybrid method proved that the HMM classification rate could be enhanced when malfunctioning sensors not more than 25% of all sensors. However, the processing time of the HMM could be an issue if the number of sensors inside buildings increased.

6.3 Future Work

A novel approach to building collapse pattern classification was examined in this thesis. This opens a new research area that needs further investigation to advance knowledge. Creating a prototype building and using real sensors could be one of the next stages of this research. A comparison could be drawn between simulated results and prototype results and gives more validity to this approach. In addition, the hybrid

method could be further investigated by using various combinations of classification algorithms to achieve better results. Finally, examining various collapse scenarios for the same collapse pattern could also lead to the investigation of unsupervised algorithms in order to increase the number of subclasses in the same class and gain more reliability.

REFERENCES

- [1]. C. Ozdel. (2011). *Turkey earthquake: building negligence amounts to murder, says Erdogan* [online]. Available:
<http://www.guardian.co.uk/world/2011/oct/26/turkey-earthquake-building-negligence-erdogan> [March 20, 2013]
- [2]. A. Mehrabian and A. Haldar, "Some lessons learned from post-earthquake damage survey of structures in Bam, Iran earthquake of 2003", *Structural Survey*, vol. 23, no. 3, pp.180-192, 2005.
- [3]. Y. Wang, "Earthquake resistance present situation and suggestion of the rural buildings," *2nd Int. Conf. on Mechanic Automation and Control Engineering (MACE)*, 2011, pp.2103-2106.
- [4]. S. Tadokoro, T. Takamori, K. Osuka, S. Tsurutani, "Investigation report of the rescue problem at Hanshin-Awaji earthquake in Kobe," *IEEE/RSJ In. Conf. on Intelligent Robots and Systems (IROS)*, 2000, pp.1880-1885.
- [5]. W. Ze-gen, Y. Yan-mei, X. Jun-nan, "Study on the Earthquake Cooperative Emergency Rescue Auxiliary System," *WiCom '09. 5th Int. Conf. on Wireless Communications, Networking and Mobile Computing*, 2009, pp.1-5.
- [6]. D. S. Ignatova, R. K. Oransky, "Mechatronic approach to investigations of rescue operations," *14th Int. Symp. MECHATRONIKA*, 2011, pp.103-108.
- [7]. J. Ma, S. Qin, "Automatic depicting algorithm of earthquake collapsed buildings with airborne high resolution image," *IEEE Int. On Geoscience and Remote Sensing Symposium (IGARSS)*, 2012, pp.939-942.
- [8]. D. Brunner, G. Lemoine, L. Bruzzone, "Earthquake Damage Assessment of Buildings Using VHR Optical and SAR Imagery," *IEEE Trans. on Geoscience and Remote Sensing*, 2010, pp.2403-2420.
- [9]. F. Pacifici, M. Chini, C. Bignami, S. Stramondo, W. Emery, "Automatic damage detection Using pulse-coupled neural networks For the 2009 Italian earthquake," *IEEE Int. Conf. on Geoscience and Remote Sensing Symposium (IGARSS)*, 2010, pp.1996-1999.
- [10]. M. Hosokawa, J. Byeong-Pyo, O. Takizawa, "Earthquake intensity estimation and damage detection using remote sensing data for global rescue operations," *IEEE Int. Geoscience and Remote Sensing Symp. IGARSSI*, 2009, pp.420-423.
- [11]. L. Li, Z. Li, R. Zhang, J. Ma; L. Lei, "Collapsed buildings extraction using morphological profiles and texture statistics - A case study in the 5.12 wenchuan earthquake," *IEEE Int. Conf. on Geoscience and Remote Sensing Symp. (IGARSS)*, 2010, pp.2000-2002.

- [12]. D. Xue, Z. He, S. Tao, "High-resolution remote sensing image disaster emergency investigation and analysis," *Int. Conf. on Remote Sensing, Environment and Transportation Engineering (RSETE)*, 2011, pp.3195-3197.
- [13]. L. Hengjian, M. Kohiyama, K. Horie, N. Maki, H. Hayashi, and S. Tanaka, "Building Damage and Casualties after an Earthquake," *Natural Hazards*, vol 29, no. 3, pp 387-403, July 2003.
- [14]. F. Matsuno, S. Tadokoro, "Rescue Robots and Systems in Japan," *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2004, pp.12-20.
- [15]. K. Harihara, S. Dohta, T. Akagi, F. Zhang, "Development of a search type rescue robot driven by pneumatic actuator," *Proc. of SICE Ann. Conf.*, 2010, pp.1311-1317.
- [16]. U. Starossek, "Typology of progressive collapse," *Engineering Structures*, vol. 29, no. 9, pp 2302-2307, September 2007.
- [17]. J. Tsuru and S Murkami, "The 2011 of the Pacific coast of Tohoku Earthquake," Building Research Institute, Japan , Tech. Rep., 2011, [May 5, 2013].
- [18]. M. Bavusi, A. Loperte, V. LaPenna, F. Soldovieri, "Rebars and defects detection by a GPR survey at a L'Aquila school damaged by the earthquake of April 2009," *13th Int. Conf. on Ground Penetrating Radar (GPR)*, 2010, pp.1-5.
- [19]. Q. Zhou, K. Yang, "Progressive Collapse of a Chinese Ancient Building by Simulation," *3rd Int. Conf. on Information and Computing (ICIC)*, 2010, pp.192-194.
- [20]. W. Tie-cheng, L. Zhi-ping, "Progressive collapse simulation of concrete frame structures with floors," *Int. Conf. on Consumer Electronics, Communications and Networks (CECNet)*, 2011, pp.3372-3376.
- [21]. K. Kotani, C. Qiu, T. Ohmi, "Face recognition using vector quantization histogram method," *Proc. Image Processing*, 2002, pp.105-108.
- [22]. A. V. Nefian, M. H. Hayes, "Face detection and recognition using hidden Markov models," *Proc. Image Processing ICIP*, 1998, pp.141-145.
- [23]. T. Ke, Z. Zhang, Y. Zhang, "The Parallel Processing of DPGrid in the Fast Response for China 5.12 Wenchuan Earthquake Rescue," *Int. Conf. on Computational Intelligence and Software Engineering CiSE*, 2009, pp.1-5.
- [24]. E. Sumer, M. Turker, "Building damage detection from post-earthquake aerial imagery using building grey-value and gradient orientation analyses," *Proc. of 2nd Int. Conf. on Recent Advances in Space Technologies RAST*, 2005, pp.577-582.

- [25]. S. Grattan, S. Taylor, P. Basheer, T. Sun, K. Grattan, "Structural health monitoring - better solutions using fiber optic sensors," *Sensors, 2009 IEEE*, pp.811-814, Oct. 2009.
- [26]. J. Paek, K. Chintalapudi, R. Govindan, J. Caffrey, S. Masri, "A Wireless Sensor Network for Structural Health Monitoring: Performance and Experience," *The 2nd IEEE Workshop on Embedded Networked Sensors EmNetS-II*, 2005, pp.1-10.
- [27]. R. Morello, C. De Capua, A. Meduri, "Remote monitoring of building structural integrity by a smart wireless sensor network," *Instrumentation and Measurement Technology Conference (I2MTC)*, 2010, pp.1150-1154.
- [28]. K. Chintalapudi, J. Paek, N. Kothari, S. Rangwala, R. Govindan, E. Johnson, "Embedded sensing of structures: a reality check," *Proc. 11th IEEE Int. Conf. on Embedded and Real-Time Computing Systems and Applications*, 2005, pp.95-101.
- [29]. L. Si, Z. Chen, X. Yu, Q. Zhang, X. Liu, "Embedded real-time damage detection and identification algorithms in wireless health monitoring system for smart structures," *IEEE Int. Conf. on Communications Technology and Applications, ICCTA*, 2009, pp.676-681.
- [30]. A. Mahajan, K. Wang, P. Ray, "Multisensor integration and fusion model that uses a fuzzy inference system," *IEEE/ASME Trans. on Mechatronics*, 2001, pp.188-196.
- [31]. P. Sanguansat, "Two-Dimensional Principal Component Analysis and Its Extensions," in *Principal Component Analysis*, 1st ed. P. Sanguansat, Rijeka, Coratia: In Tech, 2012, pp. 1-23.
- [32]. J. I. Jolliffe, *Principal Component Analysis*, 2nd ed. London, Springer, 2002.
- [33]. L.I. Kuncheva and W. Faithfull, "PCA Feature Extraction for Change Detection in Multidimensional Unlabelled Data", *IEEE Trans. On Neural Networks and Learning Systems*, vol. 25, no. 1, pp.69-80, 2014.
- [34]. A. Martinez and A. Kak, "PCA versus LDA", *IEEE Trans. On Pattern Analysis and Machine intelligence*, vol. 23, no. 2, pp. 228-233, 2001.
- [35]. X. Huan, C. Caramanis, S. Sanghavi, "Robust PCA via Outlier Pursuit", *IEEE Trans. On Information Theory*, vol. 58, no. 5, pp. 3047-3064, 2012.
- [36]. R. Good, D. Kost, G. Cherry, "Introducing a Unified PCA Algorithm for Model Size Reduction", *IEEE Trans. On Semiconductor Manufacturing*, vol. 23, no. 2, pp. 201-209, 2010.
- [37]. Y. Jian, D. Zhang, A. Frangi, Y. Jing-Yu, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131-137, 2004.

