Unsupervised Structural Damage Detection and Localization Using Deep Learning and Machine Learning

By

Zilong Wang

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
In partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Civil Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

Copyright © 2021 by Zilong Wang
Many data-driven approaches have been developed in recent decades to address problems with damage detection for civil infrastructure. According to training modes of the statistical models or neural networks adopted in the studies, these data-driven damage detection methods can be roughly categorized into supervised modes and unsupervised modes. Supervised damage detection approaches require the recorded data (i.e., ground truth data) from the undamaged and various damaged structural scenarios to train statistical models or neural networks. Then, the trained models or networks can be utilized to detect damage using future data measured from unknown structural scenarios. However, acquiring numerous training datasets from various damage scenarios for the monitored structures is time-consuming and costly, and it is hard to obtain many damage scenarios for the infrastructures in service. To address these challenges encountered in practice, structural damage detection in unsupervised learning mode has become increasingly interesting to researchers. The proposed unsupervised damage detection methods in my study require only the data measured from undamaged structural scenarios or baseline structures in their training processes. This thesis aims to propose novel unsupervised damage detection methods to address the problems facing structural damage detection and localization. Specifically, a novel unsupervised damage detection approach using a deep learning technique is proposed for detecting damage in a simulated multi-story frame and a laboratory-scale steel bridge model in Chapter 3. Additionally, a comparative study with an advanced unsupervised damage detection approach using deep restricted Boltzmann machines is carried out to evaluate their effectiveness of detecting light damage in the steel bridge. In Chapter 4, an unsupervised novelty detection method based on an original technique of fast clustering is developed to roughly locate the damage positions in a small-scale building frame. To verify the effectiveness of the developed method for structural damage localization, several existing machine learning and deep learning methods are
developed and converted to the uniform unsupervised novelty detection mode in Chapter 5 for extensive comparative studies.
Co-Authorship

**Wang, Z. & Cha, Y. J. (2021).** Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect structural damage. *Structural Health Monitoring*, 20(1), 406–425. DOI: 10.1177/1475921720934051 (Impact factor: 4.87), [Chapter 3]. I initiated the research project by proposing a research plan. I contributed to generating and collecting the experimental datasets, designing and developing the unsupervised damage detection approach, conducting the comparative studies, visualizing the damage detection results, writing the paper manuscript, and responding to the reviewers’ comments.

Cha, Y. J. & **Wang, Z.** (2018). Unsupervised novelty detection-based structural damage localization using a density peaks-based fast clustering algorithm. *Structural Health Monitoring*, 17(2), 313–324. DOI: 10.1177/14759217177691260 (Impact factor: 4.87), [Chapter 4]. I designed and developed the unsupervised damage localization method, presenting the results of the case study, conducting the comparative study, writing the paper manuscript, and responding to the reviewers’ comments.
Acknowledgements

First of all, I would like to express my deep gratitude to my advisor, Professor Young-jin Cha, who provided a great deal of guidance and support throughout the writing of this thesis and my research work at the Department of Civil Engineering. I would also like to thank Professor Dagmar Svecova and Professor Yang Wang, who works at the Department of Computer Science, for the valuable advice given by them has been a great help in keeping my progress in the research study. I would also like to extend my sincere thanks to the University of Manitoba Steel Bridge Team in 2017 for their generosity and assistance in my preparation work for the experimental setups presented in this thesis. Lastly, I wish to take this opportunity to express my gratitude to my loving parents for their meticulous concern and wise counsel in my life.

Funding: The conducted research study in this thesis was under the support of an NSERC Discovery grant (Common Personal Identifier: 1262624).
# Table of Contents

Abstract ......................................................................................................................................... ii  
Co-Authorship .............................................................................................................................. iv  
Acknowledgements ...................................................................................................................... v  
Table of Contents .......................................................................................................................... vi  
List of Figures .................................................................................................................................. viii  
List of Tables ................................................................................................................................... xi  

Chapter 1: Introduction .................................................................................................................. 1  
  1.1 Background .......................................................................................................................... 2  
  1.2 Objectives ........................................................................................................................... 4  
  1.3 Scope of work ....................................................................................................................... 4  
  1.4 Novelties and contributions ............................................................................................... 6  

Chapter 2: Literature review of supervised and unsupervised damage detection methods ...... 7  
  2.1 Distance-based machine learning technique .................................................................. 8  
  2.2 Clustering-based machine learning technique ............................................................... 9  
  2.3 Support vector machine technique .............................................................................. 10  
  2.4 Limitations of existing works with deep learning ....................................................... 11  

Chapter 3: Unsupervised deep learning method for structural damage detection ............... 13  
  3.1 Deep auto-encoders and OC-SVMs ............................................................................. 14  
    3.1.1 Data reconstruction with deep auto-encoders ..................................................... 15  
    3.1.2 Damage-sensitive features ............................................................................. 19  
    3.1.3 Damage detection with OC-SVM ................................................................. 21  
  3.2 Numerical and experimental setups ............................................................................. 23  
    3.2.1 Numerical multi-story frame ........................................................................ 24  
    3.2.2 Laboratory-scale bridge model .................................................................. 26  
  3.3 Case studies of damage detection .............................................................................. 28  
    3.3.1 Case study of the multi-story building frame .............................................. 29  
    3.3.2 Case study of the small-scale steel bridge .................................................. 35
3.4 Comparative studies ........................................................................................................42
3.5 Limitations ......................................................................................................................47

Chapter 4: Unsupervised fast novelty detection for damage localization .................. 49
4.1 Unsupervised novelty detection method ..................................................................50
4.2 Damage-sensitive features .......................................................................................57
   4.2.1 Crest factor ..........................................................................................................58
   4.2.2 T-CWT .................................................................................................................58
4.3 Experiment setup .......................................................................................................60
4.4 Case studies of damage localization .......................................................................62

Chapter 5: Comparative studies of damage detection and localization .................... 68
5.1 Unsupervised novelty detection methods ...............................................................69
   5.1.1 Unsupervised KNN .............................................................................................69
   5.1.2 FCM method ........................................................................................................71
   5.1.3 GMMs ..................................................................................................................74
   5.1.4 OC-SVMs .............................................................................................................77
   5.1.5 Unsupervised CNNs ..........................................................................................81
5.2 Comparative studies ................................................................................................84
   5.2.1 Unsupervised KNN method ................................................................................84
   5.2.2 FCM method ........................................................................................................86
   5.2.3 GMM method ........................................................................................................88
   5.2.4 OC-SVM method ..................................................................................................89
   5.2.5 Unsupervised CNN method ...............................................................................91
5.3 Summary .....................................................................................................................93

Chapter 6: Conclusions .................................................................................................... 96
6.1 Achievements ..............................................................................................................97
6.2 Limitations .................................................................................................................98
6.3 Future works .............................................................................................................99

References ......................................................................................................................100
List of Figures

Figure 3-1. Procedures of damage detection with my developed unsupervised deep learning method (Wang and Cha, 2021).......................................................... 14

Figure 3-2. Architecture of neural network for a deep auto-encoder (Wang and Cha, 2021)........................................................................................................... 17

Figure 3-3. Reconstruction performances of a deep auto-encoder for acceleration responses from an IS (a) and a DS (b) (Wang and Cha, 2021). .............. 19

Figure 3-4. Unsupervised damage detection using OC-SVMs (Wang and Cha, 2021). .. 22

Figure 3-5. 12-story building frame (Wang and Cha, 2021). ................................. 24

Figure 3-6. Components of the excitation wave (Wang and Cha, 2021)............... 25

Figure 3-7. Small-scale steel bridge (Wang and Cha, 2021). ................................. 27

Figure 3-8. Deployment of the sensors installed on the steel bridge (Wang and Cha, 2021). ........................................................................................................... 27

Figure 3-9. Data reconstruction performances of deep auto-encoders at different depths using the training data for the 1st floor of the multi-story frame (Wang and Cha, 2021). ........................................................................................................... 30

Figure 3-10. Data reconstruction losses of the datasets for the 3rd floor of the multi-story frame (Wang and Cha, 2021). ...................................................... 31

Figure 3-11. Distributions of formed feature points on a 2D plane for all floors of the multi-story frame (Wang and Cha, 2021)........................................... 32

Figure 3-12. Damage detection with an OC-SVM at the 2nd floor of the multi-story frame (Wang and Cha, 2021). ................................................................. 33

Figure 3-13. Data reconstruction loss of the datasets from Sensor #5 on the steel bridge (Wang and Cha, 2021)................................................................. 36

Figure 3-14. Distributions of feature points and damage detection with OC-SVMs on a 2D plane for 10 sensors of the steel bridge (Wang and Cha, 2021)........ 38
Figure 3-15. Distributions of feature points and damage detection with OC-SVMs in a 3D space for 10 sensors of the steel bridge (Wang and Cha, 2021)................................. 40
Figure 3-16. Damage detection with the Mahalanobis distance technique at Sensor #1 of the steel bridge (Wang and Cha, 2021)............................................................... 43
Figure 4-1. Local density of a data point (Cha and Wang, 2018)........................................ 51
Figure 4-2. Example distance $\delta 9$ calculation (Cha and Wang, 2018). ......................... 52
Figure 4-3. Selection of density peak points (Cha and Wang, 2018). .............................. 52
Figure 4-4. Performance of Step 3 in clustering (Cha and Wang, 2018)............................ 53
Figure 4-5. Distribution of the final formed clusters (Cha and Wang, 2018)....................... 53
Figure 4-6. Border region between two clusters (Cha and Wang, 2018). .......................... 54
Figure 4-7. Detected novelty points in the formed clusters (Cha and Wang, 2018)............. 55
Figure 4-8. Damage detection procedure of the improved DPFC method (Cha and Wang, 2018)........................................................................................................... 57
Figure 4-9. Peaks in the amplitude of the CWT of acceleration responses of a structural joint in its intact scenario (Cha and Wang, 2018). ................................................. 59
Figure 4-10. Peaks in the amplitude of the CWT of acceleration responses of a structure in its damaged scenario (Cha and Wang, 2018). .............................................. 60
Figure 4-11. Small-scale steel building model (Cha and Wang, 2018)................................. 61
Figure 4-12. Connection details of a structural joint and an installed sensor (Cha and Wang, 2018). ................................................................................................................. 61
Figure 4-13. DPFC method: Selection of cluster density peak points for damage detection at Joint 3 (Cha and Wang, 2018). ................................................................. 65
Figure 4-14. DPFC method: The well-trained statistical model and the detected halo points for Joint 3 (Cha and Wang, 2018)............................................................... 65
Figure 4-15. DPFC method: The performance of damage detection at Joint 3 for DS4 (Cha and Wang, 2018)....................................................................................... 66
Figure 5-1. Novelty detection with the unsupervised KNN method. .................................. 70
Figure 5-2. Novelty detection with the FCM method........................................................... 73
Figure 5-3. Novelty detection with the GMM method using Mahalanobis distance........... 76
Figure 5-4. Binary classification with an SVM. ................................................................. 77
Figure 5-5. Novelty detection using an OC-SVM. ............................................................ 79
Figure 5-6. Convolution and pooling operations on a 2D matrix (a) and a CNN for supervised classification (b). ................................................................. 82

Figure 5-7. Neural network of an unsupervised CNN for data reconstruction (Wang and Cha, 2018). ................................................................................. 83

Figure 5-8. Calculated rates of damage detection at eighteen joints with the unsupervised KNN method................................................................. 85

Figure 5-9. Unsupervised KNN method: Performance of damage detection at Joint 5 for DS3. ..................................................................................... 85

Figure 5-10. Calculated rates of damage detection at eighteen joints with the FCM method. ................................................................................. 87

Figure 5-11. FCM method: Performance of damage detection at Joint 3 for DS1. ... 87

Figure 5-12. Calculated rates of damage detection at eighteen joints with the GMM method. ................................................................................. 88

Figure 5-13. GMM method: Performance of damage detection at Joint 17 for DS2. 89

Figure 5-14. Calculated rates of damage detection at eighteen joints with the OC-SVMs. ......................................................................................... 90

Figure 5-15. OC-SVM method: Performance of damage detection at Joint 11 for DS3. 90

Figure 5-16. Damage detection with the Mahalanobis distance technique at Joint 1. 92

Figure 5-17. Calculated rates of damage detection at eighteen joints with the unsupervised CNN method................................................................. 92

Figure 5-18. Calculated rates of damage detection at eighteen joints (Table 4-2) with the DPFC method. ................................................................. 93
List of Tables

Table 3-1. Structural scenarios and numbers of vibration tests (Wang and Cha, 2021)... 26
Table 3-2. Various structural scenarios and numbers of vibration tests (Wang and Cha, 2021). ............................................................................................................. 28
Table 3-3. Damage detection with OC-SVMs for all floors of the multi-story frame (Wang and Cha, 2021). ............................................................................................................. 34
Table 3-4. Damage detection with OC-SVMs on a 2D feature plane for all sensors of the steel bridge (Wang and Cha, 2021). ............................................................... 37
Table 3-5. Damage detection with OC-SVMs in a 3D feature space for all sensors of the steel bridge (Wang and Cha, 2021). ............................................................... 41
Table 3-6. Damage detection with the Mahalanobis distance technique using the 3D feature dataset for the steel bridge (Wang and Cha, 2021). ....................... 44
Table 3-7. Calculated global and local SHIs with a DRBM model for damage detection on the steel bridge (Wang and Cha, 2021). ............................... 47
Table 4-1. Structural scenarios and test numbers (Cha and Wang, 2018). ................. 62
Table 4-2. Calculated rates of damage detection at the structural joints with the DPFC method (Cha and Wang, 2018). ............................................................... 63
Table 5-1. Summary of the advantages and limitations of the proposed unsupervised learning methods. ......................................................................................... 94
Chapter 1: Introduction

In this chapter, a broad scientific background of the developed means of vibration-based structural damage detection is introduced, and the drawbacks of a selection of related state-of-the-art approaches are outlined. Afterwards, several objectives focusing on the techniques of unsupervised damage detection are presented, which can address the outlined drawbacks of the research studies in this chapter. Finally, the scope of research work, the novelties and contributions in this thesis are stated.
1.1 Background

Civil infrastructures inevitably accumulate many types of damage and experience strength attenuation due to the influences of various loads and ambient environmental effects that impact their structural safety. During the last decades, more and more attention has been paid to the health monitoring of civil infrastructures (Rogers, 1979; Tarricone, 1990). In fact, human sources with experience of structural damage inspection are quite limited and some specific parts of monitored structures, such as the load-bearing structural components at the bottom of bridges and the external surfaces of high towers, are not easy to access. To overcome these obvious shortcomings in practice, developing efficient and effective technical tools to detect various types of damage in civil infrastructures is in demand (Sohn et al., 2002; Farrar and Worden, 2012).

Structural health monitoring (SHM) has emerged as an in-situ field experimental technique since the early 1990s to monitor the behaviors and performances of infrastructure under real loadings and ambient environmental conditions, and to further ensure their safety and durability in service (Li and Ou, 2016). The aim of SHM is to understand the health conditions of the monitored structures and to provide timely information on the presence of structural damage by collecting and analyzing the data measured from the sensor network installed on them. An SHM system built on a monitored structure should reliably achieve the goals of preventing disastrous structural collapses, detecting damage in time, reducing maintenance cost, and prolonging structural service life. The technique of vibration-based damage detection is a common and widely used approach in the discipline of SHM because the changes in the measured vibrational responses of the monitored structures before and after damage can be identified to indicate the presence of damage (Farrar and Worden, 2012). The vibration-based damage detection approaches can be broadly classified into physical model-based and data-driven approaches (Barthorpe, 2010).

In physical model-based damage detection, the technique of finite element (FE) modeling is commonly used to diagnose the health conditions of the monitored structures (Lee and Eun, 2014). Specifically, a FE model (FEM) as a baseline must be built with the fundamental information of a monitored structure in its healthy scenario. The built FEM of that structure can be updated by using the data obtained from its real structural scenario.
Then, the updated FEM can be compared with the baseline FEM to evaluate the real health conditions of that structure (Jayanthan and Srinivas, 2015).

In data-driven approaches, only the data measured from the system of sensor network built on structures are required for assessing structural health conditions (Barthorpe, 2010; Farrar and Worden, 2012). For data acquisition in the SHM system, the most widely used sensors are accelerometers, strain gauges, inclinometers, etc. The collected data from the installed sensors are usually in the form of signals in time series, which contain useful structural and damage information. According to the training modes of the statistical models or neural networks used, data-driven damage detection can be generally classified into supervised and unsupervised modes.

Supervised damage detection methods need a large quantity of data acquired from both intact (undamaged) and different damaged structural scenarios as training data. Even though the supervised learning methods may be able to show excellent performance in structural damage detection (Abdeljaber et al., 2017; Lin et al., 2017; Eren et al., 2018), they have a serious shortcoming with respect to acquiring numerous training data for the following practical reasons: 1) it is hard or even impractical to collect numerous training data from different damage scenarios for the same structure, especially for civil infrastructure in service; 2) each structure has its own unique structural properties and boundary conditions, thus the training data collected from a specific structure can only be used in damage detection for that structure. Even if some structures have the same shape, size, and materials, due to the issues of aging, variation of material properties, and their unique boundary conditions, each of them still has unique dynamic modal properties and dynamic behaviors; and 3) even though sufficient training data for various damaged structural scenarios can be easily generated by FEMs of the monitored structures, it is still difficult to develop fine-tuned FEMs that can address the issues of uncertain ambient excitations, complex boundary conditions, and nonlinear relationships between extents of structural damage and changes of properties of FEMs (Jaishi and Ren, 2005; Sanayei et al., 2015). Therefore, unsupervised learning methods have recently been receiving more attention for the problems facing structural damage detection in practice.
1.2 Objectives

Compared to the supervised damage detection methods requiring large amounts of data measured from various known scenarios of the monitored structures in their training processes, unsupervised damage detection methods do not need their training data labeled with different damaged structural scenarios. The key concept of the proposed unsupervised damage detection methods in this thesis is that only the data measured from the intact structural scenarios are used to train the adopted statistical models and neural networks.

