SoleimaniBabakamali_MH_D_2022.pdf

By Hesam Soleimani

WORD COUNT	49804	TIME SUBMITTED	03-OCT-2022 07:15PM
		PAPER ID	91197626

Toward a general novelty detection framework in structural health monitoring; challenges and opportunities in deep learning.

Mohammad Hesam Soleimani-Babakamali

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Civil and Environmental Engineering

Rodrigo Sarlo, Chair

Ismini Lourentzou

Farrokh Jazizadeh Karimi

Kourosh Nasrollahzadeh Neshelli

September 20, 2022

Blacksburg, Virginia

Keywords: Structural health monitoring, Reliability Analysis, Dimensionality reduction, Generative Adversarial Networks, Sensor output validation

Copyright 2022, Mohammad Hesam Soleimani-Babakamali

Toward a general novelty detection framework in structural health monitoring; challenges and opportunities in deep learning.

Mohammad Hesam Soleimani-Babakamali

(ABSTRACT)

Structural health monitoring (SHM) is an anomaly detection process. Data-driven SHM has gained much attention compared to the model-based strategy, specifically with the current state-of-the-art machine learning routines. Model-based methods require structural information, time-consuming model updating, and may fail with noisy data, a persistent condition in real-time SHM problems. However, there are several hindrances in supervised and unsupervised settings in machine learning-based SHM. This study identifies and addresses such hindrances with the versatility of state-of-the-art deep learning strategies. While managing those complications, we aim at proposing a general, structure-independent (*i.e.*, requires no prior information) SHM framework. Developing such techniques plays a crucial role in the SHM of smart cities. In the supervised SHM and sensor output validation (SOV) category, data class imbalance results from the lack of data from nuanced structural states. Employing Long Short-Term Memory (LSTM) units, we developed a general technique that manages both SHM and SOV. The developed architecture accepts high-dimensional features, enabling the train of Generative Adversarial Networks for data generation, addressing the complications of data imbalance. GAN-generated SHM data improved accuracy for low-sampled classes from 44.77% to 64.58% and from 73.39% to 90.84% in two SOV and SHM case studies, respectively. Arguing the unsupervised SHM as a practical category since it identifies novelties (*i.e.*, unseen states), the current application of dimensionality reduction (DR) in unsupervised SHM is investigated. Due to the curse of dimensionality, classical

unsupervised techniques cannot function with high-dimensional features, driving the use of DR techniques. Investigations highlighted the importance of avoiding DR in unsupervised SHM, as data dimensions that DR suppresses may contain damage-sensitive features for novelties. With DR, novelty detection accuracy declined up to 60% in two benchmark SHM datasets. Other obstacles in the unsupervised SHM area are case-dependent features, lack of dynamic-class novelty detection, and the impact of user-defined detection parameters on novelty detection accuracy. We chose the fast Fourier transform-based (FFT) of raw signals with no dimensionality reduction to develop the SHM framework. A deep neural network scheme is developed to perform the pattern recognition of that high-dimensional data. The framework does not require prior information, with GAN models implemented, offering robustness to sensor placement in structures. These characteristics make the framework suitable for developing general unsupervised SHM techniques.

Toward a general novelty detection framework in structural health monitoring; challenges and opportunities in deep learning.

Mohammad Hesam Soleimani-Babakamali

(GENERAL AUDIENCE ABSTRACT)

Detecting abnormal behaviors in structures from the input signals of sensors is called Structural health monitoring (SHM). The vibrational characteristics of signals or direct pattern recognition techniques can be applied to detect anomalies in a data-driven scheme. Machine learning (ML) tools are suitable for data-driven methods; However, challenges exist on both supervised and unsupervised ML-based SHM. Recent improvements in deep learning are employed in this study to address such obstacles after their identification. In supervised learning, the data points for the normal state of structures are abundant, and datasets are usually imbalanced, which is the same issue for the sensor output validation (SOV). SOV must be present before SHM takes place to remove anomalous sensor outputs. First, a unified decision-making system for SHM and SOV problems is proposed, and then data imbalance is alleviated by generating new data objects from low-sampled classes. The proposed unified method is based on the recurrent neural networks, and the generation mechanism is Generative Adversarial Network (GAN). Results indicate improvements in accuracy metrics for data classes in the minority. For the unsupervised SHM, four major issues are identified, including data loss during feature extraction, case-dependency of such extraction schemes. These two issues are solved with the fast Fourier transform (FFT) of signals to be the features with no reduction in their dimensionality. The other obstacles are the lack of discrimination between different novel classes (*i.e.*, only two classes of damage and undamaged) and the effect of the detection parameters, defined by users, on the SHM analysis. The latter two

predicaments are also addressed by online generating new data objects from the incoming signal stream with GAN and tuning the detection system to have the same performance regarding user-defined parameters regarding GAN-generated data. The proposed unsupervised technique is further improved to be insensitive to the sensor placement on structures by employing recurrent neural network layers in the GAN architecture, with the GAN that has overfitted discriminator.