- [38]. A. Rajagopalan, R. Chellappa, N. Koterba, "Background learning for robust face recognition with PCA in the presence of clutter", *IEEE Trans. On Image Processing*, vol. 14, no. 4, pp. 832-843, 2005.
- [39]. M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*, 1st ed. Washington, SPIE, 1991.
- [40]. V. Cherkassky and F. Mulier, *Learning from Data*, 2nd ed. New Jersey, Wiley, 2007.
- [41]. T. Kinnunen, E. Karpov, P. Franti, "Real-time speaker identification and verification", *IEEE Trans. On Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 277-288, 2006.
- [42]. A. Iosifidis, A. Tefas, I. Pitas, "Minimum Class Variance Extreme Learning Machine for Human Action Recognition", *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1968-1979, 2013.
- [43]. B. Zhang, Y. Zhou, H. Pan, "Vehicle Classification with Confidence by Classified Vector Quantization", *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 8-20, 2013.
- [44]. F. Yu, B. Liu, Z. Lu, "Fast image colourisation based on twin-codebook vector quantisation", *Electronics Letters*, vol. 46, no. 2, pp. 132-134, 2010.
- [45]. A. Vasuki and P. Vanathi, "A review of vector quantization techniques", *IEEE Potentials*, vol. 25, no. 4, pp. 39-47, 2006.
- [46]. S. Chen and F. Li, "Fast codebook design of vector quantisation", *Electronics Letters*, vol. 48, no. 15, pp. 921-922, 2012.
- [47]. S. Chen and F. Li, "Initial codebook method of vector quantisation in Hadamard domain", *Electronics Letters*, vol. 46, no. 9, pp. 630-631, 2010.
- [48]. Y. Shum and D. Lun, "Vector quantisation codebook initialisation for discrete multiwavelet transform", *Electronics Letters*, vol. 43, no. 22, 2007.
- [49]. J. Hao, T. Shibata, "A Vlsi-Implementation-Friendly EGO-Motion Detection Algorithm Based on Edge-Histogram Matching," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing ICASSP*, 2006, pp.245-248.
- [50]. L. Satish, B. Gururaj, "Use of hidden Markov models for partial discharge pattern classification," *IEEE Trans. on Electrical Insulation*, 1993, pp.172-182.
- [51]. A. Abushariah, T. Gunawan, O. Khalifa, M. Abushariah, "English digits speech recognition system based on Hidden Markov Models," *Int. Conf. on Computer and Communication Engineering (ICCCE)*, 2010, pp.1-5.
- [52]. W. Wan, H. Liu, G. Shi, W. Li, "Real-time Recognition of Multi-category Human Motion Using μ IMU Data," *Int. Conf. on Mechatronics and Automation ICMA*, 2007, pp.1845-1850.

- [53]. G. L. Kouemou, "History and Theoretical Basic of Hidden Markov Models," in *Hidden Markov Models, Theory and Applications*, 1st ed. Croatia, InTech, 2011.
- [54]. G. Panahandeh, N. Mohammadiha, A. Leijon, P. Handel, "Continuous Hidden Markov Model for Pedestrian Activity Classification and Gait Analysis", *IEEE Trans. On Instrumentation and Measurement*, vol. 62, no. 5, pp. 1073-1083, 2013.
- [55]. M. Xiang, D. Schonfeld, A. Khokhar, "Video Event Classification and Image Segmentation Based on Noncausal Multidimensional Hidden Markov Models", *IEEE Trans. On Image Processing*, vol. 18, no. 6, pp. 1304-1313, 2009.
- [56]. Z. Hui, Q. Wu, T. Nguyen, "Modified student's t-hidden Markov model for pattern recognition and classification", *IET on Signal Processing*, vol. 7, no. 3, pp. 219-227, 2013.
- [57]. J. Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and Hidden Markov Models," Tech. Rep., International Computer Science Institute, 1997 [November 5, 2013].
- [58]. M. Oudelha, R. Ainon, "HMM parameters estimation using hybrid Baum-Welch genetic algorithm," *Int. Symp. in Information Technology (ITSim)*, 2010 , pp.542-545.
- [59]. Y. Bar-Shalom, X. LI, *Estimation and Tracking: Principles, Techniques and Software*. Boston: Artech House, 1993.
- [60]. L. Baum, T. Petrie, G. Soules, N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164-171, (Feb., 1970).
- [61]. R. Hsiao, T. Yik-Cheung, T. Schultz, "Generalized Baum-Welch algorithm for discriminative training on large vocabulary continuous speech recognition system," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing ICASSP* , 2009, pp.3769-3772.
- [62]. C. Qiu, H. Ren, H. Zou, S. Zhou, "Performance comparison of target classification in SAR images based on PCA and 2D-PCA features," *2nd Asian-Pacific Conf. on Synthetic Aperture Radar APSAR*, 2009, pp.868-871.
- [63]. A. Banthia, A. Jayasumana, Y. Malaiya, "Data size reduction for clustering-based binning of ICs using principal component analysis (PCA)," *Proc. IEEE Int. Workshop on Current and Defect Based Testing DBT*, 2005, pp.24-30.
- [64]. T. Wu, C. Su, "Application of Principal Component Analysis and Clustering to Spatial Allocation of Groundwater Contamination," *5th Int. Conf. on Fuzzy Systems and Knowledge Discovery FSKD*, 2008, pp.236-240.

- [65]. T. Wu, C. Huang, "Application of Principal Component Analysis on Source Identification and Tracking of Groundwater Contamination from Southern Taiwan Science Park," *6th Int. Conf. on Fuzzy Systems and Knowledge Discovery FSKD*, 2009, pp.527-531.
- [66]. X. Xiong, Y. Kim, Y. Baek, D. Rhee, S. Kim, "Analysis of breast cancer using data mining & statistical techniques," *6th Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, and First ACIS Int. Workshop on Self-Assembling Wireless Networks SNPD/SAWN*, 2005, pp.82-87.
- [67]. I. Wijaya, K. Uchimura, H. Zhencheng, "Why the alternative PCA provides better performance for face recognition," *10th Workshop on Image Analysis for Multimedia Interactive Services WIAMIS*, 2009, pp.149-152.
- [68]. Y. Tang, C. Cai, F. Zhao, "Wood Identification Based on PCA, 2DPCA and (2D)2PCA," *5th Int. Conf. on Image and Graphics ICIG*, 2009, pp.784-789.
- [69]. Z. Cui, H. Han, R. Park, "Image quality assessment using a vector quantization histogram," *IEEE 13th Int. Symp. on Consumer Electronics ISCE*, 2009, pp.294-297.
- [70]. X. Feng, P. Perona, "Human action recognition by sequence of movelet codewords," *1st Int. Symp. on 3D Data Processing Visualization and Transmission*, 2002, pp.717-721.
- [71]. S. Foo, L. Dong, "A boosted multi-HMM classifier for recognition of visual speech elements," *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP)*, 2003, pp.II,285-8.
- [72]. K. Hara, T. Omori, R. Ueno, "Detection of unusual human behavior in intelligent house," *Proc. of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing*, 2002, pp.697-706.
- [73]. M. Ahmad, S. Lee, "HMM-based Human Action Recognition Using Multiview Image Sequences," *18th Int. Conf. on Pattern Recognition ICPR*, 2006, pp.263-266.
- [74]. W. Wan, H. Liu, L. Wang, G. Shi, W. Li, "A hybrid HMM/SVM classifier for motion recognition using μ IMU data," *IEEE Int. Conf. on Robotics and Biomimetics ROBIO*, 2007, pp.115-120.
- [75]. B. Adams and et al., "The Bam (Iran) Earthquake of December 26, 2003: Preliminary Reconnaissance Using Remotely Sensed Data and the VIEWS (Visualizing the Impacts of Earthquakes with Satellite Images) System," MCEER Earthq. Eng. Res. Cent., Univ. New York, New York, Inv. Rep. 2003.
- [76]. S. Chang and et al., "Preliminary Reports from the Hyogo-ken Nambu Earthquake of January 17, 1995," NCREE Earthq. Eng. Res. Cent., Primary Report, Univ. New York, New York, 1995.

- [77]. G. Lee, and C. Loh, "Preliminary Report from MCEER-NCREE," Workshop on the 921 Taiwan Earthquake," MCEER-NCREE Earthq. Eng. Res. Cent., Primary Report, Univ. New York, New York, 1999.
- [78]. D. Kusumastuti and et al., "Overview of the West Sumatra Earthquake of September 30, 2009," MCEER Earthq. Eng. Res. Cent., Summary Report, Univ. New York, New York, 2009.
- [79]. FEMA Org. (2008, Feb.). Student Manual Structure Collapse Awareness. [Online]. Available: <http://http://disasterengineer.org/LinkClick.aspx?fileticket=gKEYfpXzYxo%3D&...> [February 6, 2013].
- [80]. T. Kruthammer. (2009, Jan. second). Prevent collapse. Magazine Features. Available: <http://www.metalarchitecture.com/articles/magazine-features/prevent-collapse-finite-element-analysis-software-could-augment-construction-guidelines-for-steel-structures.aspx> [March 17, 2013]
- [81]. L. Allen. (2013, Jun. 12). Hybrid lateral load-resisting system [LH]. Available: <http://www.nexus.globalquakemodel.org/gem-building-taxonomy/overview/glossary/hybrid-lateral-load-resisting-system--lh>. [March 17, 2013]
- [82]. C. Scawthorn, "Preliminary Report Kocaeli (Izmit) Earthquake of 17 August 1999" MCEER Earthq. Eng. Res. Cent., Summary Report, Univ. New York, New York, 1999.
- [83]. M. Patoli, M. Gkion, A. Al-Barakati, W. Zhang, P. Newbury, M. White, "An open source Grid based render farm for Blender 3D," *Power Systems Conference and Exposition PES*, 2009, pp.1-6 .
- [84]. A. Nakahata, H. Itoh, H. Fukumoto, H. Wakuya, T. Furukawa, "A natural-language-based 3D-CG system with shape estimation for unknown words," *Proc. of SICE Ann. Conf.*, 2011, pp.2847-2850.
- [85]. T. Yu, "Chess Gaming and Graphics Using Open-Source Tools," *Int. Conf. on Computing, Engineering and Information ICC*, 2009, pp.253-256.
- [86]. N. Rowe, "The Visualization and Animation of Algorithmically Generated 3D Models Using Open Source Software," *1st Int. Conf. on Advances in Multimedia MMEDIA*, 2009, pp.44-49.
- [87]. R. Smits, H. Bruyninckx, "Composition of complex robot applications via data flow integration," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp.5576-5580.
- [88]. A. Kitanov, S. Bisevac, I. Petrovic, "Mobile robot self-localization in complex indoor environments using monocular vision and 3D model," *IEEE/ASME int. Conf. on Advanced intelligent mechatronics*, 2007, pp.1-6.