The primary goal of this thesis is to develop novel unsupervised damage detection methods. To realize this goal, three branch goals are defined as follows:

1. Develop a novel unsupervised damage detection method using advanced deep learning technique.
2. Develop a novel unsupervised damage localization method using the technique of fast-clustering-based novelty detection.
3. Conduct comparative studies for both newly developed methods for validation through numerical and experimental tests with a selection of related up-to-date machine learning- and deep learning-based unsupervised damage detection methods.

1.3 Scope of work

This thesis comprises six chapters. Chapter 1 presents a scientific background of the state-of-the-art data-driven damage detection methods in the research area of SHM, as well as the defined objectives, the scope of research works, the novelties and contributions in this thesis. Chapter 2 provides an extensive literature review of up-to-date data-driven damage detection methods using various machine learning and deep learning techniques in recent years.

Chapter 3 presents a novel unsupervised deep learning approach to detecting structural damage. For details, Section 3.1 develops a novel unsupervised deep learning-based damage detection method with a deep auto-encoder and a one-class support vector machine (OC-SVM). Section 3.2 describes a numerical simulated multi-story frame and a laboratory-scale bridge model to verify the effectiveness of the developed method in
damage detection. Section 3.3 shows the case studies of damage detection with the developed method. Section 3.4 applies the technique of Mahalanobis distance and a recently developed unsupervised deep learning method for comparative studies with the developed method. Section 3.5 states the limitations of the developed unsupervised damage detection method.

Chapter 4 develops an unsupervised novelty detection method to roughly localize damage positions. Specifically, Section 4.1 proposes an unsupervised novelty detection method using an improved technique of density peak-based fast clustering. Section 4.2 introduces two novel damage-sensitive features prepared for generating training and testing datasets for structural damage localization with the developed method. These features are directly extracted from the recorded acceleration data from various structural scenarios. Section 4.3 describes a small-scale steel building frame to verify the effectiveness of the developed method in damage localization. Section 4.4 provides a case study of damage localization with the developed method, and analyzes its advantages and limitations according to the performance in damage localization.

Chapter 5 shows extensive comparative studies of damage detection and localization with several developed unsupervised damage detection methods. For details, Section 5.1 proposes five unsupervised novelty detection methods based on several techniques of machine learning and deep learning reviewed in Chapter 2. Many modifications are made to transform these methods into a uniform unsupervised novelty detection mode, and some improvements are made to enhance their performances in structural damage detection and localization. Section 5.2 provides extensive comparative studies between the five developed unsupervised damage detection methods and the method introduced in Chapter 4, and some pros and cons of these developed methods are listed based on their performance in structural damage detection and localization.

Chapter 6 draws conclusions based on the performance of the methods developed in this thesis with respect to structural damage detection and localization and provides their limitations and possible future works.
1.4 Novelties and contributions

In this thesis, a comprehensive efficient strategy of SHM is proposed for monitoring the health conditions of civil infrastructures. In Chapter 3, a novel unsupervised deep learning approach is originally developed to indicate the global health conditions of monitored structures. The developed approach requires few parametric studies when compared with many existing unsupervised damage detection methods. Besides, the developed approach is effective to detect minor structural damage, which is verified in a comparative study with an up-to-date unsupervised deep learning-based damage detection approach. In Chapter 4, a novel unsupervised novelty detection method using a fast clustering technique is developed to localize different types of damage in the damaged structures. In the case studies, the damage positions in all damaged structural scenarios can be accurately located with the densely sensor network built on a laboratory-scale steel building model. Meanwhile, some effective damage-sensitive features extracted from the measured acceleration responses before and after damage are developed, which are verified to be sensitive to local structural damage. The developed method shows its superiority in structural damage localization in the extensive comparative studies with several unsupervised damage detection methods using various techniques of machine learning and deep learning in Chapter 5.
Chapter 2: Literature review of supervised and unsupervised damage detection methods

This chapter comprises a comprehensive literature review of supervised and unsupervised data-driven methods using various machine learning and deep learning techniques for detecting damage in mechanical and civil structures. Generally speaking, the goal of structural damage detection can be achieved by various means of machine learning and deep learning, such as classification, novelty detection, pattern recognition, etc. (Pimentel et al., 2014). Past research studies applying these techniques for structural damage detection are reviewed in the following sections.
2.1 Distance-based machine learning technique

The technique of nearest-neighbors is an effective tool to implement classification for multi-class data points. The k-nearest neighbor (KNN) algorithm is an effective tool for data classification with a supervised learning manner (Ding et al., 2014; Rogers and Girolami, 2016). The KNN algorithm assumes that there are multi-class data points in a space and that each data point is labeled with a specific class. Due to the similarity of the data points from the same class, these data points should be relatively close to each other and be relatively far from the data points from other classes. An unknown testing point should be assigned to the major class of their number of k nearest points.

The KNN algorithm has been applied in many research studies of structural damage detection (Junior et al., 2018; Li and Sun, 2020). For example, Walsh et al. (2013) utilized the KNN algorithm to detect damage in a real damaged bridge. Specifically, the proposed method in their study used the KNN algorithm to analyze the differences in structural geometry between the deformed structure in its damaged state and the initial undeformed structure in its healthy condition. Their experimental results showed a good performance with respect to structural damage detection, verifying the effectiveness of the KNN algorithm in damage detection. Rafiei and Adeli (2017) employed the KNN algorithm to detect structural damage in the form of cracks in a small-scale reinforced concrete building. By using the damage-sensitive features extracted from the measured acceleration data transformed by the techniques of synchrosqueezed wavelet transform and fast Fourier transform, the KNN algorithm can achieve damage detection with 82% accuracy.

The technique of Mahalanobis distance is another widely used unsupervised distance-based novelty detection tool for damage detection (Figueiredo et al., 2011; Zhou et al., 2015; Sarmadi and Karamodin, 2020). It assumes that a number of data points from a single class can form a cluster in a space and that each data point in this cluster has a short Mahalanobis distance to the cluster centroid. The testing data points from other classes should have relatively long Mahalanobis distances to that centroid; thus, they can be clearly identified as the novelty data points or outliers. For instance, Yeager et al. (2019) utilized the technique of Mahalanobis-squared distance to differentiate the frequency-domain damage-sensitive features in the form of cross-power spectral density from different structural scenarios. Overall, this technique performed a good damage detection
for a glass epoxy pane. Wang et al. (2016) developed a novel approach using the index of root mean square deviation (RMSD) calculated using the Mahalanobis distance technique to evaluate the degrees of damage in timber specimens. Their experimental results showed that a novel index of structural damage (Mahalanobis distance-based RMSD) is superior to the conventional RMSD index in damage detection.

2.2 Clustering-based machine learning technique

The technique of clustering-based machine learning is widely used for classification in unsupervised learning manners such as K-means, fuzzy c-means, and Gaussian mixture model (Yu et al., 2013; Diez et al., 2016; Zhou et al., 2017; Zhou and Yang, 2020). The rule of clustering is based on an assumption that the data points from the same class should form an isolated cluster in a space, at a certain distance from the clusters formed by the data points from other classes (Rogers and Girolami, 2016).

Typically, clustering-based algorithms aim to classify numerous multi-class data points into a predefined number of clusters. The data points being classified into the same cluster are considered to be from the same class. The second step is to check these clusters to determine whether the data points are accurately classified. For example, Gul et al. (2007) integrated a traditional K-means clustering method with the Mahalanobis distance technique to develop a novel damage detection method. The developed method can successfully detect structural damage in the form of changes in the boundary conditions of a simply supported beam. Silva et al. (2007) utilized a fuzzy c-means (FCM) method to detect a number of patterns of structural damage simulated by the FE method in a multi-story steel frame model. Their experimental results showed that five different damage patterns can be detected. Additionally, Silva et al. (2008) utilized the FCM method to train a statistical model for a healthy condition of a small-scale bookshelf structure. Their case study demonstrated that the trained statistical model was able to recognize the abnormal data measured from the damaged scenarios of that structure, and the experimental results proved that the FCM method achieves 85% accuracy of damage detection.

The Gaussian mixture model (GMM) is another widely used unsupervised clustering technique (Nair and Kiremidjian, 2007; Noh et al., 2009; Zhang et al., 2019). In
clustering with a GMM, calculating the statistical probability density distribution for a large amount of data points can divide them into a predefined number of clusters. For example, Farhidzadeh et al. (2013) developed a novel data-drive method with a GMM technique to detect cracks in a reinforced concrete wall that were generated by reversed cyclic loading. Their experimental results validated the effectiveness of the developed method in identifying different extents of the cracks in the reinforced concrete wall. Qiu et al. (2019) proposed a technique of improved density-peaks clustering-based Expectation Maximization to improve the performance of the GMM method for detecting structural damage. A full-scale aircraft fatigue test was conducted to assess the proposed GMM-based method, and their experimental results for detecting cracks in a landing gear spar and a wing panel were stable and reliable in random fatigue load conditions.

2.3 Support vector machine technique

As one branch of support vector machines, OC-SVMs are widely used as novelty detection tools in an unsupervised learning manner (Chang and Lin, 2011; Manevitz and Yousef, 2001; Saari et al., 2019). The purpose of novelty detection with an OC-SVM is to train a large number of data points from a single class in order to obtain an optimal hyperplane able to enclose the majority of these data points, only leaving a small number of data points outside of the enclosed hyperplane as outliers. The trained OC-SVM classifier can be used to identify the testing points sitting outside the enclosed hyperplane as novelty points. For instance, Long and Buyukozturk (2014) utilize an OC-SVM to detect structural damage of bolt loosening and reduced cross section area of structural components in a multi-story steel building model. The results showed that the trained OC-SVM was able to detect damage in the structure with high damage detection accuracy. Khoa et al. (2014) utilized an OC-SVM to detect structural damage in a small-scale multi-story frame and the Sydney Harbor Bridge in Australia. Their experimental studies validated the significant capacity of OC-SVMs in damage detection and that OC-SVMs can function appropriately in the real world. Meanwhile, Lu et al. (2015) used an OC-SVM on a carbon fiber-reinforced plastic plate for structural damage detection, and their experimental results showed a good damage detection performance of over 90% accuracy.
2.4 Limitations of existing works with deep learning

The main difficulty of the data-driven damage detection methods using machine-learning techniques is the extraction of damage-sensitive features from the measured data from the monitored structures, which usually requires an intensive research study to discover effective features in specific applications. In the structural damage detection with data-driven methods, ideal features should be sensitive to the presence of structural damage; meanwhile, they should be insensitive to operational and environmental variability in the normal range. However, this challenging task can be appropriately managed with deep learning techniques, as they can automatically extract useful features from the measured structural data through the training processes of deep neural networks (Cha et al., 2017; Gulgec et al., 2019; Teng et al., 2019).

For example, Abdeljaber et al. (2017) developed a novel adaptive supervised deep learning method with convolutional neural networks (CNNs) to detect damage of bolt loosening in a small-scale steel frame. Based on their experimental results, the developed supervised damage detection with CNNs was effective in detecting loosened bolts with extremely high accuracy, and their proposed method was able to perform real-time damage detection. Lin et al. (2017) also proposed a supervised damage detection approach with CNNs to detect structural damage in a numerical simulated beam. Various damaged structural states were generated by adjusting the height of a small target part in the beam. Their case study indicated that the proposed approach using CNNs performed well in damage detection. Meanwhile, Eren et al. (2018) used a large quantity of measured raw vibrational data from a bearing to detect bearing faults to train an adaptive CNN classifier. The experimental study indicated the effectiveness of the trained CNN classifier in damage detection. As a result, many research studies of damage detection were conducted using CNNs to train efficient classifiers in “supervised” learning mode, and many of them could perform damage detection for civil and mechanical structures well.

As introduced in Chapter 1, even though supervised deep learning approaches usually perform very well in structural damage detection with extremely high detection accuracies, they require sufficient labeled training data from different structural damage scenarios, which makes them unfeasible in the practical applications of structural damage detection. In order to address this main shortcoming of supervised deep learning
approaches for structural damage detection, a small number of unsupervised damage detection methods using deep learning techniques were developed within a few recent years (Rafiei and Adeli, 2018; Ma et al., 2020). For example, Rafiei and Adeli (2018) applied deep restricted Boltzmann machines (DRBMs) to detect the concrete cracks in a laboratory-scale reinforced concrete building model. By using appropriately selected DRBMs, represent features at the hidden layers could be automatically learned from the transformed signals in the frequency domain. Their experimental results demonstrated that the proposed unsupervised damage detection method with DRBMs can successfully detect severe damage of cracks in the building model, but its sensitivity in detecting minor crack damage was quite weak. Their method is applied in a comparative study with my developed unsupervised deep learning method for structural damage detection in Chapter 3.
Chapter 3: Unsupervised deep learning method for structural damage detection

An advanced unsupervised damage detection method by integration of a deep auto-encoder and an OC-SVM developed in my past study for structural damage detection is described in this chapter. To evaluate the effectiveness of the developed method, a numerical multi-story building frame and a small-scale bridge model were prepared for verification. The case studies showed the performance of damage detection for the prepared numerical building frame and the steel bridge model. In addition, the comparative studies of damage detection with the Mahalanobis distance technique and a state-of-the-art unsupervised deep learning approach indicated the effectiveness of my developed method for detecting minor damage in the structures.

* This chapter is developed based on my previous journal publication [Wang, Z. and Cha, Y. J. (2021). Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect structural damage. Structural Health Monitoring, 20(1), 406-425.].
3.1 Deep auto-encoders and OC-SVMs

This section introduces an advanced unsupervised damage detection-based approach that can be applied for evaluating the health conditions of infrastructures. In my study, this unsupervised damage detection approach is implemented by an integrated mechanism with a deep auto-encoder and an OC-SVM. In its training process, only the acceleration responses measured from an infrastructure in its intact scenario are used to train an appropriately selected deep auto-encoder. The measured acceleration responses from other unknown scenarios of the same structure are used to test this trained deep auto-encoder to detect the possible damage in these structural scenarios. An overall flowchart of structural damage detection with the developed approach is presented in Figure 3-1.

![Diagram](image)

**Figure 3-1.** Procedures of damage detection with my developed unsupervised deep learning method (Wang and Cha, 2021).

In Figure 3-1, the solid arrows indicate the training process of a deep auto-encoder using only a group of acceleration responses measured from a known intact/baseline scenario of a structure under a certain extent of external excitations with random noises. The dashed line arrows indicate the testing steps in which the measured acceleration responses from various testing scenarios of that structure are used. The role of a deep auto-encoder in this study is to reconstruct its inputs (i.e., the measured acceleration responses) from the intact/baseline structural scenario through the training processes. In its testing processes, the well-trained deep auto-encoder can reconstruct its inputs with extremely low loss of data reconstruction if the input acceleration responses are measured from the same intact/baseline structural scenario; otherwise, the trained deep auto-encoder will inaccurately reconstruct its inputs with a high loss of data reconstruction. Using metrics,
the differences (errors) between the inputs and outputs of the trained deep auto-encoder are quantified. The quantified differences are considered features that are sensitive to different structural scenarios in the context of this study. Then, an OC-SVM can be utilized to employ novelty detection using these features to identify the acceleration responses measure from the damaged structural scenarios. A detailed description regarding the developed method is available in the following sections.

3.1.1 Data reconstruction with deep auto-encoders

Auto-encoders are traditional, unsupervised neural networks that aim to reconstruct their input data through repeated backpropagation (Hinton, 1990). Typically, an auto-encoder comprises an input layer, a hidden layer, and an output layer. The size of the output layer is the same as that of the input layer for reconstructing inputs. In data reconstruction, an auto-encoder is constituted by an encoder and a decoder. The encoder tends to learn representative features from the inputs and map the learned features in the hidden layer. The decoder forces reconstructing the inputs by using the learned features and mapping them in the output layer. The data reconstruction with an auto-encoder can be mathematically expressed as follows:

\[
\begin{align*}
    h_j &= f(W_jx_i + b_1) \\
    y_i &= g(\hat{W}_{ij}h_j + b_2)
\end{align*}
\]

where \( h_j \) denotes the \( j \)th feature representation in the hidden layer, \( x_i \) and \( y_i \) denote the \( i \)th input and \( i \)th output in the input layer and output layer, respectively, and \( f \) and \( g \) represent the activation functions of the encoder and decoder, respectively. For example, the sigmoid function and hyperbolic tangent function are commonly used activation functions in many studies. \( W \) and \( \hat{W} \) represent the weights for the encoder and decoder for learning feature representations and reconstructing inputs, respectively, and \( b_1 \) and \( b_2 \) represent the biases for the encoder and decoder, respectively.

In data reconstruction with an auto-encoder, a loss or cost function is typically employed to tune the weights and biases for the auto-encoder through repeated
backpropagation. The goal of backpropagating the calculated errors between the inputs and outputs by the cost function in each repetition is minimizing the data reconstruction losses through adjusting the parameters of weights and biases. The mean squared error ($MSE$) is often used as a cost function to quantify the data reconstruction losses of many auto-encoders in their studies. The role of $MSE$ in an auto-encoder can be mathematically expressed as follows:

$$L(W, \tilde{W}, b_1, b_2) = \arg \min_{W, \tilde{W}, b_1, b_2} \frac{1}{k} \sum_{i=1}^{k} \|y_i - x_i\|^2$$  \hspace{1cm} (3-3)

where $k$ denotes the size of the input layer and output layer.

An auto-encoder can be trained by updating its weights and biases through minimizing the cost function expressed in Equation 3-3. For parameter optimization, optimizers of stochastic gradient descent (SGD) are usually applied to tune these parameters with repeated backpropagation in many studies. The parameter updating with an SGD optimizer is mathematically formulated as follows:

$$w(t + 1) = w(t) - \eta \frac{\partial L}{\partial w(t)}$$  \hspace{1cm} (3-4)

where $w$ denotes the updating parameters of weights and biases, $t$ denotes the iteration number, $\eta$ represents the learning rate, and $L$ is the cost function.