Dedication

To my precious Parents, whom I do not deserve, whom so gracious to accept the remorseful and helpless I, whom so absorbing that propelled my wickedness, and so pervasive that deprived me of imbecility. I sought to offer this degree to your Greatness as an appreciation, yet, I am fibbing myself, as commodities are helpless to thank You. Finding myself powerless, I only have my Love to offer. The very Love you poured into my Glass of emptiness at every moment inside moments. The power of Love or the fear of thirst, Glass dreams of fracture to wait for you on the other side of the Dream.

Acknowledgments

I would like to express my sincerest gratitude to Dr. Rodrigo Sarlo for his support, guidance, and granting of liberty to me for working on such broad and novel ideas. Special thanks also go to all the committee members who helped and guided me throughout my time as a Ph.D. student.

Contents

List of Figures **xiv**

List of Tables **xx**

1 Introduction and research gaps **1**

1.1 Supervised learning and research gaps 2

1.2 Unsupervised learning and research gaps 3

1.3 The need for a general SHM approach and associated challenges 5

1.4 Research contributions 6

1.4.1 Research attribution 8

2 A general framework for supervised structural health monitoring and sensor output validation mitigating data imbalance with generative adversarial networks-generated high-dimensional features **11**

2.1 Introduction 12

2.2 Case Studies 16

2.2.1 Yellow Frame 16

2.2.2 Sutong Bridge 17

2.3 Methodology 19

2.3.1 Sequential high-dimensional features and RNNs 20

2.3.2 General neural network architecture for SHM and SOV classification. 21

2.3.3 Network validation 22

2.3.4 Imbalanced datasets and GAN's network architecture 24

2.3.5 Capped GAN generated data objects 25

2.4 Yellow Frame's results and discussion 26

2.4.1 Yellow Frame classification 26

2.4.2 Network validation and testing results 27

2.4.3 Synthesized low-sampled class and GAN-generated data objects . . . 29

2.4.4 Classification improvement via capped GAN-generated features . . . 31

2.5 Sutong Bridge's results and discussion 32

2.5.1 Network validation and testing results 33

2.5.2 GAN-generated data objects for the low-sampled class, PC4 35

2.5.3 Classification improvement via capped GAN-generated features . . . 36

2.6 Limitations of GAN-generated samples for the studied SHM and SOV datasets 37

2.7 Conclusion 40

3 On effectiveness of dimensionality reduction for unsupervised structural health monitoring anomaly detection **42**

3.1 Introduction 43

3.2 Methodology 44

3.2.1 GAN-ensemble technique 45

3.2.2 Bias-free comparison 47

3.2.3 Dimensionality reduction techniques 48

3.3 Experimental details 48

3.4 GAN architecture and training optimizer 49

3.4.1 Comparison 50

3.4.2 Results and discussion 51

3.5 Conclusion 52

4 A system reliability approach to real-time unsupervised structural health monitoring without prior information **54**

4.1 Introduction 55

4.2 Research Significance 60

4.3 Case studies 63

4.3.1 Yellow Frame 63

4.3.2 Z24 Bridge 65

4.4 Methodology 68

4.4.1 Parameter Selection 70

4.4.2 Training Phase 72

4.4.3 Tuning and detection phases 77

4.5 Yellow Frame results and discussion 84

4.5.1 GAN and 1-CG training, Yellow Frame 85

4.5.2 Novelty detection system tuning for V_L 85

4.5.3 Unsupervised real-time novelty detection results, Yellow Frame . . . 86

4.5.4 Literature comparison; Yellow Frame 92

4.6 Z24 Bridge results and discussion 93

4.6.1 GAN and 1-CG training, Z24 Bridge 94

4.6.2 Novelty detection system's tuning for V_L 94

4.6.3 Unsupervised real-time novelty detection results, Z24 Bridge 95

4.6.4 Results comparison; Z24 99