- [89]. G. Echeverria, N. Lassabe, A. Degroote, S. Lemaignan, “Modular open robots simulation engine: MORSE,” *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp.46-51.
- [90]. Standards Australia. “Structural design action” Aus. Standard 1170.4-1993, Oct. 9, 2007.

APPENDIX A

Consequences of Failure for Importance Levels [74]

Consequences of Failure	Description	Importance level	Comment
Low	Low consequence for loss of human life, or small or moderate economic, social or environmental consequences	1	Minor structures (failure not likely to endanger human life)
Ordinary	Medium consequence for loss of human life, or considerable economic, social or environmental consequences	2	Normal structure and structures not falling into other levels
High	High consequence for loss of human life, or very great economic, social or environmental consequences	3	Major structures
		4	Post-disaster structure
Exceptional	Circumstances where reliability must be set on a case by case basis	5	Exceptional structures

Annual Probability of Exceedance [74]

Design working life	Importance level	Annual probability of exceedance for ultimate limit states			Annual probability of exceedance for serviceability limit states	
		Wind	Snow	Earthquake	SLS1	SLS2 importance level 4 only
Construction equipment	2	1/100	1/50	1/100	1/25	
Less than 6 months	1	1/25	1/25	1/25	-	
	2	1/100	1/50	1/100	1/25	
	3	1/250	1/100	1/250	1/25	
	4	1/1000	1/250	1/1000	1/25	
5 years	1	1/25	1/25	1/25	-	-
	2	1/100	1/50	1/100	1/25	-
	3	1/250	1/100	1/250	1/25	-
	4	1/1000	1/250	1/1000	1/25	1/250
25 years	1	1/25	1/25	1/25	-	-
	2	1/100	1/50	1/100	1/25	-
	3	1/250	1/100	1/250	1/25	-
	4	1/1000	1/250	1/1000	1/25	1/250
50 years	1	1/100	1/50	1/100	-	-
	2	1/500	1/150	1/500	1/25	-
	3	1/1000	1/250	1/1000	1/25	-
	4	1/2500	1/500	1/2500	1/25	1/500
100 years or more	1	1/250	1/150	1/250	-	-
	2	1/1000	1/250	1/1000	1/25	-
	3	1/2500	1/500	1/2500	1/25	-
	4	*	*	*	1/25	*

* for importance level 4 structures with a design working life of 100 years or more, the design events are determined by a hazard analysis but need to have probabilities less than or equal to those for importance level 3. Design events for importance level 5 structures should be determined on a case by case basis.

Importance Levels for Building Type- New Zealand Structures [74]

Importance level	Comment	Example
1	Structures presenting a low degree of hazard to life and other property	Structures with a total floor area of $< 30 \text{ m}^2$ Farm buildings, isolated structures, towers in rural situations Fences, masts, walls, in-ground swimming pools
2	Normal structures and structures not in other importance levels	Buildings not included in importance level 1,3, or 4 Single family dwellings Car parking buildings
3	Structures that as a whole may contain people in crowds or contents of high value to the community or pose risks to people in crowds	Buildings and facilities as follows: a) where more than 300 people can congregate in one area b) day care facilities with a capacity greater than 150 c) primary school or secondary school facilities with a capacity greater than 250 d) colleges or adult education facilities with a capacity greater than 500 e) health care facilities with a capacity of 50 or more resident patients but not having surgery or emergency treatment facilities f) airport terminals, principal railway stations with a capacity greater than 250 g) correctional institutions h) multi-occupancy residential, commercial, industrial, office and retailing buildings designed to accommodate more than 5000 people and with a gross area greater than 10000 m^2 i) public assembly buildings, theatres and cinemas of greater than 10000 m^2 emergency medical and other emergency facilities not designated as post-disaster Power-generating facilities, water treatment and waste water treatment facilities and other public utilities not designated as post-disaster Building and facilities not designated as post-disaster containing hazardous materials capable of causing hazardous conditions that do not extend beyond the property boundaries
4	Structures with special post-disaster functions	Buildings and facilities designated as essential facilities Buildings and facilities with special post-disaster function Medical emergency or surgical facilities Emergency service facilities such as fire, police stations and emergency vehicle garages Utilities or emergency shelters, designated emergency centres and ancillary facilities Buildings and facilities containing hazardous materials capable of causing hazardous conditions that extend beyond the property boundaries
5	Special structures	Structures that have special functions of whose failure poses catastrophic risk to a large area

Probability Factor (K_p) [74]

Annual probability of exceedance P	Probability factor K_p
1/2500	1.8
1/2000	1.7
1/1500	1.5
1/1000	1.3
1/800	1.25
1/500	1.0
1/250	0.75
1/200	0.7
1/100	0.5
1/50	0.35
1/25	0.25
1/20	0.20

Maximum Depth limits for Site Sub-Soil Class C [74]

Soil type and description		Property		Maximum depth of soil
		Representative undrained shear strengths (KPa)	Representative SPT N -values (Number)	(m)
Cohesive soils	Very soft	<12.5	-	0
	Soft	12.5-25	-	20
	Firm	25-50	-	25
	Stiff	50-100	-	40
	Very stiff or hard	100-200	-	60
Cohesion less soils	Very loose	-	<6	0
	Loose dry	-	6-10	40
	Medium dense	-	10-30	45
	Dense	-	30-50	55
	Very dense	-	>50	60
	Gravels	-	>30	100

Selection of Earthquake Design Categories [74]

Importance level, type of structure	$(K_p Z)$ for site sub-soil class				Structure height, $h_n(\text{m})$	Earthquake design category
	E_c or D_c	C_c	B_c	A_c		
1	-				-	Not required to be designed for earthquake actions
Domestic structure (housing)	-				Top of roof <8.5	
					Top of roof >8.5	Design as importance level 2
2	<0.05	<0.08	<0.11	<0.14	<12 >12, <50 >50	I II III
	>0.05- <0.08	>0.08- <0.12	>0.11- <0.17	>0.14- <0.21	<50 >50	II III
	>0.08	>0.12	>0.17	>0.21	<25 >25	II III
3	<0.08	<0.12	<0.17	<0.21	<50 >50	II III
	>0.08	>0.12	>0.17	>0.21	<25 >25	II III
4	-				<12 >12	II III

Values of K_s for Structures not Exceeding 15m. [74]

Total number of stories	Sub-soil class	K_s factor				
		Storey under consideration				
		5 th	4 th	3 rd	2 nd	1 st
5	A_e	2.5	1.9	1.4	1.0	0.5
	B_e	3.1	2.5	1.8	1.2	0.6
	C_e	4.4	3.5	2.6	1.7	0.9
	D_e, E_e	6.1	4.9	3.6	2.5	1.2
4	A_e	-	2.7	2.0	1.4	0.6
	B_e	-	3.5	2.6	1.7	0.9
	C_e	-	4.9	3.6	2.5	1.2
	D_e, E_e	-	5.8	4.4	3.0	1.4
3	A_e	-	-	3.1	2.0	1.0
	B_e	-	-	3.9	2.6	1.3
	C_e, D_e, E_e	-	-	5.5	3.6	1.8
2	A_e	-	-	-	3.1	1.6
	B_e	-	-	-	3.9	1.9
	C_e, D_e, E_e	-	-	-	4.9	2.5
1	A_e	-	-	-	-	2.3
	B_e	-	-	-	-	3.0
	C_e, D_e, E_e	-	-	-	-	3.6

Structural Ductility Factor (μ) and Structural Performance Factor (S_p)- Basic Structures [74]

Structure system	Description	μ	S_p	S_p/μ	μ/S_p
Steel structure					
	Special moment-resisting frames (fully ductile)*	4	0.67	0.17	6
	Intermediate moment-resisting frames	3	0.67	0.22	4.5
	Ordinary moment-resisting frames	2	0.77	0.38	2.6
	Moderately ductile concentrically braced frames	3	0.67	0.22	4.5
	Limited ductile concentrically braced frames	2	0.77	0.38	2.6
	Fully ductile eccentrically braced frames*	4	0.67	0.17	6
	Other steel structure not defined above	2	0.77	0.38	2.6
Concrete structure					
	Special moment-resisting frames*	4	0.67	0.17	6
	Intermediate moment-resisting frames	3	0.67	0.22	4.5
	Ordinary moment-resisting frames	2	0.77	0.38	2.6
	Ductile coupled walls*	4	0.67	0.17	6
	Ductile partially coupled wall*	4	0.67	0.17	6
	Ductile shear walls	3	0.67	0.22	4.5
	Limited ductile shear walls	2	0.77	0.38	2.6
	Ordinary moment-resisting frames in combination with a limited ductile shear walls	2	0.77	0.38	2.6
	Other concrete structures not listed above	2	0.77	0.38	2.6
Timber structures					
	Shear walls	3	0.67	0.22	4.5
	Braced frames	2	0.77	0.38	2.6
	Moment-resisting frames	2	0.77	0.38	2.6
	Other wood or gypsum based seismic-force-resisting systems not listed above	2	0.77	0.38	2.6
Masonry structures					
	Close-spaced reinforced masonry	2	0.77	0.38	2.6
	Wide-spaced reinforced masonry	1.5	0.77	0.5	2
	Unreinforced masonry	1.25	0.77	0.62	1.6
	Other masonry structures not complying with AS 3700	1.00	0.77	0.77	1.3

* The design of structures with $\mu > 3$ is outside the scope of this standard.