In this study, a deep auto-encoder was appropriately selected for reconstructing the measured acceleration responses from the monitored structures, as presented in Figure 3-2. Compared with the above introduced auto-encoders with only one hidden layer, a deep auto-encoder has several hidden layers in its neural network, which enables it to learn more multi-level underlying representations from its complex input data (Hinton and Salakhutdinov, 2006; Vincent et al., 2010; Zhuang et al., 2015). In Figure 3-2, the part from layer L1 to layer L4 represents the encoding part of the deep auto-encoder, and layers L4 to L7 comprise the decoding part of the deep auto-encoder. In its training process through data reconstruction, the inputs of one layer are the outputs of its former layer.
the encoding part, $W_1$, $W_2$, and $W_3$ represent the parameters of weights and biases, and the learned feature representations of the encoder are mapped at layer L4. The decoder with the parameters of $W_4$, $W_5$, and $W_6$ aims to use the learned features at layer L4 to reconstruct the initial input signals, and the final reconstructed outputs are mapped at layer L7.

Figure 3-2. Architecture of neural network for a deep auto-encoder (Wang and Cha, 2021).

In this study, the developed unsupervised damage detection method only used a group of acceleration responses measured from the intact scenario (IS) or baseline of a structure for training a deep auto-encoder, as presented in Figure 3-2. The size of layer L1 equals the total number of the measured acceleration data in a given sampling time period. In the case studies, the mean absolute value was used as a cost function for this deep auto-encoder. The employed cost function for updating parameters of weights and biases in the deep auto-encoder can be mathematically formulated as follows:
where $x_i$ and $y_i$ denote the $i$th input and $i$th output, respectively, $m$ stands for the size of the input layer, $W_j$ and $b_j$ represent the tuning parameters of weights and biases, respectively, for the $j^{th}$ layer, and $n$ represents the total number of parameter groups of $W_j$ and $b_j$.

The presence of structural damage alters the structural properties, which results in variations in dynamical responses. Acceleration response data are widely used in data-driven damage detection methods because they are relatively easy to measure in structural vibration through accelerometers installed on monitored structures. In the case studies, only the acceleration responses measured from the IS of an infrastructure were used as the inputs of a deep auto-encoder, shown in Figure 3-2, in its training processes. When the deep auto-encoder was trained appropriately, it could obtain the reconstructed outputs with an extremely low loss of data reconstruction, as presented in Figure 3-3(a).

In the testing processes of the trained deep auto-encoder, the acceleration responses measured from different structural scenarios of that infrastructure were used as the testing dataset for the trained deep auto-encoder. As presented in Figure 3-3(b), if there are testing acceleration responses measured from a damage scenario (DS) of the same structure, the trained deep auto-encoder outputs the reconstructed acceleration responses with a relatively high level of data reconstruction loss. Furthermore, the acceleration data measured from different DSs should have variation in losses of data reconstruction. The different extents of errors between inputs and reconstructed outputs of the trained deep auto-encoder for different structural scenarios could be properly quantified as damage-sensitive features in this study, which are explained in the subsection below.
3.1.2 Damage-sensitive features

Extracting useful features that are sensitive to structural damage has been a critical part of many data-driven damage detection methods with machine learning techniques in recent decades (Farhat and Hemez, 1993; Li et al., 2006; Cha and Wang, 2016). However, the research studies on using deep neural networks to extract proper features for structural damage detection are few (Lu et al., 2017; Pathirage et al., 2018). The unsupervised deep learning approaches in their studies usually directly used the learned representations of inputs at the hidden layers of deep neural networks as damage-sensitive features, which were used to diagnose the health conditions of their experimental structures. In this study, three indexes of differences between original and reconstructed acceleration data measured from different structural scenarios by a deep auto-encoder, as shown in Figure 3-2, were considered damage-sensitive features. To demonstrate the effectiveness of these
features in damage detection, a comparative study with an advanced unsupervised learning approach directly using learned features at the last hidden layer of DRBMs as damage-sensitive features was conducted (see Section 3.4).

**MSE.** In this study, *MSEs* of damage-sensitive features were used to quantify the differences between inputs and reconstructed outputs of a trained deep auto-encoder, as presented in Figure 3-2. The feature of *MSE* indicates the mean squared error between the original and reconstructed acceleration responses measured from a structure, and it can be mathematically formulated as follows:

\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (a_i - \bar{a}_i)^2
\]  

(3-6)

where *n* represents the number of acceleration data in measured acceleration responses, \(a_i\) denotes the *i*th input data in the original acceleration responses, and \(\bar{a}_i\) denotes the *i*th output data in the reconstructed acceleration responses.

**Original-to-reconstructed-signal ratio (ORSR).** Similar to the widely used signal-to-noise ratio comparing the level of serial signals to the level of noises in it (Welvaert and Rosseel, 2013), a novel feature, the ORSR, was proposed in this study. ORSR represents the ratio in decibels of the sum squared acceleration data for the original acceleration responses to that for the reconstructed acceleration responses. The mathematical expression of ORSR is as follows:

\[
\text{ORSR} = 10 \log_{10} \frac{\sum_{i=1}^{n} a_i^2}{\sum_{i=1}^{n} \bar{a}_i^2}
\]  

(3-7)

where *n* denotes the number of acceleration data in measured acceleration responses, \(a_i\) stands for the *i*th input data in the original acceleration responses, and \(\bar{a}_i\) stands for the *i*th output data in the reconstructed acceleration responses.

**Difference of Arias intensity (DAI).** The index of Arias intensity is commonly used to quantify the energy intensity of the measured acceleration responses in a given time period (Bullock et al., 2017). In this study, the DAI between the original and reconstructed
acceleration responses was considered damage-sensitive feature. The index of Arias intensity represents the time-integral of squared acceleration magnitude of the acceleration responses, and it is calculated as follows:

$$AI = \frac{\pi}{2g} \int_{0}^{T} a(t)^2 dt$$  \hspace{1cm} (3-8)

where $g$ denotes the gravitational acceleration, $a(t)$ stands for the measured acceleration responses, and $T$ denotes the time duration of $a(t)$.

### 3.1.3 Damage detection with OC-SVM

To establish an automatic system for detecting structural damage, an OC-SVM is implemented on the extracted features, introduced in Subsection 3.1.2, for unsupervised damage detection. An OC-SVM is known as an outlier or novelty detector, which aims to use only the training data from a same class to train an OC-SVM classifier. The built OC-SVM classifier can be utilized to identify the novelty data that do not belong to the class of training data (Rogers and Girolami, 2016; Wang and Cha, 2017). In a two-dimensional space, a trained OC-SVM classifier can be shown by an optimal enclosed boundary, which can enclose the training data points in a single class. The testing data points from other classes should be out of the enclosed boundary, and they are identified as novelty points by the trained OC-SVM classifier.

The OC-SVM training work is mathematically expressed by an optimization problem as follows:

$$\min_{w} \frac{1}{2} w^2 + \frac{1}{\nu N} \sum_{i=1}^{N} \xi_i - \rho$$  \hspace{1cm} (3-9)

subject to $(w \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0$

where $\nu$ represents the maximum allowable fraction of abnormal data (outliers) in a single-class training dataset, and the minimum allowable number of training data detected
as support vectors, which are indicated in Figure 3-4, \( N \) stands for the number of training data, \( \xi_i \) represents a slack variable that controls the allowed extents of training error, \( x_i \) denotes the \( i \)th training data, and \( w \) and \( \rho \) represent the terms of margin and bias, respectively.

By applying Lagrangian multiplier and quadratic programming, the minimization problem of Equation 3-9 is converted into a dual form. In addition, a kernel function is utilized for mapping training data to a higher dimensional domain to enable them linearly separable. The converted dual problem is mathematically expressed as follows:

\[
\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K(x_j, x_i) \tag{3-10}
\]

subject to \( 0 \leq \alpha_i \leq \frac{1}{\nu N}, \sum_{i=1}^{N} \alpha_i = 1 \)

where \( \alpha_i \) denotes the Lagrange coefficient of \( x_i \), and \( K(x_j, x_i) \) represents the utilized kernel function. In this study, the Gaussian kernel function, \( K(x, y) = \exp(-\gamma (x - y)^2) \), is utilized, and \( \gamma \) in this function is called kernel parameter, which controls the smoothness of the decision boundary, as presented in Figure 3-4.

![Figure 3-4. Unsupervised damage detection using OC-SVMs (Wang and Cha, 2021).](image)
After the training of an OC-SVM, testing data \( x_t \) from a different class can be identified by the decision function:

\[
f(x_t) = \text{sgn}
\left(\sum_{i=1}^{N} \alpha_i K(x_i, x_t) - \rho\right)
\]

(3-11)

where \( N \) denotes the total number of training data \( x_i \), and \( x_i \) is detected as a support vector if its associated \( \alpha_i \) is non-zero. The decision boundary shown in Figures 3-4 can be mathematically defined by \( \sum_{i=1}^{N} \alpha_i K(x_i, x) - \rho = 0 \). When the sign function \( f(x_t) \) is minus, the testing data points \( x_t \) should be out of the enclosed boundary and identified as a novelty point.

In the case studies, only the feature points from the IS of a structure are used as training points to train an OC-SVM, as shown in Figure 3-4. A trained OC-SVM classifier in the form of a decision boundary should be able to enclose the training points from a single class. The predefined parameters \( \nu \) and \( \gamma \) introduced above control the shape and the smoothness of the decision boundary in the training process of the OC-SVM. Because of the numerical similarity of the features for the same structural scenario, the formed points in a feature domain should be close to each other and form a cluster. At the same time, this cluster should have certain distances to the clusters for other structural scenarios. In the case studies, the testing feature points for all DSs of a structure should be out of the enclosed boundary, which only encloses the training and testing feature points for the IS of the structure, as shown in Figure 3-4.

### 3.2 Numerical and experimental setups

To assess the effectiveness of the developed unsupervised deep learning approach in Section 3.1 in structural damage detection, a numerical multi-story frame and a small-scale bridge model were simulated and assembled, respectively.
3.2.1 Numerical multi-story frame

As presented in Figure 3-5, a 12-story frame was generated by deriving a state-space formulation in a MATLAB environment (Sakellariou and Fassois, 2006). The mass (m) of each floor was 3,000 kg; the height (h) of each story was 4 m. The second-order inertial moment of each column was $5 \times 10^{-3} \text{ m}^4$, and its Young’s modulus was 200 GPa. The critical damping ratio was set to 0.05 in the case studies. The floors were assumed to be rigid bodies in vibration, and they can move in the horizontal direction only. The calculated stiffness of each story (k) was $1.875 \times 10^8 \text{ N/m}$, and the calculated first twelve natural frequencies of this frame were 0.5, 1.5, 2.5, 3.4, 4.3, 5.1, 5.8, 6.4, 7.0, 7.4, 7.7, and 7.9 Hz, respectively.

![Figure 3-5. 12-story building frame (Wang and Cha, 2021).](image)

In order to vibrate the multi-story building frame, a multi-frequency sine wave was generated to be applied at the randomly selected 2nd, 5th, 8th, and 11th floors, as shown in Figure 3-6(a). The acceleration responses of each floor were measured in numerous vibration tests. A multi-frequency sine wave can be created by superimposing several sine waves at different frequencies, and it can create the resonances of structures at different natural frequencies (Napolitano and Linehan, 2009; Deng et al., 2014). The multi-frequency sinusoidal wave can be formulated as the function of time ($t$):
\[ s(t) = \sum_{f=1}^{n} \sin(2\pi ft) \]  

where \( f \) represents the frequency of the sine wave, and \( n \) represents the upper bound on the predefined frequency range. According to the obtained natural frequencies of the multi-story frame, \( n \) is set to 10 Hz in the case studies.

\[ f = \frac{1}{3-13} \]

Figure 3-6. Components of the excitation wave (Wang and Cha, 2021).

In order to simulate the random ambient excitation during the vibration tests, white Gaussian noises were superimposed on the generated multi-frequency sine wave (Sohn, 2007), as illustrated in Figure 3-6(b). In the case studies, the added random white Gaussian noises in each vibration test followed a Gaussian distribution with a mean of 0 and a standard deviation of 0.08. Thus, the level of the added white Gaussian noises was about 3.4–3.7% root-mean-square noise-to-signal ratio.
To prepare sufficient training and testing datasets in the case studies, four DSs of the multi-story building frame were created. In each DS, structural damage was generated by changing the stiffness of the specified floors. The details of damage information and numbers of vibration test are shown in Table 3-1. To investigate the sensitivity of the unsupervised approach to detect different extents of damage, the generated damage range was 10–20% of stiffness reduction at different floors. In addition, the sampling time in each vibration test was 10 s, and the sampling frequency was set to 200 Hz.

Table 3-1. Structural scenarios and numbers of vibration tests (Wang and Cha, 2021).

<table>
<thead>
<tr>
<th>Structural scenario</th>
<th>Test number</th>
<th>Damage type and location</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>500</td>
<td>No damage</td>
</tr>
<tr>
<td>DS1</td>
<td>200</td>
<td>Stiffness reduced by 10% at 6th floor</td>
</tr>
<tr>
<td>DS2</td>
<td>200</td>
<td>Stiffness reduced by 15% at 8th floor</td>
</tr>
<tr>
<td>DS3</td>
<td>200</td>
<td>Stiffness reduced by 20% at 10th floor</td>
</tr>
<tr>
<td>DS4</td>
<td>200</td>
<td>Stiffness reduced by 10% at 2nd and 4th floors</td>
</tr>
</tbody>
</table>

3.2.2 Laboratory-scale bridge model
To assess the damage detection performance of the developed unsupervised deep learning method for real structures, a small-scale steel bridge (Figure 3-7) was fabricated (SBDTUM, 2017). The length, width, and height of the steel bridge were 6.0, 0.6, and 0.72 meters, respectively. In the construction of this bridge model, steel bolts were used to connect the major structural components at the primary structural joints. In its vibration tests, a rubber hammer was utilized to excite the steel bridge model, and the excitation location is indicated in Figure 3-8. A sensor network of 10 accelerometers was attached to the steel bridge at randomly selected major structural joints to measure their acceleration responses during the vibration tests. All accelerometers were single-axial, which only recorded the acceleration response data in the vertical direction. The sampling
frequency in the vibration tests was set to 2,000 Hz, and the sampling time was 1.0 s during each vibration test.

Figure 3-7. Small-scale steel bridge (Wang and Cha, 2021).

Figure 3-8. Deployment of the sensors installed on the steel bridge (Wang and Cha, 2021).

For the purpose of data acquisition, one IS and four DSs for the steel bridge were created to collect sufficient training and testing data. Bolt loosening is a common type of structural damage occurring within real steel bridges. In the experiments of this study, the bolts at randomly selected primary structural joints were loosened to simulate structural damage in the steel bridge. The numbers of vibration tests and the information of the created DSs are included in Table 3-2.
Table 3-2. Various structural scenarios and numbers of vibration tests (Wang and Cha, 2021).

<table>
<thead>
<tr>
<th>Structural scenario</th>
<th>Test number</th>
<th>Damage type and location</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>250</td>
<td>No damage</td>
</tr>
<tr>
<td>DS1</td>
<td>90</td>
<td>Bolts loosened adjacent to Sensor #2</td>
</tr>
<tr>
<td>DS2</td>
<td>78</td>
<td>Bolts loosened adjacent to Sensor #4</td>
</tr>
<tr>
<td>DS3</td>
<td>85</td>
<td>Bolts loosened adjacent to Sensor #6</td>
</tr>
<tr>
<td>DS4</td>
<td>78</td>
<td>Bolts loosened adjacent to Sensor #8</td>
</tr>
</tbody>
</table>

3.3 Case studies of damage detection

In this section, the simulated multi-story frame and the small-scale bridge model introduced in Section 3.2 were utilized to access the performances of the developed unsupervised deep learning approach in damage detection. Sufficient training and testing datasets could be obtained from the illustrated structural scenarios by numerous vibration tests, as shown in Tables 3-1 and 3-2. The CPU of the computer used in the case studies was the Intel Core i7-7700hq with 16GB of RAM. The cases of damage detection for each floor in the building frame and each sensor position in the steel bridge were independently carried out. In order to minimize the adverse effects of variation loading and ambient excitations during the vibration tests, the measured raw acceleration responses from various structural scenarios needed to be normalized before they were input into a deep auto-encoder for training and testing (Nair et al., 2006). The normalization of acceleration responses is mathematically formulated as follows:

\[ \tilde{a}(t) = \frac{a(t) - \mu}{\sigma} \]  

(3-14)

where \( a(t) \) represents the raw measured acceleration responses, \( \tilde{a}(t) \) is the normalized acceleration responses, and \( \mu \) and \( \sigma \) denote the mean and standard deviation of \( a(t) \), respectively.
3.3.1 Case study of the multi-story building frame

The first case study with the developed unsupervised damage detection method is for the simulated 12-story frame, which is shown in Figure 3-5. According to the introduction in Section 3.1, the developed method used only the acceleration data measured from the IS of this building frame as training data. Considering the over-trained (over-fitting) issue of a deep auto-encoder caused by excessive amounts of data from the IS, the 500 samples of acceleration data measured from the IS tests (Table 3-1) were partitioned into 300 samples and 200 samples for training and testing, respectively. The samples of acceleration data measured from the DSs (i.e., DS1–DS4) tests were used as the testing dataset.