4.7 Computational complexity analysis 101

4.8 Future work 102

4.9 Conclusion 102

5 Towards a general unsupervised novelty detection framework in structural health monitoring 105

5.1 Introduction 106

5.2 Methodology 110

5.2.1 Unsupervised novelty detection 110

5.2.2 GAN Architecture 112

5.2.3 GAN training and loss patterns 114

5.2.4 Adaptive threshold 116

5.3 Experimental details 117

5.3.1 Datasets 117

5.3.2 Research questions 119

5.3.3 Hyper-parameters & Experimental Setup 124

5.3.4 Supplementary analyses 126

5.4 QUGS Results and Discussion 129

5.4.1 Loss Pattern analysis results 129

5.4.2 Architecture Selection and Recurrent Novelty Detection analyses results 132

5.4.3 Sensor Configuration Analysis 133

5.4.4 Contributions w.r.t. prior works 134

5.5 Yellow Frame results and discussion 136

5.5.1 Loss Pattern, Architecture Selection, and Recurrent Novelty Detection analyses results 136

5.5.2 Sensor Configuration analysis results 136

5.6 Supplementary analysis results 140

5.6.1 Time Complexity analysis results 141

5.6.2 Detection Sensitivity analysis results 142

5.7 Conclusions and Future Work 143

6 Conclusion **145**

Bibliography **149**

Appendix A First Appendix **172**

A.1 Codes 172

A.2 Instructions 172

A.2.1 Data_Networks_Training_Tuning 173

A.2.2 Accuracy_Evaluation 177

List of Figures

2.1 Yellow Frame: (a) structure [4] (b) brace labeling, and sensor locations. Used with author's permission. 17

2.2 Sutong Bridge instrumentation [100]. 18

2.3 Seven classes of sensor output patterns for Sutong Bridge. 20

2.4 The feature extraction methodology, N is the number of data channels. . . . 22

2.5 General network architecture. 23

2.6 Proposed Generator and Discriminator architectures. 25

2.7 A sample F_I and F_II features of a data object, shown for all fifteen data channels. 27

2.8 The mean Confusion matrix, Yellow Frame classification on original dataset. 29

2.9 The DC9 GAN's training loss for (a) the F_I feature (b) the F_II feature. . 31

2.10 GAN-generated data objects' features for DC9 (a) the F_I feature(b) the F_II feature. 32

2.11 The mean confusion matrix, Sutong Bridge classification. 35

2.12 The PC4 GAN's training loss for (a) the F_I feature (b) the F_II feature. . 36

2.13 GAN-generated data objects' features for PC4 (a) the F_I feature(b) the F_II feature. 37

2.14 The mean classification confusion matrices for different scenarios (a) R-1-C (b) R-26-C (c) R-26-NC. 38

2.15 Test resembling data object's F_I generations in the SHM problem, DC9. . . 39

2.16 Test resembling data object's F_II generations in the SOV problem, PC4. . 39

3.1 The modified GAN-ensemble approach; shaded blocks are the modified modules compared to the initial approach proposed in Soleimani-Babakamali et al. [92]. 45

3.2 Updated T. 47

3.3 Updated T. 48

3.4 Sensor and brace planning for Yellow Frame. 49

3.5 LSTM-based discriminator, LSTM (i, h, n), has an input size of i, n stacked layers, and a hidden size of h. 50

3.6 AE-I and AE-II architectures. 50

3.7 Stochastic Analysis: (a) PCA-based DR, (b) Kernel-PCA-based DR. 52

3.8 Stochastic Analysis; autoencoder-based DR. 53

4.1 Yellow Frame (a) structure (in color) (b) brace labeling and sensors locations. 64

4.2 Z24 Bridge [78]. 67

4.3 Unsupervised real-time SHM framework; shaded blocks represent user-defined parameters. 70

4.4 Feature extraction step, N is the number of data channels. 73

4.5 $\mathcal{D}$ architecture, L-ReLU(α) is Leaky RuLU activation with negative slope of α, and Dropout(p) is the dropout activation layer with a probability of p. . . 75

4.6 $\mathcal{G}$ architecture, L-ReLU(α) is Leaky RuLU activation with negative slope of α, and BN is the batch normalization layer. 75