Spectral Shape Factor ($C_h(T)$) [74]

Period (seconds)	Site sub-soil class				
	A_e Strong rock	B_e Rock	C_e Shallow soil	D_e Deep or soft soil	E_e Very soft soil
0.0	2.35(0.8)*	2.94(1.0)*	3.68(1.3)*	3.68(1.1)*	3.68(1.1)*
0.1	2.35	2.94	3.68	3.68	3.68
0.2	2.35	2.94	3.68	3.68	3.68
0.3	2.35	2.94	3.68	3.68	3.68
0.4	1.76	2.2	3.12	3.68	3.68
0.5	1.21	1.76	2.5	3.68	3.68
0.6	1.17	1.47	2.08	3.3	3.68
0.7	1.01	1.26	1.79	2.83	3.68
0.8	0.88	1.1	1.56	2.48	3.68
0.9	0.78	0.98	1.39	2.2	3.42
1.0	0.70	0.88	1.25	1.98	3.08
1.2	0.59	0.73	1.04	1.65	2.57
1.5	0.47	0.59	0.83	1.32	2.05
1.7	0.37	0.46	0.65	1.03	1.6
2.0	0.26	0.33	0.47	0.74	1.16
2.5	0.17	0.21	0.3	0.48	0.74
3.0	0.12	0.15	0.21	0.33	0.51
3.5	0.086	0.11	0.15	0.24	0.38
4.0	0.066	0.083	0.12	0.19	0.29
4.5	0.052	0.065	0.093	0.15	0.23
5.0	0.042	0.053	0.075	0.12	0.18

* Values in brackets correspond to values of spectral shape factor for the modal response spectrum and the numerical integration time history methods and for use in the method of calculation of forces on parts and components.

APPENDIX B

Basic PCA (2 sensors failure)							
				PCA MIoA case			
Overall classification rate				60%			
First column (FC)				4%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				30%			
Uncollapsed (UC)				66%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	2	0	33	0	0	15	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	25	0	15	5	5	50
UC	0	2	0	15	33	0	50

Basic PCA (3 sensors failure)							
				PCA MIoA case			
Overall classification rate				56.8%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				18%			
Uncollapsed (UC)				66%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	26	0	9	5	10	50
UC	0	2	0	15	33	0	50

Basic PCA (4 sensors failure)							
				PCA MIoA case			
Overall classification rate				53.2%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				0%			
Uncollapsed (UC)				66%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	3	0	0	47	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	4	0	0	0	46	50
UC	0	2	0	15	33	0	50

Basic PCA (5 sensors failure)							
				PCA MIoA case			
Overall classification rate				54%			
First column (FC)				4%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				0%			
Uncollapsed (UC)				54%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	2	0	1	0	0	47	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	0	0	50	50
UC	0	2	0	15	33	0	50

Basic PCA (6 sensors failure)							
				PCA MIoA case			
Overall classification rate				57.2%			
First column (FC)				4%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				0%			
Uncollapsed (UC)				82%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	2	0	1	0	0	47	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	0	0	50	50
UC	0	2	0	7	41	0	50

Basic PCA (1 sensor failure)							
				PCA LIoA case			
Overall classification rate				74.8%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				74%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	13	37	0	50

Basic PCA (2 sensors failure)							
				PCA LIoA case			
Overall classification rate				74.8%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				74%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	13	37	0	50

Basic PCA (3 sensors failure)							
				PCA LIoA case			
Overall classification rate				74.8%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				74%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	13	37	0	50

Basic PCA (4 sensors failure)							
				PCA LIoA case			
Overall classification rate				74.8%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				74%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	13	37	0	50

Basic PCA (5 sensors failure)							
				PCA LIoA case			
Overall classification rate				74.8%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				74%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	2	0	11	37	0	50

Basic PCA (6 sensors failure)							
				PCA LIoA case			
Overall classification rate				74.8%			
First column (FC)				0%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				74%			
Processing time in seconds				200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	33	0	0	17	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	2	0	11	37	0	50

Sensor-by-sensor based PCA (1 sensor failure)							
				PCA MLoA case			
Overall classification rate				99.2%			
First column (FC)				100%			
First storey (FS)				96%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	48	0	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (2 sensors failure)							
				PCA MIoA case			
Overall classification rate				78.8%			
First column (FC)				0%			
First storey (FS)				94%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	0	50	50
FS	0	47	1	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (3 sensors failure)							
				PCA MIoA case			
Overall classification rate				78.8%			
First column (FC)				0%			
First storey (FS)				94%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	0	50	50
FS	0	47	1	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (4 sensors failure)							
				PCA MIoA case			
Overall classification rate				58.8%			
First column (FC)				0%			
First storey (FS)				94%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				0%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	0	50	50
FS	0	47	1	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	0	50	50

Sensor-by-sensor based PCA (5 sensors failure)							
				PCA MIoA case			
Overall classification rate				58.8%			
First column (FC)				0%			
First storey (FS)				94%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				0%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	0	50	50
FS	0	47	1	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	0	50	50

Sensor-by-sensor based PCA (6 sensors failure)							
				PCA MIoA case			
Overall classification rate				58.8%			
First column (FC)				0%			
First storey (FS)				94%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				0%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	0	50	50
FS	0	47	1	2	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	0	50	50

Sensor-by-sensor based PCA (1 sensors failure)							
				PCA LIoA case			
Overall classification rate				99.6%			
First column (FC)				100%			
First storey (FS)				98%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	1	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (2 sensors failure)							
				PCA LIoA case			
Overall classification rate				99.6%			
First column (FC)				100%			
First storey (FS)				98%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	1	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (3 sensors failure)							
				PCA LIoA case			
Overall classification rate				99.6%			
First column (FC)				100%			
First storey (FS)				98%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	1	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (4 sensors failure)							
				PCA LIoA case			
Overall classification rate				99.6%			
First column (FC)				100%			
First storey (FS)				98%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	1	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (5 sensors failure)							
				PCA LIoA case			
Overall classification rate				99.6%			
First column (FC)				100%			
First storey (FS)				98%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	1	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Sensor-by-sensor based PCA (6 sensors failure)							
				PCA LIoA case			
Overall classification rate				99.6%			
First column (FC)				100%			
First storey (FS)				98%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				211/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	49	1	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

VQ (1 sensors failure)							
				VQ MIoA case			
Overall classification rate				89.6%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				90%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	6	0	2	0	50
MS	2	3	45	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (2 sensors failure)							
				VQ MIoA case			
Overall classification rate				89.6%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				90%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	6	0	2	0	50
MS	2	3	45	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (3 sensors failure)							
				VQ MIoA case			
Overall classification rate				88.8%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				86%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	5	0	3	0	50
MS	6	1	43	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (4 sensors failure)							
				VQ MIoA case			
Overall classification rate				88.8%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				86%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	5	0	3	0	50
MS	6	1	43	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (5 sensors failure)							
				VQ MIoA case			
Overall classification rate				86.8%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				76%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	6	0	2	0	50
MS	11	1	38	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (6 sensors failure)							
				VQ MIoA case			
Overall classification rate				87.2%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				80%			
Pancake (PCK)				98%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	7	0	1	0	50
MS	9	1	40	0	0	0	50
PCK	0	0	1	49	0	0	50
UC	12	0	0	0	38	0	50

VQ (1 sensors failure)							
				VQ LIoA case			
Overall classification rate				90.4%			
First column (FC)				100%			
First storey (FS)				84%			
Mid-storey (MS)				92%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	42	6	0	1	0	50
MS	1	3	46	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (2 sensors failure)							
				VQ LIoA case			
Overall classification rate				90.4%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				94%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	5	0	3	0	50
MS	1	2	47	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (3 sensors failure)							
				VQ LIoA case			
Overall classification rate				90%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				92%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	7	0	1	0	50
MS	1	3	46	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (4 sensors failure)							
				VQ LIoA case			
Overall classification rate				90%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				92%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	7	0	1	0	50
MS	1	3	46	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (5 sensors failure)							
				VQ LIoA case			
Overall classification rate				89.6%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				90%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	6	0	2	0	50
MS	4	1	45	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (6 sensors failure)							
				VQ LIoA case			
Overall classification rate				88.8%			
First column (FC)				100%			
First storey (FS)				82%			
Mid-storey (MS)				86%			
Pancake (PCK)				100%			
Uncollapsed (UC)				76%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	41	6	0	2	0	50
MS	6	1	43	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	12	0	0	0	38	0	50

VQ (1st Column failure)							
				Results			
Overall classification rate				76%			
First column (FC)				100%			
First storey (FS)				72%			
Mid-storey (MS)				72%			
Pancake (PCK)				98%			
Uncollapsed (UC)				38%			
Processing time in seconds				290/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	2	36	12	0	0	0	50
MS	14	0	36	0	0	0	50
PCK	0	0	1	49	0	0	50
UC	31	0	0	0	19	0	50

VQ (2nd Column failure)							
						Results	
Overall classification rate						80.4%	
First column (FC)						100%	
First storey (FS)						72%	
Mid-storey (MS)						66%	
Pancake (PCK)						98%	
Uncollapsed (UC)						66%	
Processing time in seconds						290/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	5	36	9	0	0	0	50
MS	17	0	33	0	0	0	50
PCK	0	0	1	49	0	0	50
UC	17	0	0	0	33	0	50