To investigate the effects of deep auto-encoders with different layers in data reconstruction, the training data for the 1st floor of the simulated multi-story frame was used for training five deep auto-encoders at different depths. Their performances of data reconstruction are presented in Figure 3-9. The parameters of biases in these deep auto-encoders were all initialized to 0, and a linear activation function was utilized on each layer for encoding and decoding. The total number of training epochs was set to 300, and the mini-batch size of the training data was set to 30. The learning rate was set to 0.01. In Figure 3-9, the curves in different colors indicate the data reconstruction performances of the deep auto-encoders at different depths. Overall, the losses of data reconstruction decreased to low levels when the number of training epochs increased to 250 for all deep auto-encoders.
In regard to specific details, the curves of reconstruction loss presented in Figure 3-9 indicate that the deep auto-encoders having five and seven layers were able to reconstruct their inputs very well with a stable performance and extremely low loss of data reconstruction after 250 training epochs. Therefore, deep auto-encoders having five to seven layers were appropriate for data reconstruction of the measured acceleration responses in the case studies. It is known that deeper neural networks can learn more underlying feature representations from complex inputs (Hinton and Salakhutdinov, 2006). Thus, a deep auto-encoder having seven layers was applied in the case studies, and the size of its input layer was the same as the length of the measured acceleration responses in each vibration test. The sizes of layers L1–L7 of the deep auto-encoder in Figure 3-2 were 2,000, 1,000, 600, 500, 600, 1,000, and 2,000, respectively.

In the cases of damage detection for all floors of the building frame, the prepared training dataset for each floor was separately input to the carefully selected deep auto-encoder for training. Then, the prepared testing datasets for the IS and four DSs were used to test the trained deep auto-encoders. In Figure 3-10, the performances of data reconstruction for all training and testing datasets were clearly shown by the curves of reconstruction loss as the number of epochs increased. As shown, after 250 training epochs, the level of reconstruction loss for the training dataset became stable with an extremely small value. Thus, the predefined 300 training epochs were sufficient to train this deep auto-encoder, and the performance of the trained deep auto-encoder was stable in data reconstruction in the case studies. Moreover, the gap between the data reconstruction curves of the training and testing datasets for the same IS was very small, and there were variations in the extents of the gaps between the curves for the IS and the curves for all four DSs.
Figure 3-10. Data reconstruction losses of the datasets for the 3rd floor of the multi-story frame (Wang and Cha, 2021).

From the performances of data reconstruction shown in Figure 3-10, we can determine that the relatively high reconstruction losses for all four DSs and the extremely small reconstruction losses for the IS cases were useful to reflect the health conditions of the testing structural scenarios. Thus, appropriate indexes of differences between the inputs and reconstructed outputs for the training and testing datasets can be quantified as damage-sensitive features, which are described in Subsection 3.1.2. First, the extracted features of MSE and ORSR introduced by Equations 3-6 and 3-7 were mapped on a two-dimensional (2D) plane to form feature points. The distributions of the corresponded feature points to the training and testing datasets for the IS and the four DSs are shown in Figure 3-11.
Figure 3-11. Distributions of formed feature points on a 2D plane for all floors of the multi-story frame (Wang and Cha, 2021).
Due to the close values of the features of MSE and ORSR for a same structural scenario, the distributions of the feature points on the 2D plane for all floors shown in Figure 3-11 indicate that the clusters formed by the training and testing feature points from the same IS overlapped completely. Meanwhile, the gaps between the clusters of the IS and four DSs were clear. Based on the obvious gaps among the distributions of clusters for different structural scenarios, the objective of damage detection can be visually achieved on the 2D feature plane. Considering the damage detection for high-dimensional invisible feature datasets prepared by mapping more types of features, an OC-SVM was applied on these feature points to perform unsupervised damage detection.

Figure 3-12 presents an example of a damage detection case on the 2nd floor of a building frame. This figure illustrates how an OC-SVM (introduced in Subsection 3.1.3) is applied to the formed training and testing points of the IS and four DSs in this feature domain, thus indicating that the two extracted features (MSE and ORSR) in different measurement scales will influence the damage detection performances of the OC-SVM. Therefore, before the application of the OC-SVM on these training and testing points, they have to be normalized into the same scale of [0–1].

![Figure 3-12. Damage detection with an OC-SVM at the 2nd floor of the multi-story frame (Wang and Cha, 2021).](image)
In Figure 3-12, only the training points of IS were used to train the OC-SVM, and an enclosed decision boundary could be determined by the trained OC-SVM classifier, as described in Subsection 3.1.3. The performance of damage detection in the figure shows that all the testing points of the IS were in the decision boundary as the training points of the IS were because they were from the same structural scenario. The majority of testing points of all four DSs were out of the enclosed boundary, leaving only a small part of testing points of DS3 sitting inside of the enclosed boundary.

The results of damage detection cases at all floors of the multi-story frame are presented in Table 3-3. As noted in Subsection 3.1.3, the two parameters $\nu$ and $\gamma$ in the OC-SVM determined the shape and smoothness of the decision boundary. In the case studies, the parameter $\nu$ introduced in Equation 3-9 was set to 0.0033 for the damage detection cases of all floors in order to encourage an extremely small number of outliers in the training points. I applied the heuristics approach to investigate the sensitivity of parameter $\gamma$. The variations of parameter $\gamma$ in certain ranges are shown in Table 3-3. The damage-detection performance was stable and high in this sensitivity analysis of tuning parameter $\gamma$. The tuned values of kernel parameter $\gamma$ for all damage detection cases are shown in the table.

Table 3-3. Damage detection with OC-SVMs for all floors of the multi-story frame (Wang and Cha, 2021).

<table>
<thead>
<tr>
<th>Building frame</th>
<th>Tuned $\gamma$</th>
<th>IS</th>
<th>DS1</th>
<th>DS2</th>
<th>DS3</th>
<th>DS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1\textsuperscript{st} floor</td>
<td>170–190</td>
<td>97.0%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>2\textsuperscript{nd} floor</td>
<td>30–50</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>89.5%</td>
<td>100%</td>
</tr>
<tr>
<td>3\textsuperscript{rd} floor</td>
<td>190–210</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>4\textsuperscript{th} floor</td>
<td>20–40</td>
<td>98.0–98.5%</td>
<td>100%</td>
<td>100%</td>
<td>96.5–100%</td>
<td>100%</td>
</tr>
<tr>
<td>5\textsuperscript{th} floor</td>
<td>35–55</td>
<td>99.0–100%</td>
<td>100%</td>
<td>100%</td>
<td>84.0–100%</td>
<td>100%</td>
</tr>
<tr>
<td>6\textsuperscript{th} floor</td>
<td>180–200</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>7\textsuperscript{th} floor</td>
<td>210–230</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>1.5–3.0%</td>
<td>100%</td>
</tr>
</tbody>
</table>
The proposed unsupervised deep learning method performed very well in damage detection, and it provided extremely high accuracies of damage detection. Overall, a mean average accuracy of 97.4% for all damage detection cases was achieved. The individual average accuracies of all floor cases were 98.5–99.2, 96.9–99.0, 100, 89.3–91.0, and 100% for the testing scenarios of IS, DS1, DS2, DS3, and DS4, respectively. According to the calculated damage detection accuracies listed in Table 3-3, the developed unsupervised deep learning approach was robust in its performance of damage detection when it was applied to the numerical multi-story building frame. The robustness of the developed approach to detect structural damage at different locations of the monitored structures was further evaluated on a real small-scale steel bridge described in Subsection 3.2.2.

### 3.3.2 Case study of the small-scale steel bridge

To further verify the effectiveness of the developed unsupervised damage detection method on real structures, a small-scale steel bridge introduced in Subsection 3.2.2 was used for its application of damage detection in practice. Similar to the application on the multi-story building frame, the cases of damage detection were independently performed for each sensor attached to the bridge model. In the preparation of training and testing datasets, the 250 samples of acceleration data measured from the IS tests listed in Table 3-2 were partitioned into 200 samples for training and 50 samples for testing. The samples of acceleration data measured from the DSs (i.e., DS1–DS4) tests were used for the testing dataset.

In the training process of the deep auto-encoder presented in Figure 3-2, the training dataset for the IS was input into the deep auto-encoder for data reconstruction. The architecture of the deep auto-encoder was the same as the one used in the case studies.
of the building frame. The differences in the predefined parameters for training were that the number of training epochs was set to 500, and the mini-batch size of the training data was set to 20. After training, the testing datasets for the IS and all four DSs were used to test the trained deep auto-encoder. For the training and testing datasets measured by Sensor #5, the curves of reconstruction loss for different structural scenarios are shown in Figure 3-13. The gap between the curves of the training and testing datasets for the same IS was small, and the losses of data reconstruction for all structural scenarios can be achieved to a stable state after 400 training epochs. In addition, the losses of data reconstruction for the testing datasets for all DSs were considerable, which means they can be used to extract useful features introduced in Subsection 3.1.2.

![Figure 3-13. Data reconstruction loss of the datasets from Sensor #5 on the steel bridge (Wang and Cha, 2021).](image)

In feature extraction, the features of MSE and ORSR introduced in Section 3.1.2 were first extracted from the errors between the inputs and reconstructed outputs of the trained auto-encoder for all training and testing datasets. Then, the extracted features were mapped on a 2D feature plane for visualization, as shown in Figure 3-14. Compared with the clear gaps among the formed clusters for different damage scenarios in Figure 3-11, the gaps among the clusters in Figure 3-14 were unobvious because of the existence of
many noisy points. It is inevitable to have a certain extent of noises in the practical application since the effect of uncertain ambient excitations during the data acquisition for real infrastructures. An OC-SVM was utilized to these formed feature points for the visualization of damage detection. The calculated accuracies of damage detection for all 10 sensors attached to the bridge are presented in Table 3-4. In all damage detection cases, the parameter $v$ was set to 0.005 for the case studies. The tuned values of the kernel parameter $\gamma$ are shown in Table 3-4 as well.

**Table 3-4.** Damage detection with OC-SVMs on a 2D feature plane for all sensors of the steel bridge (Wang and Cha, 2021).

<table>
<thead>
<tr>
<th>Small-scale steel bridge</th>
<th>Tuned kernel parameter $\gamma$</th>
<th>Damage detection accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IS</td>
</tr>
<tr>
<td>Sensor #1</td>
<td>2</td>
<td>86.0%</td>
</tr>
<tr>
<td>Sensor #2</td>
<td>3</td>
<td>90.0%</td>
</tr>
<tr>
<td>Sensor #3</td>
<td>3</td>
<td>84.0%</td>
</tr>
<tr>
<td>Sensor #4</td>
<td>15</td>
<td>96.0%</td>
</tr>
<tr>
<td>Sensor #5</td>
<td>12</td>
<td>100%</td>
</tr>
<tr>
<td>Sensor #6</td>
<td>16</td>
<td>96.0%</td>
</tr>
<tr>
<td>Sensor #7</td>
<td>13</td>
<td>100%</td>
</tr>
<tr>
<td>Sensor #8</td>
<td>22</td>
<td>98.0%</td>
</tr>
<tr>
<td>Sensor #9</td>
<td>14</td>
<td>82.0%</td>
</tr>
<tr>
<td>Sensor #10</td>
<td>2</td>
<td>96.0%</td>
</tr>
<tr>
<td><strong>Average (85.8%)</strong></td>
<td></td>
<td>92.8%</td>
</tr>
</tbody>
</table>
Figure 3-14. Distributions of feature points and damage detection with OC-SVMs on a 2D plane for 10 sensors of the steel bridge (Wang and Cha, 2021).
The distributions of feature points in Figure 3-14 showed that the training and testing clusters for the same IS overlapped to a certain extent; the testing clusters for all four DSs had certain distances to them. The calculated accuracies of damage detection listed in Table 3-4 showed good performances of the developed unsupervised deep learning approach in practical damage detection, and a mean averaged accuracy of 85.8% for all cases was achieved. The averaged accuracies of all installed sensors were 92.8, 64.7, 74.1, 98.2 and 99.1% for IS, DS1, DS2, DS3, and DS4, respectively.

To investigate the performance of using more types of features in damage detection, the feature DAI introduced in Subsection 3.1.2 was used with the previous two features, MSE and ORSR. The damage detection of the steel bridge was reinvestigated with OC-SVMs to perform novelty detection on this new combination of features. The distributions of feature points formed by mapping these extracted features into a 3D feature space are presented in Figure 3-15. The performances of damage detection with OC-SVMs for all sensors are presented in Figure 3-15 as well. The calculated accuracies of damage detection for all testing cases and the tuned values of kernel parameter $\gamma$ are listed in Table 3-5.
Figure 3-15. Distributions of feature points and damage detection with OC-SVMs in a 3D space for 10 sensors of the steel bridge (Wang and Cha, 2021).
Table 3-5. Damage detection with OC-SVMs in a 3D feature space for all sensors of the steel bridge (Wang and Cha, 2021).

<table>
<thead>
<tr>
<th>Small-scale steel bridge</th>
<th>Tuned kernel parameter $\gamma$</th>
<th>Damage detection accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IS</td>
</tr>
<tr>
<td>Sensor #1</td>
<td>1.6</td>
<td>86.0%</td>
</tr>
<tr>
<td>Sensor #2</td>
<td>2.1</td>
<td>90.0%</td>
</tr>
<tr>
<td>Sensor #3</td>
<td>3.4</td>
<td>90.0%</td>
</tr>
<tr>
<td>Sensor #4</td>
<td>26</td>
<td>94.0%</td>
</tr>
<tr>
<td>Sensor #5</td>
<td>9</td>
<td>92.0%</td>
</tr>
<tr>
<td>Sensor #6</td>
<td>19</td>
<td>90.0%</td>
</tr>
<tr>
<td>Sensor #7</td>
<td>53</td>
<td>98.0%</td>
</tr>
<tr>
<td>Sensor #8</td>
<td>73</td>
<td>96.0%</td>
</tr>
<tr>
<td>Sensor #9</td>
<td>22</td>
<td>72.0%</td>
</tr>
<tr>
<td>Sensor #10</td>
<td>2.6</td>
<td>100%</td>
</tr>
<tr>
<td><strong>Average (91.0%)</strong></td>
<td></td>
<td>90.8%</td>
</tr>
</tbody>
</table>

As shown in Table 3-5, compared with the results of damage detection presented in Table 3-4 using two types of damage-sensitive features, the proposed unsupervised deep learning method using three types of features resulted in enhanced damage detection performances. The mean averaged accuracy for all cases reached 91.0%, and the averaged accuracies of all sensors were 90.8, 78.7, 86.3, 99.4, and 100% for IS, DS1, DS2, DS3, and DS4, respectively.

The distributions of feature points shown in Figures 3-14 and 3-15 clearly indicate that the training and testing clusters for the same IS properly overlapped to a certain extent, which produced a reasonable result due to the presence of noises in practice. Overall, the feature points from a same structural scenario could form an isolated cluster, even though the gaps among a small number of clusters for different structural scenarios were unclear, which adversely affected the performances of OC-SVMs in damage detection. The calculated accuracies of damage detection with OC-SVMs are listed in Tables 3-4 and 3-5. The experiment results proved that the developed unsupervised damage detection
approach is effective to detect damage in the form of bolt loosening in the steel bridge. However, in a few subfigures of Figure 3-14, the features points from some testing DSs could not form isolated clusters, and gaps among the clusters from different structural scenarios were unobvious. Nevertheless, as more types of damage-sensitive features were used in damage detection, these adverse conditions were greatly improved, as shown in Figure 3-15. Compared with the performance of damage detection for the simulated 12-story frame, my developed unsupervised deep learning method needs to be improved when it is applied for application on the infrastructures in practice.

3.4 Comparative studies

First of all, to evaluate the effectiveness of the OC-SVMs within the developed unsupervised deep learning method in damage detection, the Mahalanobis distance technique was utilized to perform damage detection on the features MSE, ORSR, and DAI for the small-scale steel bridge. It is known that the Mahalanobis distance metric is widely applied to solve the problem of unsupervised damage detection due to its calculation efficiency (Figueiredo et al., 2011; Sarmadi and Karamodin, 2020). Statistically speaking, the Mahalanobis distance metric is capable of measuring the similarity between two multivariate matrixes. In the case studies, the training feature dataset of the IS, including \( n \) feature vectors of \( p \)-dimensional feature elements, can be assumed to be the training matrix \( X \in \mathbb{R}^{p \times n} \), and its multivariate mean vector and covariance matrix can be represented by \( \bar{x} \in \mathbb{R}^{p \times 1} \) and \( \Sigma \in \mathbb{R}^{p \times p} \), respectively. In the case studies of the steel bridge, \( n \) denotes the number of training feature points of the IS, which was 200, and \( p \) denotes the number of feature types, which was 3. The calculation of Mahalanobis distance between feature vectors of the training matrix \( X \) and a testing feature vector \( y \) in the testing dataset can be calculated as follows:

\[
d_m = (y - \bar{x})^T \Sigma^{-1} (y - \bar{x})
\]  

(3-15)

where \( d_m \) represents the squared Mahalanobis distance.
In damage detection with the Mahalanobis distance technique, it can be assumed that a test feature vector \( y \) from a DS of an infrastructure should be far from \( \bar{x} \) for the training matrix from the IS of that infrastructure, which can imply the presence of damage in the infrastructure. The Mahalanobis distances from the training and testing points to the cluster centroid of the training points of IS, as indicated Figure 3-16, were calculated by using Equation 3-15 for the damage detection cases at Sensor #1 of the steel bridge. It is obvious that the calculated Mahalanobis distances for the training and testing points of the same IS were shorter than the Mahalanobis distances for the testing points of all testing DSs, which demonstrated the effectiveness of the Mahalanobis distance technique for damage detection.

**Figure 3-16.** Damage detection with the Mahalanobis distance technique at Sensor #1 of the steel bridge (Wang and Cha, 2021).

The main weakness of the Mahalanobis distance technique in this study is that the calculated Mahalanobis distances for the testing points of the IS were a bit longer than that for the training points of the same IS, which might be provide a false warning of damage. To calculate the accuracies of damage detection with the Mahalanobis distance technique, a dashed line was plotted by setting a threshold value as shown in Figure 3-16, which was defined by the 95% cut-off over the training points of IS (Figueiredo et al., 2011). In the
calculation of damage detection accuracy, the testing points above the dashed line threshold were considered the novelty points from a DS. The obtained damage detection accuracies for all testing cases are shown in Table 3-6. Overall, the mean average accuracy of damage detection was 86.1%, which was lower than the accuracy of 91.0% shown in Table 3-5 as a result of using OC-SVMs.