VQ (3d Column failure)							
					Results		
Overall classification rate					77.2%		
First column (FC)					100%		
First storey (FS)					72%		
Mid-storey (MS)					50%		
Pancake (PCK)					98%		
Uncollapsed (UC)					66%		
Processing time in seconds					290/250		
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	3	36	11	0	0	0	50
MS	25	0	25	0	0	0	50
PCK	0	0	1	49	0	0	50
UC	17	0	0	0	33	0	50

VQ (4th Column failure)							
						Results	
Overall classification rate						77.6%	
First column (FC)						100%	
First storey (FS)						66%	
Mid-storey (MS)						72%	
Pancake (PCK)						94%	
Uncollapsed (UC)						56%	
Processing time in seconds						290/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	4	33	13	0	0	0	50
MS	14	0	36	0	0	0	50
PCK	0	0	3	47	0	0	50
UC	22	0	0	0	28	0	50

VQ (1st Row failure)							
						Results	
Overall classification rate						74.8%	
First column (FC)						100%	
First storey (FS)						52%	
Mid-storey (MS)						94%	
Pancake (PCK)						52%	
Uncollapsed (UC)						76%	
Processing time in seconds						290/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	10	26	14	0	0	0	50
MS	1	2	47	0	0	0	50
PCK	0	0	24	26	0	0	50
UC	12	0	0	0	38	0	50

VQ (2nd Row failure)							
						Results	
Overall classification rate						36.4%	
First column (FC)						100%	
First storey (FS)						54%	
Mid-storey (MS)						0%	
Pancake (PCK)						28%	
Uncollapsed (UC)						0%	
Processing time in seconds						290/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	10	27	13	0	0	0	50
MS	50	0	0	0	0	0	50
PCK	1	0	35	14	0	0	50
UC	50	0	0	0	0	0	50

VQ (2nd Row failure)							
						Results	
Overall classification rate						37.2%	
First column (FC)						100%	
First storey (FS)						52%	
Mid-storey (MS)						0%	
Pancake (PCK)						34%	
Uncollapsed (UC)						0%	
Processing time in seconds						290/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	11	26	13	0	0	0	50
MS	50	0	0	0	0	0	50
PCK	0	0	33	17	0	0	50
UC	50	0	0	0	0	0	50

HMM (2 sensors failure)							
				HMM LIoA case			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (3 sensors failure)							
				HMM LIoA case			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (4 sensors failure)							
				HMM LIoA case			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (5 sensors failure)							
				HMM LIoA case			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (6 sensors failure)							
				HMM LIoA case			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (2 sensors failure)							
				HMM MLoA case			
Overall classification rate				98.8%			
First column (FC)				100%			
First storey (FS)				94%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	47	3	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (3 sensors failure)							
				HMM MIoA case			
Overall classification rate				87.6%			
First column (FC)				100%			
First storey (FS)				38%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	19	31	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (4 sensors failure)							
				HMM MIoA case			
Overall classification rate				97.6%			
First column (FC)				100%			
First storey (FS)				88%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	44	6	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (5 sensors failure)							
				HMM MIoA case			
Overall classification rate				80.8%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				4%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	48	2	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (6 sensors failure)							
				HMM MIoA case			
Overall classification rate				80.8%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				4%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	48	2	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (1 st column failure)							
					Results		
Overall classification rate					65.2%		
First column (FC)					0%		
First storey (FS)					26%		
Mid-storey (MS)					100%		
Pancake (PCK)					100%		
Uncollapsed (UC)					100%		
Processing time in seconds					22000/250		
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	50	0	50
FS	0	13	37	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (2 nd column failure)							
				Results			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22000/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (3 rd column failure)							
					Results		
Overall classification rate					100%		
First column (FC)					100%		
First storey (FS)					100%		
Mid-storey (MS)					100%		
Pancake (PCK)					100%		
Uncollapsed (UC)					100%		
Processing time in seconds					22000/250		
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

HMM (4 th column failure)							
						Results	
Overall classification rate						85.2%	
First column (FC)						100%	
First storey (FS)						100%	
Mid-storey (MS)						100%	
Pancake (PCK)						26%	
Uncollapsed (UC)						100%	
Processing time in seconds						22000/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	37	0	0	13	0	0	50
UC	0	0	0	0	50	0	50

HMM (1 st Row failure)							
					Results		
Overall classification rate					60%		
First column (FC)					100%		
First storey (FS)					0%		
Mid-storey (MS)					100%		
Pancake (PCK)					0%		
Uncollapsed (UC)					100%		
Processing time in seconds					22000/250		
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	1	0	0	49	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	50	0	0	0	50
UC	0	0	0	0	50	0	50

HMM (2 nd Row failure)							
						Results	
Overall classification rate						60%	
First column (FC)						100%	
First storey (FS)						100%	
Mid-storey (MS)						0%	
Pancake (PCK)						0%	
Uncollapsed (UC)						100%	
Processing time in seconds						22000/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	50	0	0	0	0	0	50
PCK	0	50	0	0	0	0	50
UC	0	0	0	0	50	0	50

HMM (3 rd Row failure)							
						Results	
Overall classification rate						80%	
First column (FC)						100%	
First storey (FS)						100%	
Mid-storey (MS)						100%	
Pancake (PCK)						0%	
Uncollapsed (UC)						100%	
Processing time in seconds						22000/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	50	0	0	0	0	50
UC	0	0	0	0	50	0	50

Hybrid (6 sensors failure)							
				MIoA case			
Overall classification rate				100%			
First column (FC)				100%			
First storey (FS)				100%			
Mid-storey (MS)				100%			
Pancake (PCK)				100%			
Uncollapsed (UC)				100%			
Processing time in seconds				22200/250			
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	50	0	0	0	0	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

Hybrid (1 st column failure)							
						Results	
Overall classification rate						80%	
First column (FC)						0%	
First storey (FS)						100%	
Mid-storey (MS)						100%	
Pancake (PCK)						100%	
Uncollapsed (UC)						100%	
Processing time in seconds						22200/250	
Confusion Matrix							
	FC	FS	MS	PCK	UC	Unclassified	Total
FC	0	0	0	0	50	0	50
FS	0	50	0	0	0	0	50
MS	0	0	50	0	0	0	50
PCK	0	0	0	50	0	0	50
UC	0	0	0	0	50	0	50

APPENDIX C

(Python code to capture sensors velocity in three axis (X,Y, and Z))

```
import bge
import time
from bge import logic

def Player():

    # appending in existing file
    store_file = open ('/Python27/Database/11.txt', 'a')

    # Capture the object
    cont = logic.getCurrentController()
    obj = cont.owner

    # chaptureing sensor Velocity
    Xspeed, Yspeed, Zspeed = obj.getLinearVelocity(False)

    xs=Xspeed
    ys = Yspeed
    zs = Zspeed

    #Get time

    from time import gmtime, strftime
    ticks_m = strftime("%M", gmtime())
    ticks_s = strftime("%S", gmtime())

    # procedure to check if the sensor went through the ground. if yes, program
    # will locate the sensor close to the ground.

    if zz < 0 :
        xs =0.0005026459693908691
        ys =0.0005026459693908691
        zs =0.0005026459693908691

    # Saving sensor velocity in file
    xspeed = str(xs)
    yspeed = str (ys)
    zspeed = str (zs)
    store_file.write(xspeed)
    store_file.write(',')
    store_file.write(yspeed)
    store_file.write(',')
    store_file.write(zspeed)
    store_file.write(',')

    # write in the csv. file
    store_file.write('\n')
```



```
#Close the csv file
store_file.close()
```

(MATLAB Code to convert .csv files to .xls files and generate plot for each sensor)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% this program does the following processing
%1. import the .csv file that contain sensor velocity in x y z
direction
%2. save sensor velocity in xls file.
%3. save the sansors reading plots as a image.

global building_sensors; % save the sensor data in the structure
called building_sensors

% Eenter the buidling collapse pattern
disp('please select the Pattern type');
strResponse = input('Enter Building Collapse
Pattern(ST_V,FC_V,FS_V,MS_V,PC_V,UN_V):','s');
N = input('Enter The Pattern Number: ','s');

% Import the txt file
ind = 0;
for i = 10:10:120
    for j = 1:4
        % Create a file name
        ind = ind +1;
        z=i+j;
        file_name = int2str(z);
        file_name_txt = strcat ('text_file\',file_name, '.txt');

        sensor = importdata (file_name_txt);
        Velocity = zeros(size(sensor));% the Velocity array

        n = length (sensor); % length of array

        % read vleocity for x y z direction
        sensor_z = sensor(:,3);
        sensor_y = sensor(:,2);
        sensor_x = sensor(:,1);

        % Save Velocity
        Velocity(1:n,3) = sensor_z ;
        Velocity(1:n,2) = sensor_y ;
        Velocity(1:n,1) = sensor_x ;

        % save the sensor data in the structure called
        building_sensors

        building_sensors(ind).name = file_name;
```



```

building_sensors(ind).data = Velocity;

% Save the Velocity in Excel file format (.xls)
file_name_xls = strcat ('xls_file\',file_name, '.xls');
xlswrite (file_name_xls,Velocity);
end
end

% Rename the structure based on the building collapse pattern and
save the
% structure in m-file
switch ( strResponse)
    case 'FC_V'

        assignin('base', strcat ('FC',N,'_V'), building_sensors) ;
        save(strcat ('FC',N,'_V.mat'),strcat ('FC',N,'_V'));

    case 'FS_V'
        assignin('base', strcat ('FS',N,'_V'), building_sensors) ;
        save(strcat ('FS',N,'_V.mat'),strcat ('FS',N,'_V'));

    case 'MS_V'
        assignin('base', strcat ('MS',N,'_V'), building_sensors) ;
        save(strcat ('MS',N,'_V.mat'),strcat ('MS',N,'_V'));

    case 'PC_V'
        assignin('base', strcat ('PC',N,'_V'), building_sensors) ;
        save(strcat ('PC',N,'_V.mat'),strcat ('PC',N,'_V'));

    case 'UN_V'
        assignin('base', strcat ('UN',N,'_V'), building_sensors) ;
        save(strcat ('UN',N,'_V.mat'),strcat ('UN',N,'_V'));

    case 'ST_V'

        assignin('base', strcat ('ST',N,'_V'), building_sensors) ;
        save(strcat ('ST',N,'_V.mat'),strcat ('ST',N,'_V'));

    otherwise
        disp('You entered Undefine Pattern');
end
%
% %save the sensor plotes as an Image files
h = figure;
for i = 1:48
    sensor = building_sensors(i).name;

    plot(building_sensors(1,i).data,'DisplayName','building_sensors(1,1)
        .data','YDataSource','building_sensors(1,1).data');