**Table 3-6.** Damage detection with the Mahalanobis distance technique using the 3D feature dataset for the steel bridge (Wang and Cha, 2021).

<table>
<thead>
<tr>
<th>Small-scale steel bridge</th>
<th>Damage detection accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IS</td>
</tr>
<tr>
<td>Sensor #1</td>
<td>20.0%</td>
</tr>
<tr>
<td>Sensor #2</td>
<td>68.0%</td>
</tr>
<tr>
<td>Sensor #3</td>
<td>42.0%</td>
</tr>
<tr>
<td>Sensor #4</td>
<td>92.0%</td>
</tr>
<tr>
<td>Sensor #5</td>
<td>46.0%</td>
</tr>
<tr>
<td>Sensor #6</td>
<td>92.0%</td>
</tr>
<tr>
<td>Sensor #7</td>
<td>98.0%</td>
</tr>
<tr>
<td>Sensor #8</td>
<td>94.0%</td>
</tr>
<tr>
<td>Sensor #9</td>
<td>72.0%</td>
</tr>
<tr>
<td>Sensor #10</td>
<td>66.0%</td>
</tr>
<tr>
<td><strong>Average (86.1%)</strong></td>
<td><strong>69.0%</strong></td>
</tr>
</tbody>
</table>

Furthermore, a state-of-the-art unsupervised deep learning approach developed by Rafiei and Adeli (2018) was utilized on the small-scale steel bridge for a comparative study with my developed method in this chapter. A neural network model of deep restricted Boltzmann machine (DRBM) was employed in their study to evaluate the health conditions of a laboratory-scale multi-story frame model. The acceleration responses measured from the accelerometers installed on the multi-story building were used to extract effective features to indicate the health conditions of the building model. Their
experimental results showed that their proposed method can successfully detect severe and near-collapse damage in the building model, but its detection performances of minor and moderate damage were weak.

Specifically, there were five steps in their proposed method to evaluate the health conditions of the multi-story building, which was divided into several substructures for assessment. In the first step, two datasets of acceleration responses were measured from each substructure in its healthy and unknown states, and they were represented by $H_i$ and $U_j$, respectively. In the second step, the technique of synchrosqueezed wavelet transform was used to denoise the measured raw acceleration responses in $H_i$ and $U_j$. In the third step, the fast Fourier transform technique was utilized to transform the denoised acceleration responses in the frequency domain. In the fourth step, a model of DRBM was utilized to learn features from the transformed data in $H_i$ and $U_j$.

Specifically, a restricted Boltzmann machine (RBM) is an artificial neural network with an input layer and a hidden layer, and the learned feature representations from the inputs were mapped on its hidden layer. The architecture of a DRBM comprises an input layer and multiple hidden layers of RBM, and it typically consists of an encoder and a decoder for learning features. The encoder of a DRBM intends to learn feature representations from inputs layer by layer. The decoder of a DRBM is symmetric to its corresponding encoder for the reconstruction of the learned features in the hidden layers. In the encoder of a DRBM, the learned feature representations in a hidden layer are the inputs of its next hidden layer. In the fifth step, a structural health index (SHI) was developed to diagnose the health conditions of each substructure, which was calculated by an equation as follows:

$$\text{SHI} = \exp\left(\frac{-\left\| \frac{1}{N_H} \sum_{i=1}^{N_H} H_i - \frac{1}{N_U} \sum_{j=1}^{N_U} U_j \right\|^2}{2}\right)$$

(3-16)

where $H_i$ and $U_j$ denote the learned features at the last hidden layer of a DRBM encoder for the healthy and unknown structural states, respectively. $N_H$ and $N_U$ denote the total
number of feature vectors in $\overline{H}_i$ and $\overline{U}_j$, respectively. In their study, the elements of feature representation in the respective feature vectors of $\overline{H}_i$ and $\overline{U}_j$ needed to be normalized with a mean value of 0 and a standard deviation of 1. We can see that $U_j$ will be close to $H_i$ if the unknown state is healthy. Moreover, a probability density function, presented in Equation 3-16, is used to calculate the values of SHI; thus, the SHI values should have been in the range $0–1$. Thus, the healthier the infrastructure is, the closer its value of SHI is to 1.

According to the introduced damage detection procedures of the unsupervised deep learning method proposed by Rafiei and Adeli (2018), there are differences in feature extraction between their and my developed methods. As introduced in Section 3.1, only the acceleration responses measured from an IS are required for training a deep autoencoder. The acceleration datasets from different scenarios are used for testing the trained deep auto-encoder. In addition, the indexes of difference between the inputs and reconstructed outputs are considered damage-sensitive features. In the damage detection method of Rafiei and Adeli (2018), all the acceleration data measured from the IS and various unknown scenarios were fed into a model of DRBM to learn feature representations from its inputs, and the final learned features at the last hidden layer were used as damage-sensitive features.

In the application of the method of Rafiei and Adeli, on the small-scale bridge model presented in Figure 3-7, the prepared training and testing datasets in this study were used as $H_i$ and $U_j$, respectively. A model of DRBM with two hidden layers was used in the case studies. The size of the $m$th hidden layer was denoted by $J_m$, and the size of the last hidden layer $J_2$ was set to 3. The number of learning epochs was denoted by $E$, and two values of $E$ were investigated in the case studies: $E = 10$ and $E = 20$. The learning process of DRBM model was run 100 times, the parameters of weights and biases were randomly initialized during each time, and the averaged calculated SHI was used as the final SHI.

The calculated local SHIs for all sensors installed on the steel bridge and their averages as the global SHIs are listed in Table 3-7. According to the calculated SHIs for the cases of IS and four DSs, it is clear that all the global SHIs were in the range $0.97–0.99$ for all cases. Therefore, the experimental results cannot accurately indicate the health
conditions of the steel bridge. Rafiei and Adeli (2018) provided the four ranges of global SHIs for different extents of structural damage in their case studies: 0.89–0.96 for light damage, 0.44–0.72 for moderate damage, 0.03–0.26 for severe damage, and 0.02–0.25 for near-collapse damage. In addition, due to the lack of testing cases for the IS of the small-scale building model in their case studies, they could not provide a reasonable range of global SHI for the IS of that structure. The calculated global SHIs shown in Table 3-7 were beyond the suggested range of 0.89–0.96 for light damage, so their developed unsupervised deep learning approach was not able to detect light damage in the form of bolt loosening in the steel bridge in this study.

Table 3-7. Calculated global and local SHIs with a DRBM model for damage detection on the steel bridge (Wang and Cha, 2021).

<table>
<thead>
<tr>
<th>Sensor</th>
<th>IS</th>
<th>DS1</th>
<th>DS2</th>
<th>DS3</th>
<th>DS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor #1</td>
<td>0.986/0.993</td>
<td>0.981/0.983</td>
<td>0.986/0.986</td>
<td>0.992/0.993</td>
<td>0.988/0.987</td>
</tr>
<tr>
<td>Sensor #2</td>
<td>0.990/0.991</td>
<td>0.989/0.993</td>
<td>0.990/0.993</td>
<td>0.960/0.974</td>
<td>0.984/0.988</td>
</tr>
<tr>
<td>Sensor #3</td>
<td>0.984/0.993</td>
<td>0.986/0.986</td>
<td>0.988/0.991</td>
<td>0.979/0.986</td>
<td>0.977/0.979</td>
</tr>
<tr>
<td>Sensor #4</td>
<td>0.987/0.991</td>
<td>0.990/0.991</td>
<td>0.985/0.988</td>
<td>0.991/0.993</td>
<td>0.987/0.988</td>
</tr>
<tr>
<td>Sensor #5</td>
<td>0.982/0.983</td>
<td>0.989/0.991</td>
<td>0.993/0.993</td>
<td>0.991/0.993</td>
<td>0.971/0.984</td>
</tr>
<tr>
<td>Sensor #6</td>
<td>0.981/0.990</td>
<td>0.980/0.993</td>
<td>0.984/0.991</td>
<td>0.983/0.988</td>
<td>0.901/0.939</td>
</tr>
<tr>
<td>Sensor #7</td>
<td>0.968/0.985</td>
<td>0.981/0.981</td>
<td>0.986/0.991</td>
<td>0.986/0.986</td>
<td>0.974/0.983</td>
</tr>
<tr>
<td>Sensor #8</td>
<td>0.967/0.973</td>
<td>0.987/0.992</td>
<td>0.989/0.987</td>
<td>0.982/0.984</td>
<td>0.983/0.991</td>
</tr>
<tr>
<td>Sensor #9</td>
<td>0.986/0.988</td>
<td>0.993/0.995</td>
<td>0.993/0.991</td>
<td>0.990/0.992</td>
<td>0.989/0.988</td>
</tr>
<tr>
<td>Sensor #10</td>
<td>0.986/0.991</td>
<td>0.988/0.990</td>
<td>0.989/0.992</td>
<td>0.972/0.982</td>
<td>0.953/0.970</td>
</tr>
<tr>
<td>Average</td>
<td>0.982/0.988</td>
<td>0.986/0.989</td>
<td>0.988/0.990</td>
<td>0.983/0.987</td>
<td>0.971/0.980</td>
</tr>
</tbody>
</table>

3.5 Limitations

Compared with the Mahalanobis distance technique and an advanced unsupervised damage detection approach, my developed unsupervised damage detection approach with a deep
auto-encoder and an OC-SVM as described in this chapter proved to be more effective in detecting minor damage of loosened bolts in a laboratory-scale steel bridge model. According to the distributions of feature points for all floors of the building frame and the sensors in the steel bridge as shown in Figures 3-11, 3-14, and 3-15, the features of MSE, ORSR, and DAI were sensitive to the global health conditions of the monitored structures but indicated limited information about damage positions. Thus, the main limitation of the developed method articulated in this study is that it is unable to locate damage positions in the prepared simulated 12-story frame and the small-scale steel bridge.
Chapter 4: Unsupervised fast novelty detection for damage localization

Although an advanced unsupervised damage learning method with a deep learning technique was developed in Chapter 3 to detect damage in infrastructures, information on its performance in damage localization is still lacking. The goal of this chapter is to develop a novel unsupervised novelty detection method for structural damage detection and localization on a laboratory-scale steel building frame. Additionally, two novel damage-sensitive features directly extracted from the measured acceleration responses are proposed to verify the effectiveness of my developed method for structural damage detection and localization.

* This chapter is developed based on my previous journal publication [Cha, Y.J. and Wang, Z. (2018). Unsupervised novelty detection-based structural damage localization using a density peaks-based fast clustering algorithm. *Structural Health Monitoring*, 17(2), 313–324].
4.1 Unsupervised novelty detection method

An up-to-date density-based fast clustering algorithm was originally proposed by Rodriguez and Laio (2014) for clustering analysis. The clustering principle of their proposed algorithm was based on two assumptions. The first assumption was that the centroid of a cluster of data points is surrounded by numerous data points with a low local density. In their study, the local density of a point is an index of the number of its neighboring points. The second assumption was that there are relatively long distances among the centroids of the classified clusters. In my study, a density peak-based fast clustering (DPFC) method for unsupervised damage detection and localization was developed based on the introduced fast clustering algorithm (Rodriguez and Laio, 2014). Besides, the developed DPFC method was modified into the mode of unsupervised novelty detection for structural damage detection and localization in the case studies.

The procedures of the developed DPFC algorithm are described as follows:

Step 1. Calculate local density of each point in a large number of data points for clustering.

The data points in the context of this study are composed of damage-sensitive features (see Section 4.2) that are extracted from the measured structural responses (i.e., acceleration in time history). The local density, \( \rho_i \), of a point \( i \) is defined as:

\[
\rho_i = \sum_j \mathbb{N}(d_{ij} - d_c)
\]  

(4-1)

where \( d_c \) represents a predefined cut-off distance, and \( d_{ij} \) denotes the distance between point \( i \) and point \( j \) in a space. In this algorithm, if \( d = d_{ij} - d_c < 0 \), \( \mathbb{N}(d) = 1 \); otherwise, \( \mathbb{N}(d) = 0 \). The physical meaning of local density can be explained by Figure 4-1. In the figure, the local density of point \( i \) represents the number of points enclosed by the red circle with a radius of \( d_c \).
Figure 4-1. Local density of a data point (Cha and Wang, 2018).

In the case studies, the Gaussian kernel function of radius (Benoudjit et al., 2002) was applied to calculate the local density in Step 1 to enhance the performance of damage detection with the developed DPFC method. The Gaussian kernel function of radius has been utilized in numerous problems of structural damage detection (Zang et al., 2007; Oh et al., 2009), and its mathematical expression for calculating local density is as follows:

\[
\rho_i = \sum_j \exp \left( -\left( d_{ij} - d_c \right)^2 / 2\sigma^2 \right) 
\]  \hspace{1cm} (4-2)

where \(2\sigma^2\) represents the Gaussian kernel width. It was set to 1 in the case studies.

Step 2. A distance \(\delta_l\) for the point \(i\) is calculated in this step. Here, \(\delta_l\) is defined as the shortest distance from point \(i\) to one of the other points with a higher local density than point \(i\). In Figure 4-2, all data points are ranked in order of decreasing local densities, and \(\delta_l\) can be expressed as follows:

\[
\delta_l = \min_{j; \rho_j > \rho_i} \left( d_{ij} \right) 
\]  \hspace{1cm} (4-3)

In the algorithm, \(\delta_l = \max_j (d_{ij})\) is taken for the point having the highest local density.
Step 3. Select the data points with relatively high values of $\delta_i$ and $\rho_i$ as the density peak points (central points of the following formed clusters). The selection manner is shown in Figure 4-3. The purpose of the selection of density peak point in this step is to meet the requirements of the two previously introduced assumptions, and its performance in clustering is presented in Figure 4-4.
Step 4. Assign the remaining points to the same clusters to which their nearest neighboring points with a higher local density are assigned. The distribution of the final formed clusters can be shown in Figure 4-5.

Figure 4-5. Distribution of the final formed clusters (Cha and Wang, 2018).
Step 5. In a formed cluster in Step 4, the points having an extremely low local density should be far from the core region of a cluster filled with many points. To detect these points as outliers, border regions among the formed clusters have to be determined; the border regions are indicated by the shadowed areas in Figure 4-6. In this algorithm, the border region of a cluster is determined by several points within this cluster that are within the predefined $d_c$ to the points in other clusters, as shown in Figure 4-6. In this figure, the black double-headed arrows indicate the mutual distances of each pair of the points in different clusters. The pairs of points in different clusters are assigned in the border regions when their mutual distances are shorter than $d_c$. For instance, the mutual distance between point A and point B is smaller than $d_c$ in Figure 4-6, so these two points are considered outliers. Besides, $d_{AC}$ is shorter than $d_c$, so point C can be detected as an outlier.

Figure 4-6. Border region between two clusters (Cha and Wang, 2018).

To enhance the performance of novelty detection, the point with highest local density in a border region should be found, as the blue arrows indicate in Figure 4-6. Their local densities, $\rho_b$, can be set to the threshold values of their corresponding clusters and the
points can be detected as outliers if their local densities are lower than $\rho_b$, shown by the black points in Figure 4-7. Outliers are called halo points in the study of Rodriguez and Laio (2014). So far, the clustering steps of their proposed fast clustering algorithm have been explained in detail.

![Figure 4-7](image.png)

**Figure 4-7.** Detected novelty points in the formed clusters (Cha and Wang, 2018).

It can be seen that an integration of density-based and distance-based techniques was used in this algorithm for fast clustering. The DPFC algorithm has the ability to classify the data points into a suitable number of clusters following non-spherical distribution. This algorithm can also detect outliers in the formed clusters, which is helpful in developing it into an unsupervised novelty detection method for detecting and locating damage to a small-scale steel frame in the case studies. In my study, two new steps, Step 6 and Step 7, were developed for adding training and testing processes for unsupervised novelty detection. The details of the new added steps are as follows:

**Step 6.** The novelty points cannot be detected when the predefined value of $d_c$ is too small.

Thus, the goal in this step is to add a training process for obtaining an appropriate $d_c$ to detect novelty points. In the training process, a weight $w$ is used to modify the value of $d_c$, and new tuning cut-off distance is represented by $wd_c$. 

55
Step 7. The future testing points can be separately assigned to a trained statistical model in Step 6 in order to detect the novelty points in them. During a testing process for each testing point, its local density, $\rho_t$, is calculated. When the trained points in the model with higher local density than $\rho_t$ are found, the point nearest to the testing point is selected from these found trained points. Then, the testing point is assigned to the cluster to which the selected trained point belongs. Finally, $\rho_t$ and $\rho_b$ of the assigned cluster are compared for novelty detection. The testing point can be considered a novelty point if $\rho_t$ is smaller than $\rho_b$.

In the application of the developed DPFC method for structural damage detection, the training feature points for the IS of the monitored structures were used to build a statistical model. In the trained model, the training points were partitioned into a reasonable number of clusters, and $\rho_b$ for each formed cluster was determined. The testing points for various structural scenarios were fed into the trained model for novelty detection. Due to the dissimilarity in the value of the extracted features from different structural scenarios, the testing feature points from the DSs of a structure should have long distances to the core regions of the clusters formed by the training features points from the IS of that structure. Therefore, these testing points were detected as novelty points. Figure 4-8 illustrates the procedure of damage detection with the developed DPFC method. In my study, a group of acceleration responses were measured from several sensors installed on an experimental structure, as presented in Section 4.3. Then, damage-sensitive features were extracted from the recorded acceleration responses, which will be introduced in Section 4.2. In the case studies with the developed DPFC method in Section 4.4, the novelty detection on these extracted features for each sensor was carried out independently. Firstly, the extracted features were divided into training and testing feature datasets. Then, a statistical model was trained and tested by following the procedures of the DPFC algorithm (DPFCA) explained above. Finally, the damage position in the experimental structure could be roughly localized by its closest sensor with highest rate of novelty detection.
4.2 Damage-sensitive features

Ideal features for structural damage detection have to be sensitive to the presence of damage in the structure; meanwhile, they should be insensitive to operational and environmental variability in the normal range. In the case studies of this chapter, acceleration signals obtained from the dynamic responses of structural joints are used to extract damage-sensitive features. To fairly compare features that may be extracted under different excitation conditions, the normalization of acceleration signals was carried out (Nair et al., 2006). The normalized acceleration signals are obtained as follows:

\[ \rho_i = \sum_j \exp \left( -(d_{ij} - d_e)^2 \right) \]

\[ \delta_i = \min_{j, \rho_j > \rho_i}(d_{ij}) \]
\[ a(t) = \frac{\bar{a}(t) - \mu}{\sigma} \]  \hspace{1cm} (4-4)

where \( \bar{a}(t) \) are the raw acceleration signals before normalization, and \( \mu \) and \( \sigma \) stand for the mean and standard deviation of \( \bar{a}(t) \), respectively. The extraction of damage-sensitive features from the normalized acceleration signals is explained in the following subsections.