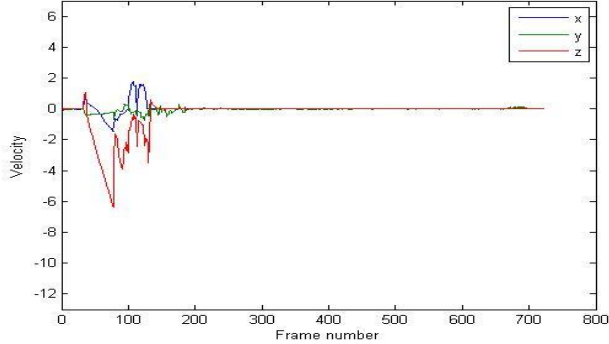
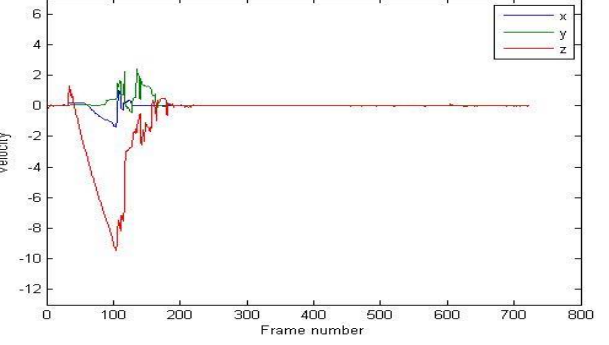
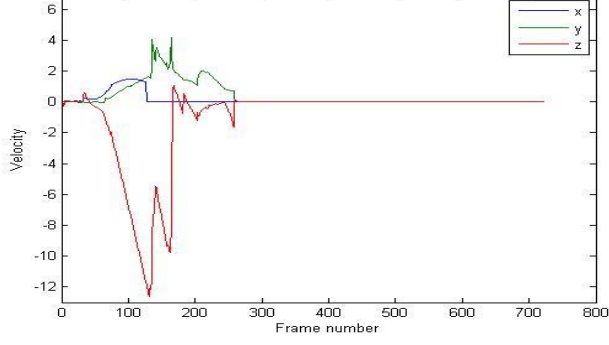
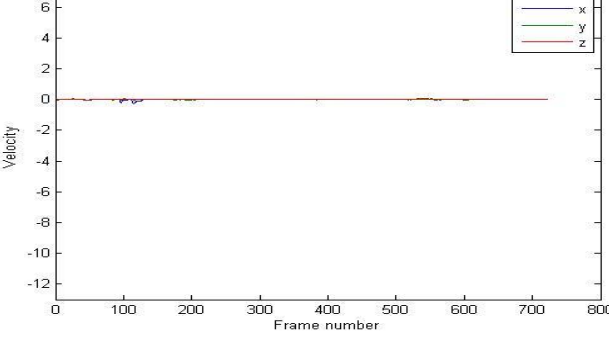
```

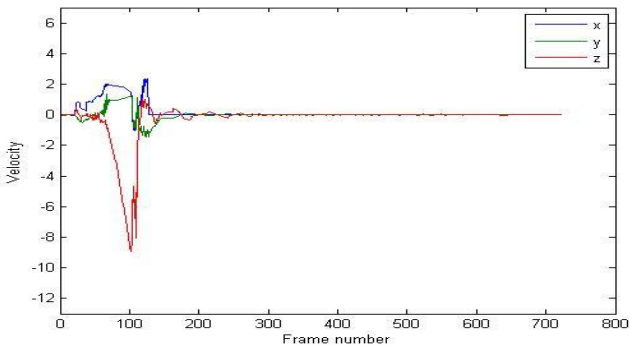
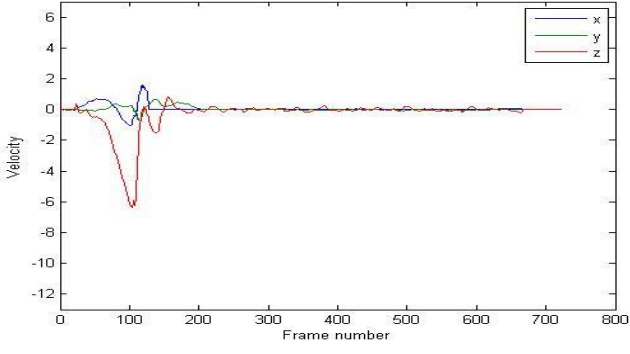
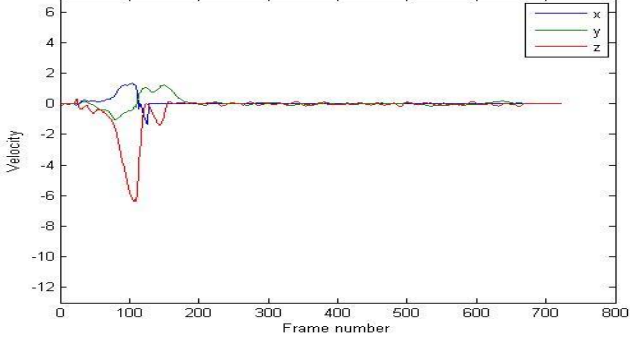


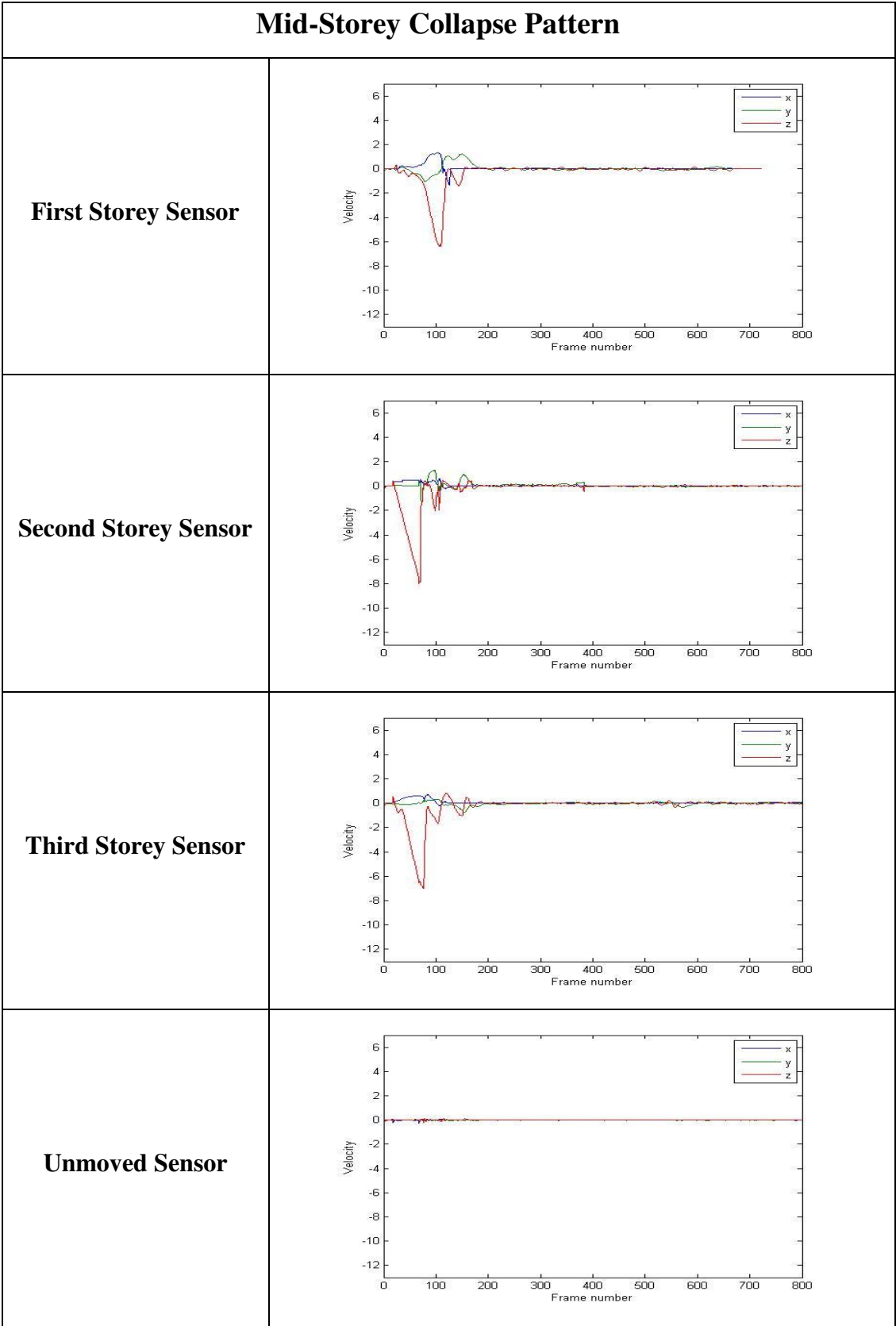
```
title(building_sensors(i).name,'FontWeight','bold');  
legend('x','y','z');  
%print(h)  
file_name_jpg = strcat ('Images\',sensor, '.jpeg');  
print(h,'-djpeg',file_name_jpg);  
end
```

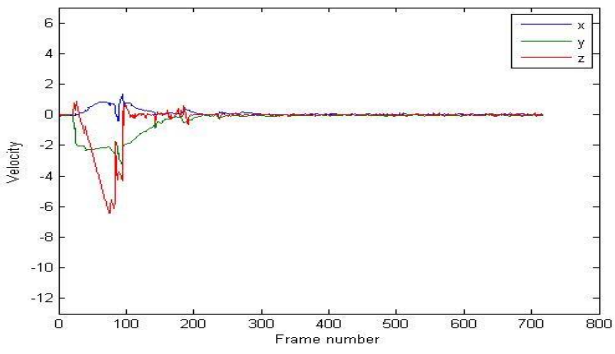
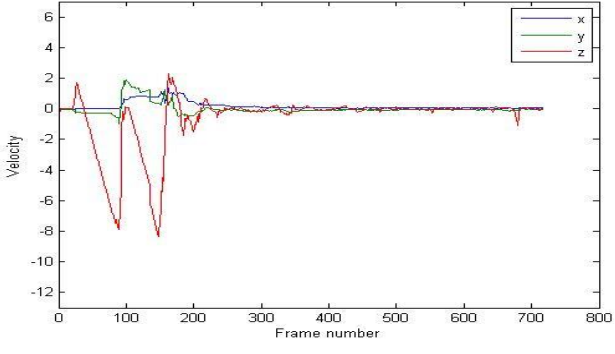
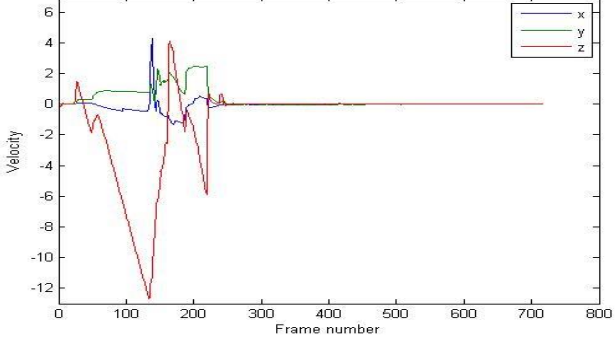

APPENDIX D

Table 4.1: Examples of sensor readings from different locations.

First Column Collapse Pattern	
First Storey Sensor	
Second Storey Sensor	
Third Storey Sensor	
Unmoved Sensor	

First Storey Collapse Pattern	
First Storey Sensor	
Second Storey Sensor	
Third Storey Sensor	
Unmoved Sensor	Not applicable



Pancake Collapse Pattern	
First Storey Sensor	 <p>This plot shows the velocity response of the first storey sensor over 800 frames. The y-axis represents velocity from -12 to 6, and the x-axis represents frame number from 0 to 800. Three data series are shown: x (blue), y (green), and z (red). All three series show a sharp initial drop to a minimum around frame 100, with the z-axis reaching the lowest point at approximately -7.5. After this initial drop, the velocities for all three axes gradually return to zero by frame 200 and remain stable until frame 700.</p>
Second Storey Sensor	 <p>This plot shows the velocity response of the second storey sensor. The axes and legend are the same as the first plot. The x, y, and z series show a similar initial drop to a minimum around frame 100, with the z-axis reaching approximately -8.5. The response is slightly more oscillatory than the first storey, with some smaller fluctuations before settling back to zero by frame 200.</p>
Third Storey Sensor	 <p>This plot shows the velocity response of the third storey sensor. The axes and legend are the same. The x, y, and z series show a sharp initial drop to a minimum around frame 150, with the z-axis reaching approximately -11.5. The response is highly oscillatory, with significant fluctuations in the x and y axes before settling back to zero by frame 200.</p>
Unmoved Sensor	Not applicable

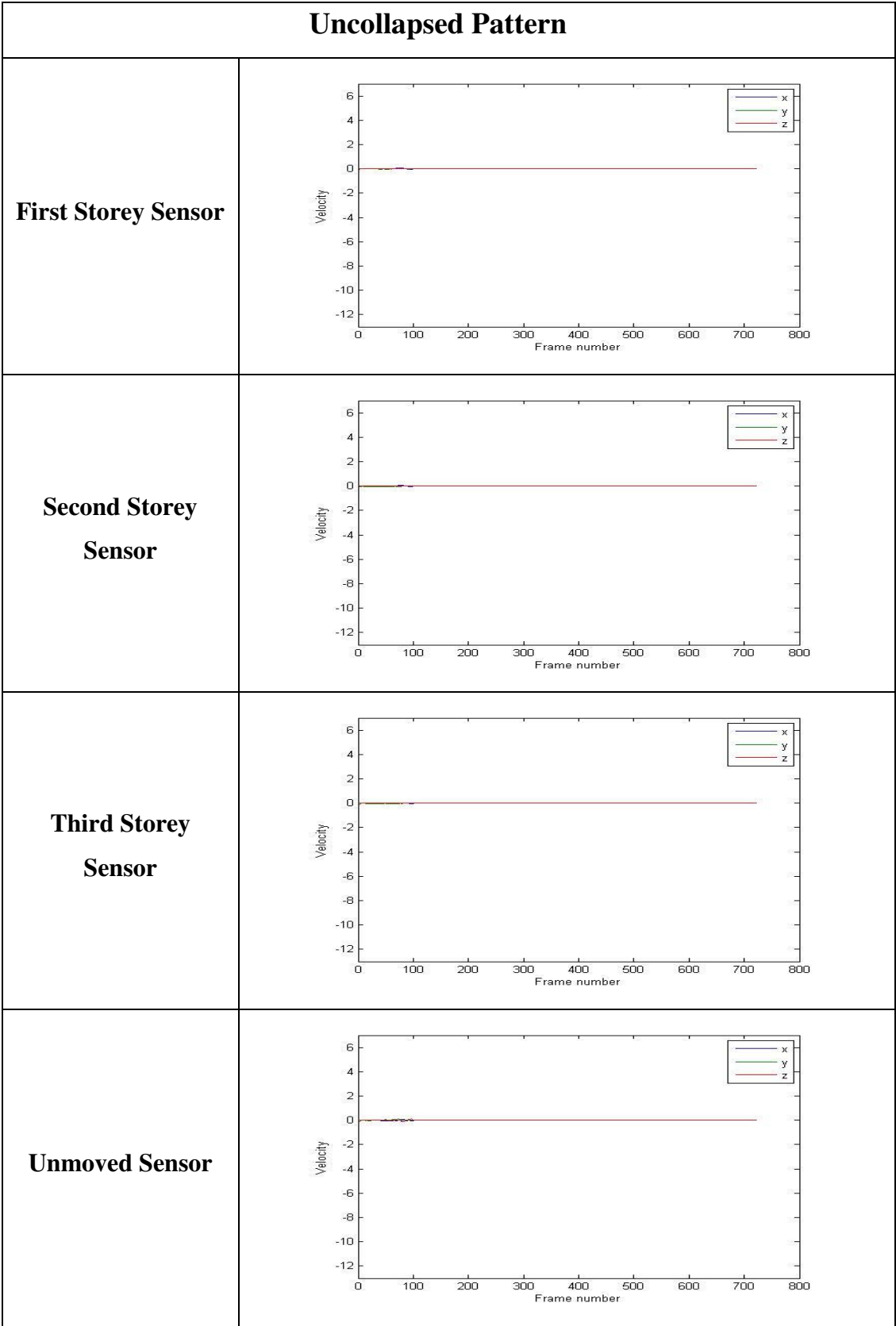
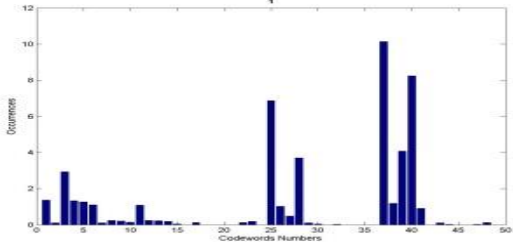
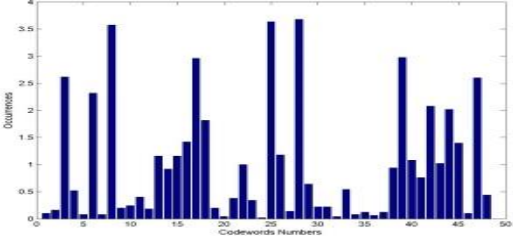
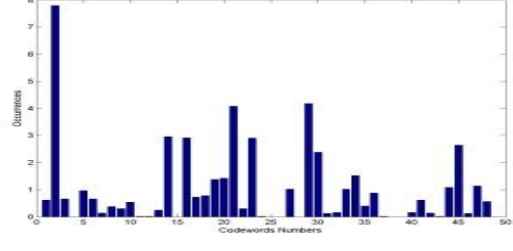
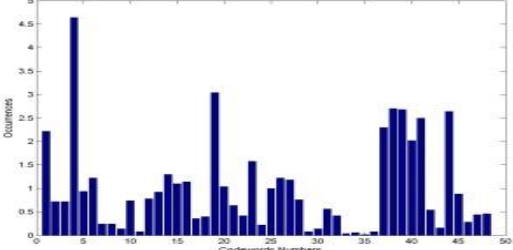
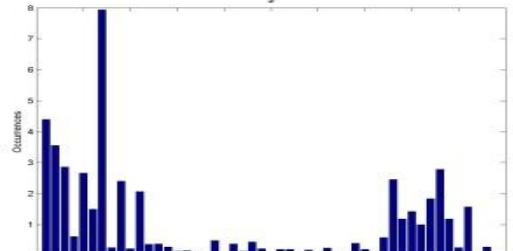
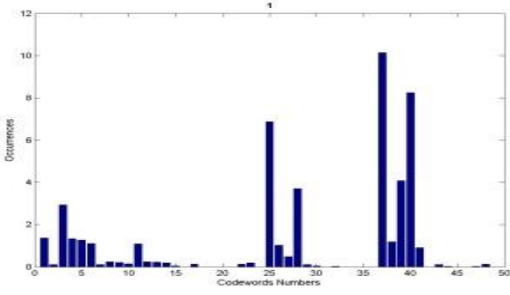
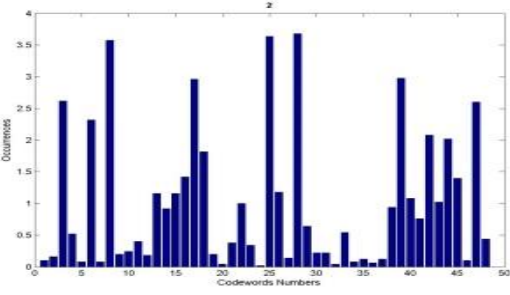
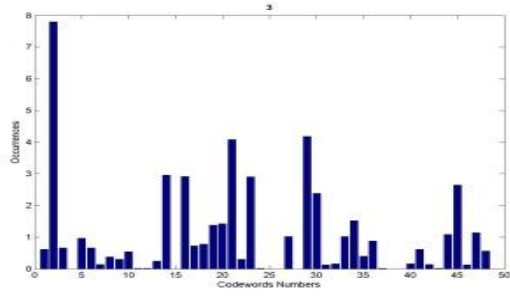
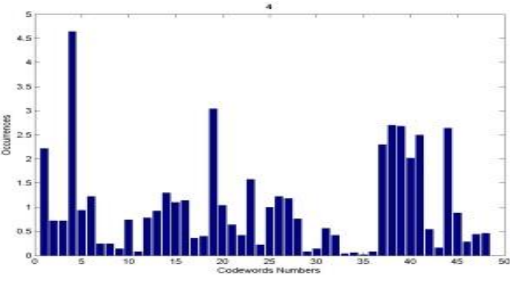
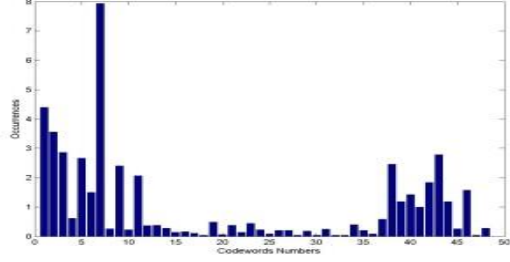
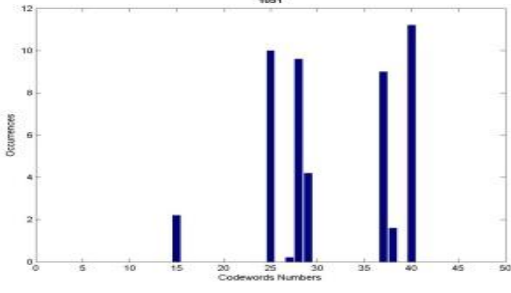
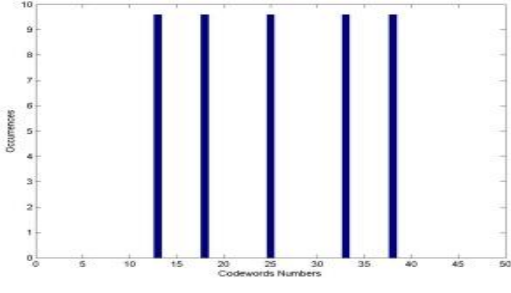
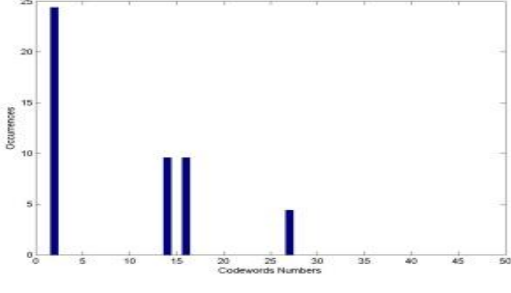
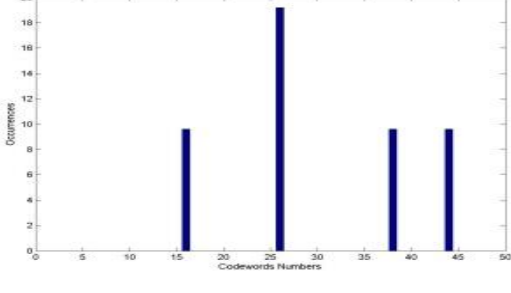
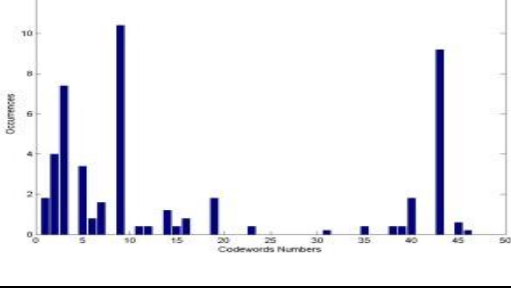


Table 4.8: Example of generated histograms using VQ codebooks.

Collapse patterns	General Histogram example
First Column	
First Storey	
Mid-Storey	
Pancake	
Uncollapsed	

Collapse patterns	General Histogram example
First Column	
First Storey	
Mid-Storey	
Pancake	
Uncollapsed	

Collapse patterns	Uncollapsed misclassified patterns																																						
First Column	 <p>1051</p> <table border="1"> <thead> <tr> <th>Codewords Numbers</th> <th>Counts</th> </tr> </thead> <tbody> <tr><td>15</td><td>2</td></tr> <tr><td>25</td><td>10</td></tr> <tr><td>28</td><td>9</td></tr> <tr><td>30</td><td>4</td></tr> <tr><td>38</td><td>9</td></tr> <tr><td>39</td><td>1</td></tr> <tr><td>40</td><td>11</td></tr> </tbody> </table>	Codewords Numbers	Counts	15	2	25	10	28	9	30	4	38	9	39	1	40	11																						