### 4.2.1 Crest factor

The first feature is the crest factor of the discrete acceleration signal in time series \( a(t_n) \) from the sensors installed on the experimental structure, which consist of \( n \) discrete acceleration data. This feature has been verified as an index that is sensitive to changes in the structural responses before and after damage (An et al., 2013), and it can be obtained using an equation as follows:

\[
\text{Crest Factor} = \frac{|a|_{\text{peak}}}{a_{\text{rms}}} \hspace{1cm} (4-5)
\]

where \( |a|_{\text{peak}} = \max|a(t_n)| \); \( a_{\text{rms}} = \sqrt{\frac{1}{n} \sum_n (a(t_n))^2} \)

### 4.2.2 T-CWT

A T-CWT is a newly damage-sensitive feature proposed in this study. It is obtained from the \( a(t) \) after continuous wavelet transformation (CWT). The CWT can be realized by the following mathematical expression:

\[
\text{CWT}(s, d) = \int_{-\infty}^{\infty} a(t) \frac{1}{\sqrt{s}} \varphi(\frac{t-d}{s}) \, dt \hspace{1cm} (4-6)
\]

where \( \text{CWT}(s, d) \) represents the two-dimensional matrix of wavelet coefficients, \( a(t) \) is the acceleration signals in time series, and \( \varphi(t) \) stands for the mother wavelet, which can
be dilated by scale parameters $s$ and translated by shift parameters $d$ to generate a group of daughter wavelets. The Daubechies5 (db5) wavelet is used as $\varphi(t)$ for CWT in the feature extraction in the case studies.

Before extracting the feature of T-CWT, it is essential to check the patterns of wavelet coefficients from the acceleration signals of a structural joint before and after damage (Sciegaj et al., 2018), as shown in Figures 4-9 and 4-10. It can be found that as the scalar parameters $s$ increase, the peak of the transformed wavelet indicated by amplitude decreases more obviously when the monitored structural location is damaged. It is explained by the fact that the damage in the pattern of stiffness reduction results in the losses of high-frequency components of vibrations (Noh et al., 2011). To obtain a good damage-sensitive feature, a scale band that is sensitive to the status of damage is defined, as shown in Figures 4-9 and 4-10.

![Figure 4-9. Peaks in the amplitude of the CWT of acceleration responses of a structural joint in its intact scenario (Cha and Wang, 2018).](image-url)
Figure 4-10. Peaks in the amplitude of the CWT of acceleration responses of a structure in its damaged scenario (Cha and Wang, 2018).

Thus, the developed feature T-CWT in this study can be calculated as follows:

\[
T\text{-CWT} = \frac{T_{\text{max}}}{T_{\text{ave}}} \tag{4-7}
\]

where \(T_{\text{max}}\) represents the maximum of the absolute peak values in amplitude in the full-scale range, which was 1–200 for the case studies in Section 4.4. \(T_{\text{ave}}\) represents the average of the absolute peak values in amplitude in the damage-sensitive band. By examining a number of representative figures, as shown in Figures 4-9 and 4-10, the defined damage-sensitive band is 75–200 in the case studies.

4.3 Experiment setup

A three-story two-bay laboratory-scale steel building frame shown in Figure 4-11 was assembled to verify the effectiveness of the developed DPFC method in Section 4.1. The height of each column was 60 cm, and the dimensions of its cross section were 5.08 cm ×
0.64 cm. The dimensions of the beams were the same as those of the columns. The columns and beams were bolted together at each structural joint with four bolts, and the base of the building was bolted to a heavy concrete foundation. Detailed information about the structural joints is presented in Figure 4-12. Eighteen accelerometers were installed on the structure adjacent to the major structural joints to record their acceleration responses. A small shaker was utilized to excite the building at its top corner, which was close to Joint 18.

**Figure 4-11.** Small-scale steel building model (Cha and Wang, 2018).

**Figure 4-12.** Connection details of a structural joint and an installed sensor (Cha and Wang, 2018).
To verify the effectiveness of the developed DPFC method in Section 4.1 for damage detection and localization, two types of damage were created in the prepared steel building model. To represent the IS of the building model, all four bolts at each of the 18 joints were tightened. In the DSs of the building model, the first type of structural damage was induced by loosening the bolts at a structural joint. In addition, a reduced cross-sectional damage column was used to replace an intact column to create another type of structural damage. The building model was tested 60 times in its IS and 10 times in each DS. The sampling frequency was set to 6,000 Hz, and the sampling time lasted 3 seconds in each vibration test. Table 4-1 summarizes the detailed information on the structural damage and test numbers.

**Table 4-1.** Structural scenarios and test numbers (Cha and Wang, 2018).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Number of tests</th>
<th>Damage type and location</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>1–60</td>
<td>No damage</td>
</tr>
<tr>
<td>DS1</td>
<td>61–70</td>
<td>Two bolts loosened at Joint 1 (minor damage)</td>
</tr>
<tr>
<td>DS2</td>
<td>71–80</td>
<td>Four bolts loosened at Joint 1 (major damage)</td>
</tr>
<tr>
<td>DS3</td>
<td>81–90</td>
<td>Four bolts loosened at Joint 10 (major damage)</td>
</tr>
<tr>
<td>DS4</td>
<td>91–100</td>
<td>Reduced cross-sectional column between Joints 2 and 3</td>
</tr>
</tbody>
</table>

### 4.4 Case studies of damage localization

The extracted features of crest factor and T-CWT introduced in Section 4.2 were combined into a group of two-dimensional feature vectors, the principal components of which were damage-sensitive features. Each feature vector corresponded to one structural joint in a single vibration test. In my study, only the data from the IS were used as training data. Because 60 tests for the IS and 10 tests for each DS shown in Table 4-1 were carried out, 60 training points and 10 testing points for each DS could be obtained for each structural joint after mapping these feature vectors on a 2D feature plane. Since the two features on this plane had different measurement scales, the mutual distances among these feature points maybe unable to clearly indicated the differences among the feature points for
different structural scenarios. Normalization is an efficient solution to the problem of different measurement scales for the feature set from each structural joint, and it can be mathematically expressed as follows:

\[
F = \frac{\bar{F} - \min(F)}{\max(F) - \min(F)}
\]

where \(\bar{F}\) represents the feature set before normalization, and \(F\) represents the normalized feature set.

The damage detection at each structural joint of the building frame was carried out independently. The concept of damage localization in my study is that the joint reaching the highest rate of damage detection can be considered the joint closest to the actual damage position. In this way, the damage position in a DS can be roughly localized by the dense sensor network built on the building frame. The results of damage detection with the developed DPFC method and its tuned parameters are presented in Table 4-2.

**Table 4-2.** Calculated rates of damage detection at the structural joints with the DPFC method (Cha and Wang, 2018).

<table>
<thead>
<tr>
<th>Joint</th>
<th>Minimum (\rho)</th>
<th>Minimum (\delta)</th>
<th>(d_c)</th>
<th>(w)</th>
<th>(\eta_p)</th>
<th>Damage Detection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DS 1</td>
</tr>
<tr>
<td>1</td>
<td>2.5</td>
<td>0.2</td>
<td>0.0550</td>
<td>1.9</td>
<td>0.417</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>2.5</td>
<td>0.1</td>
<td>0.0285</td>
<td>2.4</td>
<td>0.300</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>3.0</td>
<td>0.2</td>
<td>0.0343</td>
<td>2.4</td>
<td>0.167</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
<td>0.2</td>
<td>0.0661</td>
<td>1.7</td>
<td>0.233</td>
<td>0.1</td>
</tr>
<tr>
<td>5</td>
<td>2.5</td>
<td>0.1</td>
<td>0.0462</td>
<td>1.5</td>
<td>0.467</td>
<td>0.3</td>
</tr>
<tr>
<td>6</td>
<td>2.5</td>
<td>0.1</td>
<td>0.0183</td>
<td>3.5</td>
<td>0.300</td>
<td>0.7</td>
</tr>
<tr>
<td>7</td>
<td>2.5</td>
<td>0.1</td>
<td>0.0326</td>
<td>2.0</td>
<td>0.217</td>
<td>0.4</td>
</tr>
<tr>
<td>8</td>
<td>1.0</td>
<td>0.1</td>
<td>0.0278</td>
<td>3.8</td>
<td>0.050</td>
<td>0.1</td>
</tr>
<tr>
<td>9</td>
<td>1.0</td>
<td>0.1</td>
<td>0.0056</td>
<td>7.1</td>
<td>0.167</td>
<td>0.5</td>
</tr>
<tr>
<td>10</td>
<td>2.7</td>
<td>0.2</td>
<td>0.0341</td>
<td>2.7</td>
<td>0.400</td>
<td>0.8</td>
</tr>
<tr>
<td>11</td>
<td>2.0</td>
<td>0.1</td>
<td>0.0523</td>
<td>1.6</td>
<td>0.083</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>12</td>
<td>2.0</td>
<td>0.2</td>
<td>0.0390</td>
<td>1.4</td>
<td>0.267</td>
<td>0.6</td>
</tr>
<tr>
<td>13</td>
<td>1.5</td>
<td>0.2</td>
<td>0.0547</td>
<td>1.5</td>
<td>0.167</td>
<td>0.0</td>
</tr>
<tr>
<td>14</td>
<td>3.0</td>
<td>0.1</td>
<td>0.0656</td>
<td>1.4</td>
<td>0.150</td>
<td>0.3</td>
</tr>
<tr>
<td>15</td>
<td>2.5</td>
<td>0.1</td>
<td>0.0236</td>
<td>1.7</td>
<td>0.433</td>
<td>0.6</td>
</tr>
<tr>
<td>16</td>
<td>1.5</td>
<td>0.2</td>
<td>0.0352</td>
<td>5.1</td>
<td>0.067</td>
<td>0.7</td>
</tr>
<tr>
<td>17</td>
<td>2.0</td>
<td>0.2</td>
<td>0.0427</td>
<td>1.5</td>
<td>0.350</td>
<td>0.7</td>
</tr>
<tr>
<td>18</td>
<td>1.0</td>
<td>0.1</td>
<td>0.0062</td>
<td>21.7</td>
<td>0.367</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Specifically, the minimums of $\rho$ and $\delta$ provide the selection district to choose the appropriate density peak points of clusters in the decision graphs, as shown in Table 4-2. In Figure 4-13, the two points shown in cyan and magenta are selected as the local density peak points for the following formed clusters. Figure 4-14 shows the distribution of the two clusters formed by 60 training points from the IS of Joint 3 based on the two selected density peak points in Figure 4-13. For the statistical model of Joint 3, when the weight $w$ in $wd_c$ reaches 2.4 as shown in Table 4-2, 10 halo points (outliers) in the formed clusters are detected. By adding the testing points from the four DSs for Joint 3 to this well-trained statistical model, the DPFC method can identify the testing points from the DSs as novelty points. In Figure 4-15, nine of the 10 testing damage points from DS4 are identified as novelty points, meaning that the rate of damage detection at Joint 3 in DS4 is 0.9, as shown in Figure 4-13.
Figure 4-13. DPFC method: Selection of cluster density peak points for damage detection at Joint 3 (Cha and Wang, 2018).

Figure 4-14. DPFC method: The well-trained statistical model and the detected halo points for Joint 3 (Cha and Wang, 2018).
Based on the calculated damage detection rates listed in Table 4-2, the minor damage in DS1 is localized to two joints (Joints 1 and 10), including the actual damaged Joint 1. In addition, Joint 10 is adjacent to the damaged Joint 1, as shown in Figure 4-11. DS2 is directly localized to the actual damaged Joint 1, with the highest detection rate of 1.0. For DS3, the highest rate is at Joint 10, which is also the damaged position. In DS4, the damaged column between Joints 2 and 3 is also detected with the highest damage detection rate at Joint 3.

Overall, the performance of the developed DPFC method in structural damage localization was very good, with all the damage positions in the four DSs being localized. The developed DPFC method showed a number of advantages in it unsupervised novelty detection. The first advantage is its capacity for judicious selection of suitable cluster density peak points, as well as a reasonable number of clusters. The second advantage is that the developed DPFC method can identify outliers during the training process. This is very helpful when setting a threshold to identify the novelty points in the testing process. One weak point of the developed DPFC method is extensive work required for parameter tuning. Specifically, the parameters of \( \rho \) and \( \delta \) need to be artificially examined for good
clustering performance, and the parameter $w$ has to be manually fine-tuned in the process of novelty detection, as shown in Table 4-2.
Chapter 5: Comparative studies of damage detection and localization

The aims of this chapter are to develop several unsupervised damage detection methods using several existing machine learning and deep learning techniques reviewed in Chapter 2, and to provide extensive comparative studies with these developed methods for damage detection and localization. Section 5.1 proposes five unsupervised damage detection methods, in which several modifications and improvements are made to enhance their performances related to damage detection and localization in extensive comparative studies in Section 5.2. Section 5.3 presents the limitations of these developed methods compared with the DPFC method in Chapter 4 based on their performances in damage detection and localization.
5.1 Unsupervised novelty detection methods

In this section, five existing machine learning and deep learning methods reviewed in Chapter 2 are adopted to transform them into a uniform unsupervised novelty detection mode for extensive comparative studies of damage detection and localization in the following section.

5.1.1 Unsupervised KNN

The KNN method is a widely applied machine learning technique for supervised data classification. Due to its excellent classification performance in empirical applications (Ding et al., 2014; Rogers and Girolami, 2016), it is modified and applied to the features extracted from the acceleration responses measured from various structural scenarios introduced in Sections 4.2 and 4.3 for damage detection and localization. When classifying data (features) with the KNN algorithm, its training dataset can be mapped into a multidimensional space to form a set of training data points. These training points belong to several classes, each labeled with a specific class membership. The classification rule of the KNN algorithm is that a new testing point is classified to the majority class among the \( k \) neighboring training points of the testing point. In the KNN algorithm, \( k \) is a predefined parameter in integers, representing a specified number of the nearest neighboring training points. Therefore, we can see that the KNN method classifies the unknown testing data in supervised learning mode, requiring multi-class training data.

To enable the KNN algorithm to detect novelty points in testing datasets in an unsupervised novelty detection mode, the traditional KNN method has to be modified. The modification of the algorithm is based on a prerequisite that all the training points come from a same class, and they are close to each other because of their similarity in values, forming a point cluster. A new testing point that comes from a different class should have a certain distance from the cluster formed by the training points. For example, the rule of novelty detection with this modified KNN algorithm can be clarified with a simple case under the condition of \( k = 1 \), as shown in Figure 5-1. In this case, there are lots of training feature points and a testing feature point that comes from an unknown class. When these training points belong to a single class, they can form a cluster on this feature plane. In its
novelty detection, the modified KNN algorithm first identifies the nearest training point \( x_1 \) of the testing point \( x_t \), and the distance between the identified training point \( x_1 \) and the testing point \( x_t \) is denoted by \( d_t \). Then, the nearest training point \( x_{11} \) of the point \( x_1 \) is identified, and their mutual distance is denoted by \( d_1 \), as shown in Figure 5-1. Therefore, if \( d_t > d_1 \), the unknown testing point should be far from the cluster; otherwise, the testing point can be assigned to this cluster. Testing points that do not belong to this cluster can be detected by the modified KNN method as novelty points. Under the condition that \( k \) is greater than 1, the number of \( k \) neighboring training points of the objective points can be identified. Then, the introduced \( d_t \) and \( d_1 \) are the averaged distances from the objective points to their neighboring training points.

![Figure 5-1. Novelty detection with the unsupervised KNN method.](image)

In this study, all the training data come from the IS of a monitored structure, and the data from different structural scenarios are taken as the testing data for structural damage detection. For most practical applications in the real world, it is unavoidable that there are some noisy points in the training datasets, which can form outliers of the training point clusters in the feature domain. If there is an outlier that is extremely close to the testing point \( x_t \), the modified KNN algorithm will conduct a false novelty detection under
the condition of parameter $k = 1$. Thus, it is highly necessary to increase the value of $k$ to decrease the noise sensitivity of the modified KNN method in novelty detection (Ding et al., 2014).

By performing cross-validation on the single-class training dataset, the optimal value of predefined parameter $k$ can be found automatically (Hido et al., 2011). To achieve high fitting and novelty detection precision of the modified unsupervised KNN method, the technique of leave-one-out cross-validation (LOOCV) is applied to tune the parameter $k$. In one LOOCV process, there is only one piece of training data taken out of the training dataset as validation data, and the remaining training points are taken as the training data for validating the performance of novelty detection with the modified KNN algorithm. Then, the LOOCV process is repeated with each of the training data selected exactly once as validation data. Finally, the repeated validation results are averaged to produce a final single validation result. By tuning the value of parameter $k$ in the repetition processes of LOOCV, the optimal parameter $k$ can be found when the error of the final single validation result reaches a minimum. Based on the above introduction of LOOCV, it can be found that finding the optimal value of $k$ requires quite a large computation time.

In the case studies, there are only 60 training data points in low-dimensional feature space; the LOOCV is still feasible for them. When it comes to cases with extremely large volumes of training datasets, the process of tuning parameter $k$ with LOOCV will be time-consuming and costly. In these cases, K-fold cross-validation may be more appropriate because it can greatly reduce the training time compared with LOOCV. In the process of K-fold cross-validation, the single-class training dataset is partitioned into the number of K data subsets. In each repetition for validation, one of the K data subsets is selected as a validation dataset, and the remaining K-1 data subsets are combined as a training dataset. Thus, it is clear that the repetition of K-fold cross-validation will be K times, which is far less than the number of repetitions of performing LOOCV.

5.1.2 FCM method

The FCM method is a well-known unsupervised learning method for data clustering. In the clustering process with the FCM method, the training data points are fuzzily partitioned into several clusters, which allows each data point to belong to several clusters with varying
degrees of membership (Bezdek, 2013). The fuzzy partitioning problem can be expressed by minimizing an objective function, which is given by (Havens et al., 2012)

\[
G_s = \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij}^s \|x_i - c_j\|^2,
\]

subject to the following constraints: \(p_{ij} \in [0, 1]\), \(\sum_{j=1}^{M} p_{ij} = 1\), and \(0 < \sum_{i=1}^{N} p_{ij} < N\). Here, \(N\) represents the number of training data points; \(M\) represents the number of clusters; \(x_i\) stands for the \(i\)th training data point; \(c_j\) stands for the centroid of the \(j\)th cluster; \(p_{ij}\) is the fuzzy partition matrix, which contains the membership degree of the training point \(x_i\) belonging to the \(j\)th cluster; \(s\) stands for the exponent for fuzzy partition matrix \(p_{ij}\), which controls the extent of fuzzy overlap between clusters and is typically greater than 1 (\(s\) is set to 2 in the case studies); and \(\|\|\) is the norm expressing the similarity between \(x_i\) and \(c_j\).

The clustering of data points with the FCM method is carried out by the iterative optimization of the objective function \(G_s\) shown in Equation (5-1), updating \(p_{ij}\) and \(c_j\):

\[
p_{ij} = \frac{1}{\sum_{k=1}^{M} \left(\frac{\|x_i - c_j\|^{2s}}{\|x_i - c_k\|^{2s}}\right)}
\]
\[
c_j = \frac{\sum_{i=1}^{N} p_{ij}^s x_i}{\sum_{i=1}^{N} p_{ij}^s}
\]

This iterative optimization is terminated when \(|G_s^{(n)} - G_s^{(n-1)}| < \varepsilon\), where \(n\) represents the number of iterations for convergence, and \(\varepsilon\) stands for a specified minimum threshold, which is set to \(1 \times e^{-5}\) in the case studies. The convergence procedure of the objective function \(G_s\) is composed of the following steps:

Step 1: Randomly initialize fuzzy partition matrix \(p_{ij}\).

Step 2: Calculate the cluster centroids \(c_j\) with initialized \(p_{ij}\) by Equation (5-3).

Step 3: Update \(p_{ij}\) by Equation (5-2).

Step 4: If \(|G_s^{(n)} - G_s^{(n-1)}| < \varepsilon\), the convergence of \(G_s\) is stop; otherwise, return to Step 2.
In the clustering of data points with the FCM method, the data points are classified into the clusters with the highest membership degree according to the updated $p_{ij}$ in Step 3. To enable the FCM method to detect novelty points, a decision boundary can be obtained according to the clustering results (Masud et al., 2010). In the training process, several training data points from a same class are classified into several clusters with the FCM method, as presented in Figure 5-2. Since the FCM method is a Euclidean distance-based clustering technique, the formed clusters follow circular distributions on a 2D plane. Thus, a circle is assigned to each of the clusters, which can enclose most of the data points in that cluster. The clusters’ centroids are taken as the circle centers, as presented in Figure 5-2. In the case studies, the averaged distances from the training points to their respective cluster centroids are used as the radius of the circles, denoted by $r_K$. The training points sitting out of these circles can be considered outliers. In the testing process, testing points are fed into the trained statistical model, and the testing points sitting outside these circles can be detected as novelty points.

![Figure 5-2. Novelty detection with the FCM method.](image)
5.1.3 GMMs

The GMM technique is frequently used as a clustering algorithm in pattern classification. In clustering analysis with the GMM method, a GMM can be seen as a superposition of several components (data point cluster in the context of this study) in Gaussian densities’ distribution (Reynolds, 2009). When a group of multidimensional vectors is mapped into a space as data points, a GMM can be mathematically expressed as follows:

$$p(x | w_k, \mu_k, \Sigma_k) = \sum_{k=1}^{K} w_k g(x | \mu_k, \Sigma_k)$$ (5-4)

where $x$ represents the multidimensional vectors (data); $p$ is the component posterior probability seeking for maximization, where the data are assigned to the component yielding the highest posterior probability; $K$ represents the number of components for data classification; and $w_k$ is the weight of the corresponding $k$th component, satisfying the constraint $\sum_{k=1}^{K} w_k = 1$. In addition, $\mu_k$ and $\Sigma_k$ stand for the mean vector and covariance matrix of the component $k$, respectively, and $g(x | \mu, \Sigma)$ stands for the Gaussian probability density function of each component, which is mathematically formulated as follows:

$$g(x; \mu, \Sigma) = \frac{1}{2\pi^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2} (x - \mu) \Sigma^{-1} (x - \mu)\right\}$$ (5-5)

where $D$ represents the dimension of the multidimensional vector, $\mu$ is the mean of the multidimensional vector dataset $x$, and $\Sigma$ is the covariance matrix of the dataset $x$.

The components of a GMM have their specific parameters $\mu, \Sigma, \text{and} \ w$. These three parameters of a GMM need to be estimated to enable the model to fit the distribution of data points. The technique of expectation maximization (EM) is typically applied for the parametric estimation of the GMM method (Słoński, 2017). The parametric estimation with the EM algorithm comprises the following procedures:

Step 1: Randomly initialize the parameters $w_k, \mu_k, \text{and} \Sigma_k$ for the component $k$. Calculate the posterior probability $p$ for each data point by the following equation:

$$p(x_i | w_k, \mu_k, \Sigma_k) = \frac{w_k g(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^{K} w_k g(x_i | \mu_k, \Sigma_k)}$$ (5-6)
where $x_i$ is the $i$th data point.

Step 2: Estimate new $\tilde{w}_k$, $\tilde{\mu}_k$, and $\tilde{\Sigma}_k$ with the following equations:

$$
\tilde{w}_k = \frac{1}{N} \sum_{i=1}^{N} p(x_i | w_k, \mu_k, \Sigma_k) \tag{5-7}
$$

$$
\tilde{\mu}_k = \frac{\sum_{i=1}^{N} x_i p(x_i | w_k, \mu_k, \Sigma_k)}{\sum_{i=1}^{N} p(x_i | w_k, \mu_k, \Sigma_k)} \tag{5-8}
$$

$$
\tilde{\Sigma}_k = \frac{\sum_{i=1}^{N} (x_i - \tilde{\mu}_k)^T (x_i - \tilde{\mu}_k) p(x_i | w_k, \mu_k, \Sigma_k)}{\sum_{i=1}^{N} p(x_i | w_k, \mu_k, \Sigma_k)} \tag{5-9}
$$

where $N$ represents the number of data in dataset $x$.

Step 3: Evaluate a log-likelihood function, given by

$$
\log p(x | w_k, \mu_k, \Sigma_k) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} w_k g(x_i | \mu_k, \Sigma_k) \tag{5-10}
$$

Step 4: Return to Step 2 for the convergence of the log-likelihood functions in Equation (5-10). The iteration process is continued until a predefined convergence threshold is reached.

The purpose of the EM algorithm is to estimate $w_i$, $\mu_i$, and $\Sigma_i$ based on their randomly initialized values. The iteration of the parametric estimation process aims to increase the likelihood value of the function in Equation (5-10) until convergence. Finally, the maximum of posterior probability $p$ in the Equation (5-4) is obtained, and data are classified into the component yielding the highest posterior probability.

A set of training data points can be classified into several clusters with! a fitted GMM, as shown in Figure 5-3. Because each cluster (component of GMM) follows the Gaussian density distribution in Equation (5-5), the formed clusters often follow the elliptical distribution. Thus, it is reasonable to apply the Mahalanobis distance metric to measure the distances from each data point to their cluster centroids as it considers the covariance matrix $\Sigma$ in the data points when measuring the mutual distances among the data points in a cluster. The Mahalanobis distance can be mathematically expressed as follows:
where $\beta_i$ represents the Mahalanobis distance between the data point $x_j$ and the cluster centroid $\mu_k$ of cluster $k$ and $\Sigma_k$ is the covariance matrix of data point set $x$ in cluster $k$. Compared with the Euclidean distance in forming decision boundaries for novelty detection with the FCM method, the Mahalanobis distance is more suitable for the elliptical clusters formed by the GMM method. Using the mathematical expression in Equation (5-11), it can be easily demonstrated that surfaces in which $\beta$ is constant are ellipsoidal with the ellipse center in $\mu_k$. In this study, the elliptical decision boundary for novelty detection is determined by averaging the Mahalanobis distances from the training points in each cluster to their cluster centroid. In the novelty detection for the new testing data points, the testing points sitting outside the enclosed elliptical boundaries are considered novelty points, as shown in Figure 5-3.

\[
\beta_i^2 = (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)
\]  

(5-11)

Figure 5-3. Novelty detection with the GMM method using Mahalanobis distance.
5.1.4 OC-SVMs

OC-SVMs represent the ongoing development of traditional two-class (positive and negative) support vector machines (Rogers and Girolami, 2016). Before an overview of OC-SVMs in novelty detection, the basic concept of SVM for binary classification is briefly introduced.

For a given set of training data, \( X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \) in which \( x_i \in \mathbb{R}^d \) and \( y_i \) is the class label of the data \( x_i \), where \( y_i \in [+1, -1] \) and \( i = 1, \ldots, n \). An SVM aims to build a linear classifier in the form of an optimal separating hyperplane to maximize the “margin” between the positive and negative two-class dataset \( X \), as shown in Figure 5-4. This optimal hyperplane can be expressed with the equation \( w^T x + b = 0 \), where \( w \in F \) and \( b \in \mathbb{R} \) are two parameters that determine the position of the hyperplane in the feature space \( F \). The constraints of the data for the two classes are \( y_i (w^T x_i + b) \geq 1 \). The decision function for a new testing data \( x_t \) classification can be formulated as

\[
 f(x | w, b) = \text{sgn} (w^T x_t + b) \tag{5-12}
\]

where \( \text{sgn} (w^T x + b) = 1 \) when \( w^T x + b \geq 0 \); otherwise, \( \text{sgn} (w^T x + b) = -1 \).

Figure 5-4. Binary classification with an SVM.
The objective of SVMs in binary classification is to find the maximum margin between the two-class data points, as shown in Figure 5-4. The margin is determined by the parameters \( w \) and \( b \), which control the orientation and displacement of the linear decision boundary, respectively. Without detailed geometric derivation, the maximization of the margin can be converted to find the minimum value of \( \|w\| \). The introduced SVM was only applicable for linearly separable classification tasks; however, there are many nonlinear classification problems in the real world. For the data points that cannot be linearly separable in their original space, it is feasible that they can be linearly separated by a hyperplane when they are projected into a new space. Thus, a nonlinear function \( \varphi(x) \) is typically used to transform the original dataset \( x \) into a high-dimensional space to obtain a nonlinear decision boundary when the determined hyperplane is transformed back into the original space. Meanwhile, many practical cases allow few data points to sit in the margin region to prevent the over-fitting issue because of the existence of noisy data points. Under such conditions, a slack variable \( \xi \) is introduced to slacken the constraints of the two-class training data, and a parameter \( C > 0 \) is used to tune the trade-off between the classification error on the training dataset and margin maximization. Therefore, the optimization task can be formulated by an objective function, expressed by

\[
\min_{w, \xi, C} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i
\]

subject to 
\( y_i (w^T \varphi(x_i) + b) \geq 1 - \xi_i; \quad \xi_i \geq 0, \)

where \( x_i \) is the \( i \)th data in the training dataset \( x \), \( y_i \) is the class label of \( x_i \), where \( y_i \in [1, -1] \), and \( \xi_i \) represents the slacken parameters associated with \( x_i \).

To solve the minimization problem shown in Equation (5-13), its constraints are incorporated into the objective function through a set of Lagrange multipliers \( \alpha_i \), which are greater than or equal to 0. The converted decision function for new testing data \( x_t \) classification can be written as

\[
f(x) = \text{sgn} \left( \sum_{i=1}^{N} \alpha_i y_i K(x_t, x_i) + b \right) \quad (5-14)
\]

subject to \( 0 \leq \alpha_i \leq C \).
where $\alpha_i$ is the Lagrange coefficient associated with $x_i$ and $x_i$ is called the support vector when $\alpha_i > 0$. In the decision function in Equation (5-14), only the support vectors are weighted in the decision function $f(x)$; $K(x_i, x_i)$ is a kernel function, expressed by $\varphi(x_i)^T \varphi(x_i)$. It can be found that the outcomes of the decision function only rely on the dot product of the data vectors in the transformed space. Thus, it is not necessary to perform an explicit transformation, but directly only uses a function $K$ instead, which can provide the same results. The popular choices for the kernel function used in data classification with SVM are linear, polynomial, and radial basis function (RBF) kernels (Rogers and Girolami, 2016). The RBF kernel is a widely applied kernel function in data classification with SVMs, and it is able to map low-dimensional data into infinite high-dimensional space. According to Mercer’s theorem, the kernel function $K(x_j, x_i)$ is used to replace the inner dot product $\epsilon(x_i) \cdot \epsilon(x_j)$. This replacement is called the kernel trick, and it leads to efficient computation to obtain nonlinear decision boundaries.

An OC-SVM is the extension of an SVM to enable novelty detection in the absence of training data points in the negative class. The novelty detection classifier built by an OC-SVM can be represented by a kind of nonlinear hyperplane in a space, which can enclose most training data points only coming from a positive class, with only a few noisy points (outliers) sitting outside, as shown in Figure 5-5.

![Figure 5-5. Novelty detection using an OC-SVM.](image-url)
The training work of an OC-SVM can be considered a quadratic programming problem, which aims to minimize the value of \(\|w\|\). The objective function of OC-SVMs can be formulated as follows (Wang and Cha, 2017):

\[
\begin{align*}
\min_{w} & \quad \frac{1}{2} \|w\|^2 + \frac{1}{vN} \sum_{i=1}^{N} \xi_i - b \\
\text{subject to} & \quad w^T \varphi(x_i) - b \geq -\xi_i, \quad \xi_i \geq 0
\end{align*}
\]  

(5-15)

where \(x_i\) is the \(i\)th data point in the training dataset, \(\varphi(x)\) is a nonlinear function for data transformation, \(w\) and \(b\) stand for the terms of normal vector and bias, respectively, \(\xi_i\) is the slack variable for \(x_i\), \(N\) represents the number of training points, and \(v\) stands for a predefined upper limit fraction of the allowable outliers in the training dataset.

Incorporating the constraints with Lagrange multipliers \(\alpha\) into the objective function in Equation (5-15), and this function is transformed into

\[
\begin{align*}
\min_{w, \alpha} & \quad \frac{1}{2} w^T w + \frac{1}{vN} \sum_{i=1}^{N} \xi_i - b - \sum_{i=1}^{N} \alpha_i (w \cdot \varphi(x_i) - b + \xi_i) \\
\text{subject to} & \quad 0 \leq \alpha_i \leq \frac{1}{vN}
\end{align*}
\]  

(5-16)

This represents a partial derivative of the new transformed objective function in Equation (5-16) with respect to \(w, b\), and \(\frac{1}{vN}\). To satisfy the optimum of the objective function, its derivatives must be zero, and there are three identity equations obtained as follows: \(w = \sum_{i=1}^{N} \alpha_i \varphi(x_i), \sum_{i=1}^{N} \alpha_i = 1\), and \(\sum_{i=1}^{N} \xi_i = 0\). Substituting these identities into the objective function in Equation (5-17) gives a new objective function, which can be written as

\[
\begin{align*}
\max_{\alpha} & \quad -\frac{1}{2} \sum_{i=1, j=1}^{N} \alpha_i \alpha_j K(x_j, x_i) \\
\text{subject to} & \quad 0 \leq \alpha_i \leq \frac{1}{vN}, \sum_{i=1}^{N} \alpha_i = 1
\end{align*}
\]  

(5-17)

Thus far, the corresponding dual problem can be formulated as
\[
\min_{\alpha} \frac{1}{2} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j K(x_j, x_i) \tag{5-18}
\]

subject to \(0 \leq \alpha_i \leq \frac{1}{vN}, \sum_{i=1}^{N} \alpha_i = 1\)

where \(\alpha_i\) and \(\alpha_j\) are the Lagrange coefficients of the training data \(x_i\) and \(x_j\), respectively, whereas \(K(x_j, x_i)\) is the kernel function. In the case studies, the Gaussian kernel function \(K(x, y) = \exp(-\gamma(x - y)^2)\) was employed as the kernel function in OC-SVMs. In addition, the LOOCV technique was applied to determine the optimal \(\gamma\) in the Gaussian kernel function.

Testing data points \(x_t\) from a different class can be identified by a sign decision function as follows:

\[
f(x) = \text{sng}(\sum_{i=1}^{N} \alpha_i K(x, x_i) - b) \tag{5-19}
\]

\[
b = \frac{1}{m} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j K(x_j, x_i) \tag{5-20}
\]

subject to \(0 \leq \alpha_i \leq \frac{1}{vN}\)

where \(x_i\) can be selected as a support vector when it is associated with non-zero \(\alpha_i\), while \(m\) represents the number of support vectors in the training data. A testing point can be detected as a novelty point when the outcome of decision function \(f(x)\) is minus. In other words, novelty points in the testing datasets should sit outside the enclosed decision boundary defined by \(\sum_{i=1}^{N} \alpha_i K(x_i, x) - b = 0\), as shown in Figure 5-5.