Codewords Numbers	Counts																																						
15	2																																						
25	10																																						
28	9																																						
30	4																																						
38	9																																						
39	1																																						
40	11																																						
First Storey	 <p>1052</p> <table border="1"> <thead> <tr> <th>Codewords Numbers</th> <th>Counts</th> </tr> </thead> <tbody> <tr><td>15</td><td>9</td></tr> <tr><td>18</td><td>9</td></tr> <tr><td>25</td><td>9</td></tr> <tr><td>35</td><td>9</td></tr> <tr><td>38</td><td>9</td></tr> </tbody> </table>	Codewords Numbers	Counts	15	9	18	9	25	9	35	9	38	9																										
Codewords Numbers	Counts																																						
15	9																																						
18	9																																						
25	9																																						
35	9																																						
38	9																																						
Mid-Storey	 <p>1053</p> <table border="1"> <thead> <tr> <th>Codewords Numbers</th> <th>Counts</th> </tr> </thead> <tbody> <tr><td>2</td><td>24</td></tr> <tr><td>15</td><td>9</td></tr> <tr><td>16</td><td>9</td></tr> <tr><td>28</td><td>4</td></tr> </tbody> </table>	Codewords Numbers	Counts	2	24	15	9	16	9	28	4																												
Codewords Numbers	Counts																																						
2	24																																						
15	9																																						
16	9																																						
28	4																																						
Pancake	 <p>1054</p> <table border="1"> <thead> <tr> <th>Codewords Numbers</th> <th>Counts</th> </tr> </thead> <tbody> <tr><td>15</td><td>9</td></tr> <tr><td>25</td><td>19</td></tr> <tr><td>38</td><td>9</td></tr> <tr><td>45</td><td>9</td></tr> </tbody> </table>	Codewords Numbers	Counts	15	9	25	19	38	9	45	9																												
Codewords Numbers	Counts																																						
15	9																																						
25	19																																						
38	9																																						
45	9																																						
Uncollapsed	 <p>1055</p> <table border="1"> <thead> <tr> <th>Codewords Numbers</th> <th>Counts</th> </tr> </thead> <tbody> <tr><td>2</td><td>2</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>4</td><td>7</td></tr> <tr><td>5</td><td>3</td></tr> <tr><td>6</td><td>1</td></tr> <tr><td>7</td><td>2</td></tr> <tr><td>10</td><td>10</td></tr> <tr><td>14</td><td>1</td></tr> <tr><td>15</td><td>1</td></tr> <tr><td>16</td><td>1</td></tr> <tr><td>20</td><td>2</td></tr> <tr><td>25</td><td>0.5</td></tr> <tr><td>30</td><td>0.5</td></tr> <tr><td>35</td><td>0.5</td></tr> <tr><td>38</td><td>0.5</td></tr> <tr><td>40</td><td>2</td></tr> <tr><td>45</td><td>9</td></tr> <tr><td>46</td><td>0.5</td></tr> </tbody> </table>	Codewords Numbers	Counts	2	2	3	4	4	7	5	3	6	1	7	2	10	10	14	1	15	1	16	1	20	2	25	0.5	30	0.5	35	0.5	38	0.5	40	2	45	9	46	0.5
Codewords Numbers	Counts																																						
2	2																																						
3	4																																						
4	7																																						
5	3																																						
6	1																																						
7	2																																						
10	10																																						
14	1																																						
15	1																																						
16	1																																						
20	2																																						
25	0.5																																						
30	0.5																																						
35	0.5																																						
38	0.5																																						
40	2																																						
45	9																																						
46	0.5																																						