### 5.1.5 Unsupervised CNNs

CNNs are widely used artificial neural networks for classification in supervised learning mode; they employ a convolution operation instead of fully connecting between their hidden layers (Krizhevsky et al., 2012). Traditionally, the architectures of deep neural networks have been composed of several hidden layers, each made up of a set of neurons. Every neuron in each layer is fully connected to all neurons in the prior layer. CNNs are different from regular deep neural networks in their architectures. Generally, the input
layers in CNNs are organized in three dimensions—width, height, and depth—when they are used to deal with RGB image inputs. In this study, the inputs are the acceleration data in time series, so the sequential acceleration data are shaped into 2D matrixes before they are fed into the input layer of the CNNs, as shown in Figure 5-6. The details of the convolutional and pooling operations on a 2D data matrix are shown in Figure 5-6(a).

![Convolution and pooling operations on a 2D matrix](a)

The hidden layers of regular CNNs typically comprise several convolutional, pooling, and activation layers, followed by a fully connected layer and a softmax layer for classification, as shown in Figure 5-6(b). The final output layer in CNNs will be reduced to a single vector of neurons corresponding to the probability scores calculated by the prior

![CNN architecture](b)

**Figure 5-6.** Convolution and pooling operations on a 2D matrix (a) and a CNN for supervised classification (b).
softmax layer. CNNs generally have two components in their architectures—a feature extraction part and a classification part. In the feature extraction component, the network performs a series of convolution, pooling, and activation operations during which the feature representations are extracted from the inputs. For the classification component, the fully connected layers and softmax layers serve as a classifier for the previously extracted feature representations.

In this study, a novel unsupervised CNN is designed for data reconstruction in an unsupervised learning manner, and its neural network architecture is shown in Figure 5-7 (Wang and Cha, 2018). To achieve data reconstruction, a decoder is used to replace the classification part shown in Figure 5-6. Specifically, the developed unsupervised CNN is composed of two convolutional layers, two pooling layers, and one fully connected layer to reconstruct its inputs.

**Figure 5-7.** Neural network of an unsupervised CNN for data reconstruction (Wang and Cha, 2018).

In the case studies of structural damage detection, the procedures of the designed unsupervised CNN method are as follows (Wang and Cha, 2018):

Step 1: Normalize a group of raw acceleration responses measured from the intact and unknown structural scenarios to reduce the signal variability under different excitation conditions. Then, use the wavelet synchrosqueezed transform technique to reduce the effects of noises in the measured acceleration responses. Finally, use fast Fourier transform to convert them into the frequency domain.

Step 2: Divide the transformed signals in the frequency domain into several frequency ranges, and shape the transformed signals in each frequency range into a 2D
matrix. Then, feed the matrixes from the intact structural scenario into the
designed unsupervised CNN for training. The matrixes from various testing
structural scenarios are used to test the trained unsupervised CNN.

Step 3: Use the mean squared error (MSE) between the original inputs and reconstructed
outputs of the trained unsupervised CNN as damage-sensitive features. Combine
the extracted features from the transformed signals in several divided frequency
ranges of the same frequency domain into a feature vector.

Step 4: Apply the Mahalanobis distance technique to the training and testing feature vectors
(data points in feature domain) for damage detection, as introduced in Section 3.4.

5.2 Comparative studies

To assess the performance of damage detection and localization with the unsupervised
learning methods developed in Section 5.1, all the methods described in Section 5.1 use
the same experimental datasets as the developed DPFC method in Chapter 4.

5.2.1 Unsupervised KNN method

The experimental results of damage detection with the unsupervised KNN method are
shown in Figure 5-8. In its novelty or damage detection at Joint 5 of the building model in
Figure 4-11, 60 training points from the IS of Joint 5 are seen as a cluster. By tuning with
cross-validation, the optimized values of $k$ for the statistical model are obtained. Then, 10
testing points from each DS of Joint 5 are separately fed into this model for novelty
detection. The rate of damage detection is the ratio of the number of identified novelty
points to the total number of testing points. In Figure 5-9, 4 of the 10 testing points from
DS3 are identified as novelty points at Joint 5. Thus, the damage detection rate at Joint 5
is 0.4 in DS3, as indicated in Figure 5-8.
Figure 5-8. Calculated rates of damage detection at eighteen joints with the unsupervised KNN method.

Figure 5-9. Unsupervised KNN method: Performance of damage detection at Joint 5 for DS3.
In the context of this study for structural damage localization, the joint reaching the highest damage detection rate can be considered the joint closest to the actual damage position. From the calculated rate of damage detection at each joint, the damaged column in DS4 is detected and localized with the highest rate of 0.9 at Joint 3, which is one end of the damaged column. Joint 2, the other end of the column, reaches the second highest rate in DS4. The rate of damage detection at Joint 1 reaches one of the highest rates in DS1. The damage in DS2 is localized to Joint 10, which is adjacent to the actual damage location at Joint 1 (see Figure 4-15).

The advantage of this novel unsupervised KNN method is that it can localize the damage scenarios to a certain degree. In addition, the algorithm is stable during the detection process, with reliable results. However, cross-validation did not perform well in the case study because of the limited volume of available training points with their decentralized distribution.

5.2.2 FCM method

The rates of damage detection at all joints are presented in Figure 5-10. All the training points in each model are classified into three clusters using the FCM method. 60 training points from the IS of Joint 3 are classified into three clusters (see Figure 5-11), with five testing points from the DS1 identified as novelty points at Joint 3. According to the results in Figure 5-10, the damaged column in DS4 is directly localized because the two ends of the joints reach the highest rate. Damaged joints in DS1 and DS2 are also localized to a few joints, including the actual damaged joint, Joint 1.
Overall, the performance of damage localization with the FCM method is superior to that with the unsupervised KNN method. The main shortcoming of the FCM method in
clustering is that it needs to predefine the number of clusters; this predefined value can easily cause an unreasonable and unstable classification in the training process when the training data are insufficient. Hence, the results of novelty detection with the FCM method lack stability and reliability.

5.2.3 GMM method

The rates of damage detection with the GMM method are shown in Figure 5-12. Like the FCM method, 60 training points from the IS of Joint 17 are well classified into three clusters using GMM. In Figure 5-13, eight testing points are detected as novelty points by the well-trained GMM at Joint 17 in DS2. The damage in DS1 is localized to Joint 10 with the highest rate of 0.9, which is adjacent to the damage location at Joint 1. The damage in DS2 is localized to few joints, which include the actual damaged Joint 1. The damaged column in DS4 is well localized to Joints 2 and 3, which are the ends of the damaged column. The overall damage localization performance of the GMM method is superior compared with the FCM method.

Figure 5-12. Calculated rates of damage detection at eighteen joints with the GMM method.
The shortcoming of the GMM method is its unstable detection of novelty points. One factor relates to the small volume of the datasets. Thus, to obtain a reliable classification, a large proportion of the dataset should be supplied. The second factor is that the approach requires predefining the number of clusters. The algorithm assigns initial values to the mean multidimensional vectors $\mu_i$ for clusters. As the initial values of $\mu_i$ are diverse, the classification performance may be slightly unstable.

5.2.4 OC-SVM method
In the training process for novelty detection with an OC-SVM, the 60 training points from the IS of each joint are used by an OC-SVM to form an optimal decision boundary. By performing LOOOCV, the optimized value of $\gamma$ can be obtained. The testing points from the DSs sitting out of the enclosed decision boundary can be identified as novelty points. The damage detection results with the OC-SVMs are shown in Figure 5-14. In Figure 5-15, 9 of the 10 testing points from the DS3 of Joint 11 are outside the enclosed boundary. Thus, the rate of damage detection at Joint 11 in DS3 is 0.9. Only the damage column in DS4 can
be well localized with the highest rates at Joint 2; the detection rate at Joint 3 is higher than that of most of the other structural joints.

**Figure 5-14.** Calculated rates of damage detection at eighteen joints with the OC-SVMs.

**Figure 5-15.** OC-SVM method: Performance of damage detection at Joint 11 for DS3.
Although the neighbor of the damaged Joint 10 in DS3 is localized, the other three DSs cannot localize the damage points accurately. This poor performance in damage localization may stem from the vibration’s easy propagation throughout the structure because of the small scale of the structural model. It is expected that using feature vectors in high-dimensional space will result in good performance in damage detection. According to the damage detection results—specifically, that the rates calculated by the OC-SVMs are lower than the rates obtained by the above methods—high-dimensional feature vectors can be used to make the OC-SVM method more sensitive to novelty detection.

5.2.5 Unsupervised CNN method

To assess the effectiveness of the developed unsupervised CNN method for damage detection and localization, the Mahalanobis distance technique is applied to the feature vectors of $MSE$s introduced in Section 5.1.5. The transformed signals in the three frequency ranges of $0–300$ Hz, $300–600$ Hz, and $600–900$ Hz are separately shaped into the 2D matrixes. Because the sampling frequency in the experiments shown in Section 4.3 is 6,000 Hz and the sampling time is 3 seconds, there are 900 signal data in each frequency range. Thus, training and testing matrixes with a size of $30 \times 30$ can be obtained (see Figure 5-7). The training matrixes for each frequency range are separately fed into the unsupervised CNN for training. The size of the training matrixes for the IS is 60, and the size of the testing matrixes for each DS is 10. In the training process, the batch size is set to 10, and the number of training epochs is 100.

The calculated Mahalanobis distances for the training and testing feature vectors for damage detection at Joint 1 for the four DSs are shown in Figure 5-16. To calculate the damage detection rate for each DS, a black line is plotted by setting a threshold value, which is defined by the 95% cut-off over the training points of IS. In the calculation of damage detection rates, the testing points above the line threshold are identified as the novelty points. The obtained rates of damage detection at all joints are shown in Figure 5-17.
Figure 5-16. Damage detection with the Mahalanobis distance technique at Joint 1.

Figure 5-17. Calculated rates of damage detection at eighteen joints with the unsupervised CNN method.

Although Figure 5-17 does not provide useful information about the damage position, the damage detection rates at almost all joints for the four DSs are extremely high,
indicating the high performance of damage detection on the building frame. Because of the effect of excitation at a position extremely close to Joint 18, as indicated in Section 4.3, the damage detection at Joint 18 does not perform well. Compared with the DPFC method in Chapter 4 with all damage positions in the four DSs are localized, as presented in Figure 5-18, the unsupervised CNN method is suitable for damage detection, and it can accurately indicate the global health conditions of the monitored structures.

![Figure 5-18](image)

Figure 5-18. Calculated rates of damage detection at eighteen joints (Table 4-2) with the DPFC method.

### 5.3 Summary

Five unsupervised learning methods (i.e., unsupervised KNN, FCM, GMM, OC-SVM and unsupervised CNN) for damage detection and localization were adopted and developed in this chapter for extensive comparative studies. Vibration can easily propagate through plate-like stiff steel beams and columns due to the small scale of the experimental structure. In this way, the sensors, which are quite densely installed, measure changes in the structural dynamic properties propagated to the entire structure. This can result in inaccurate damage localization in many of the methods developed in this chapter.
Overall, the developed DPFC method performed extremely well in the structural damage localization as described in Chapter 4, proving that it is superior to the five unsupervised learning methods introduced in this chapter. The developed DPFC method has several advantages in novelty detection. First, it can judiciously select the cluster density peak points, as well as a reasonable number of clusters. The second advantage is that the developed DPFC method can identify outliers during the training process. This is helpful when setting a threshold to identify the novelty points in the testing dataset. Finally, the results of damage localization are more accurate than those of the unsupervised damage detection methods investigated in this chapter. Based on the performance of damage detection and localization, the advantages and disadvantages of the six unsupervised novelty detection methods, including the DPFC method, are summarized in Table 5-1, which will be useful for potential users when it comes to choosing suitable methods for structural damage detection and localization in their specific applications.

**Table 5-1.** Summary of the advantages and limitations of the proposed unsupervised learning methods.

<table>
<thead>
<tr>
<th></th>
<th>Unsupervised KNN method</th>
<th>FCM method</th>
<th>GMM method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>Training process is stable and results are repeatable.</td>
<td>Suitable for a large amount of training data.</td>
<td>Mahalanobis distance is suitable for boundary decision.</td>
</tr>
<tr>
<td><strong>Limitations</strong></td>
<td>Local sensitivity in novelty detection; time consuming to train model for the large amount of training data.</td>
<td>Number of clusters needs to be predefined artificially; clustering results are unstable for a small amount of training data.</td>
<td>Number of clusters needs to be predefined artificially.</td>
</tr>
<tr>
<td>OC-SVM method</td>
<td>Unsupervised CNN method</td>
<td>DPFC method</td>
<td></td>
</tr>
<tr>
<td>Advantages</td>
<td>Limitations</td>
<td>Limitations</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Decision boundaries</td>
<td>Damage-sensitive features can be extracted</td>
<td>Reasonable number of clusters; the cluster</td>
<td></td>
</tr>
<tr>
<td>can be determined automatically; high sensitivity of the novelty classifier for high dimensional data.</td>
<td>automatically; fewer tuning parameters are required in the training process.</td>
<td>centers can be detected automatically; ability to identify novelty data (outliers) in the trained normal data.</td>
<td></td>
</tr>
<tr>
<td>Several parameters need to be predefined in its training process.</td>
<td>The extracted features are only effective to indicate the global health condition of structures.</td>
<td>Several parameters need to be predefined and tuned artificially.</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6: Conclusions

A comprehensive strategy of structural damage detection and localization for the SHM discipline was proposed in this thesis. In Chapter 3, a novel unsupervised damage detection method with the technique of deep auto-encoder and OC-SVM was developed; this developed method can accurately indicate the global health conditions of the monitored infrastructures. Once the infrastructures are considered unhealthy or damaged, the developed DPFC method in Chapter 4 can be applied to roughly localize damage positions in large-scale infrastructures. The more sensors are used to monitor the health conditions of infrastructures, the more precise the results of damage localization will be. The effectiveness of the developed unsupervised damage detection methods in this thesis was demonstrated using the raw acceleration signals measured from the accelerometer network installed on the monitored structures. A review of the data-driven damage detection literature was conducted to select suitable methods for extensive comparative studies.
6.1 Achievements

In this thesis, the defined primary objective of developing novel unsupervised damage detection methods in Chapter 1 was successfully achieved. Specifically, the achieved branch goals are as follows:

1. Developing an advanced unsupervised damage detection approach using an advanced deep learning technique to indicate global health conditions of structures (Chapter 3).
2. Developing a novel unsupervised damage localization method using the technique of fast clustering–based novelty detection (Chapter 4).
3. Applying several up-to-date machine learning and deep learning–based unsupervised damage detection methods in Chapters 3 and 5 to conduct comparative studies for both newly developed methods for validation through numerical and experimental tests.

In Chapter 3, a novel unsupervised damage detection method with a deep auto-encoder and an OC-SVM was developed to diagnose the states of a simulated build frame and a laboratory-scale bridge model. The developed method in this study only requires the data measured from the intact structural scenarios as training data. In addition, the method needs few parametric studies on the deep auto-encoder and OC-SVM to enhance its performance in damage detection. Three indexes of errors between the original and reconstructed acceleration responses by the trained deep auto-encoders were selected as features, which can effectively indicate the changes in the state of the monitored structures. In addition, an advanced unsupervised deep learning–based damage detection approach using DRBM was applied to diagnose the health conditions of the steel bridge compared with my developed method; this showed the superiority of my developed method in detecting low damage in the form of bolt loosening in the steel bridge model.

In Chapter 4, a novel density peak–based fast clustering approach was developed to roughly locate damage in a small-scale steel building frame in a laboratory environment. In addition, the developed damage-sensitive features in this study were directly extracted from the measured acceleration responses, which were effective for detecting local structural damage with position information. To verify the effectiveness of the developed method for damage localization, five existing machine learning and deep learning methods (i.e., unsupervised KNN, FCM, GMM, OC-SVM, and unsupervised CNN) were carefully
developed for extensive comparative studies in Chapter 5. For the comparison of damage localization performance, a series of modifications and improvements were established to transform these methods into a uniform unsupervised novelty detection mode. In the comparative studies of damage localization, although the small-scale experimental structure caused difficulties in structural damage localization because the abnormal vibration caused by damage could easily propagate throughout the structure, the DPFC method in Chapter 4 exhibited excellent performance in terms of damage localization compared with the unsupervised damage detection methods developed in Chapter 5.

6.2 Limitations

The most evident limitation of my developed unsupervised deep learning method in Chapter 3 is that the goal of structural damage localization is hard to achieve by this method because its training data lack structural damage information. The proposed DPFC methods in Chapter 4 require heavy work in tuning parameters as part of the training process.

In Chapter 5, cross-validation in the unsupervised KNN method did not perform very well in the case studies because of the small volume of available training feature points with their decentralized distribution. The main shortcoming of the clustering-based FCM method is that it is necessary to predefine several clusters. It can be difficult to predefine a reasonable number of clusters when the distribution of the training data points is unknown. In addition, it is hard to avoid different clustering performances, which lead to unstable novelty detection results when the volume of the datasets is relatively low. The GMM method also has a drawback related to its unstable performance in novelty detection. The limitation of the OC-SVM method in this study is that the dimensions for feature vectors are low, adversely affecting this approach’s performance in novelty detection. The features extracted by the unsupervised CNN method is extremely sensitive to the presence of structural damage, thus, this method is only feasible to monitor global health conditions of structures.
6.3 Future works

Future work on the proposed unsupervised deep learning method with a deep auto-encoder and an OC-SVM in Chapter 3 will involve validating it for large-scale civil infrastructures that are in service and have various types of structural damage. The DPFC method proposed in Chapter 4 will be improved to reduce the number of predefined parameters, which will take less work of parameter tuning. Because the proposed unsupervised damage detection methods in Chapter 5 are inferior to the DPFC method in terms of damage localization, improvement work will be required to enhance their performance in this regard.
References


Masud, M., Gao, J., Khan, L., Han, J. & Thuraisingham, B. M. (2010). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering, 23*(6), 859-874.


SBDTUM (Steel Bridge Design Team at the University of Manitoba) (2017), University of Manitoba, Winnipeg, Manitoba, Canada. http://steelbridge.umes.mb.ca.