4.7 Novelty detection system. 83

4.8 Analogous system. 83

4.9 Unsupervised real-time SHM framework results for FDC1 (a) GAN' training loss (b) a generated F_I feature. 85

4.10 Load MCHS results of novelty detection system's limit-state functions (a) I (b) II (c) III. 86

4.11 Load MCHS results (without outliers) of novelty detection system's limit-state functions (a) I (b) II (c) III. 86

4.12 Dynamic baseline novelty detection results with V_L equal to ten, Yellow Frame. 88

4.13 Dynamic baseline novelty detection results with V_L equal to forty, Yellow Frame. 89

4.14 Dynamic baseline novelty detection results with V_L equal to twenty, Yellow Frame. 90

4.15 Dynamic baseline novelty detection results with V_L equal to forty, Yellow Frame. 91

4.16 Static baseline novelty detection results with V_L equal to twenty, Yellow Frame. (in color) 92

4.17 ROC curve of static baseline detection; Yellow Frame. 92

4.18 Unsupervised real-time SHM framework results for BDC1 (a) the GAN' training loss (b) a generated F_I feature. 94

4.19 Load MCHS results of novelty detection system's limit-state functions (a) I (b) II (c) III. 95

4.20 Dynamic baseline novelty detection results with V_L equal to five, Z24. . . . 96

4.21 Dynamic baseline novelty detection results with V_L equal to ten, Z24. . . . 97

4.22 Dynamic baseline novelty detection results with V_L equal to twenty-five, Z24. 98

4.23 Static baseline novelty detection results with V_L equal to ten, Z24. 99

4.24 ROC curve of static baseline detection; Z24 Bridge. 99

5.1 Unsupervised SHM framework 110

5.2 Dense $\mathcal{G}$ architecture, L-ReLU(α): Leaky ReLU activation with negative α slope, BN: batch normalization. 113

5.3 Dense $\mathcal{D}$ architecture, L-ReLU(α): Leaky ReLU activation with negative α slope, BN: batch normalization. 113

5.4 CNN-based $\mathcal{D}$ architecture, L-ReLU(α): Leaky ReLU activation with α negative slope, Conv1D(i,o): 1D convolutional layer with i input and o output channels, respectively. 114

5.5 LSTM-based $\mathcal{D}$ architecture, LSTM(i,h,n) with input size of i, hidden size of h, and n stacked layers. 115

5.6 Sharper rise of detection scores for unseen data instances, due to an overfitted $\mathcal{D}$. 118

5.7 QUGS structure 119

5.8 Induced damages in QUGS 119

5.9 Yellow Frame structure 120

5.10 Sensor placements in Yellow Frame 120

5.11 Potentially observable GAN training loss patterns 122

5.12 Loss Pattern Analysis 122

5.13 QUGS sensor selection paradigm 126

5.14 Yellow Frame sensor selection paradigm 126

5.15 Time complexities of detection framework's components 127

5.16 SDC4 GAN's training loss; QUGS 128

5.17 A sample real-time unsupervised novelty detection result for Dense, CNN, and LSTM-based architectures; Adaptive threshold approach 132

5.18 FDC5 GAN's training loss; LSTM architecture, Yellow Frame 137

5.19 Samples of Yellow Frame real-time unsupervised novelty detection results for LSTM-based $\mathcal{D}$; Simple threshold approach 137

A.1 F_I.mat contents. 173

A.2 Google Drive Subfolders made by Data_Networks_Training_Tuning.ipynb. 174

A.3 C_L.mat contents. 174

A.4 Examples of GAN's training-associated results saved in DC directories. . . . 175

A.5 Training loop function. 175

A.6 Pre-defined target epochs. 176

A.7 Stored detection scores and thresholds in the Google Drive directory. 177

A.8 Sample output of Detection Displayer cell. 178

A.9 Permutation detection matrix. 178

A.10 Filling the permutation detection matrix. 179

List of Tables

2.1 The dataset summary for Yellow Frame, including various classes of damage simulated by removing certain bracing components, with labels shown in Fig. 2.1b. 18

2.2 The dataset summary for Sutong Bridge, including abnormal patterns' definitions. 19

2.3 The neural network's training options. 24

2.4 F_1 score statistics, Yellow Frame classification on original dataset. 29

2.5 The mean F_1 score $\pm$std for different merging scenarios of GAN-generated DC9 data objects with the main training set. 32

2.6 F_1 score statistics, Sutong Bridge classification. 35

2.7 The testing accuracy for different merging scenarios of GAN-generated data objects with the main training set of PC4. 37

3.1 Yellow Frame dataset summary on accordance with (Fig. 3.4). 49

4.1 The dataset summary for Yellow Frame, including various damage classes simulated by removing certain bracing components, with labels shown in Fig. 4.1b. 65

4.2 The dataset summary for Z24 Bridge, including various global and local damage scenarios. 67

4.3 Selected channels' labels, Z24 Bridge dataset. 68

4.4 Detection summaries for Yellow Frame with $V_L = 10$, and different β values. 89

4.5 Novelty detection comparison; Yellow Frame 93

4.6 Detection summaries for Z24 Bridge with $V_L = 10$, and different β values. . . 98

4.7 Novelty detection comparison; Z24 Bridge 100

5.1 QUGS dataset description and induced damages 124

5.2 Yellow Frame description, and induced damage 124

5.3 Number of parameters in employed neural networks 128

5.4 Loss Pattern analysis results; QUGS (Adaptive and Simple thresholding) . . 131

5.5 Architecture Selection and Recurrent Novelty Detection analyses results; QUGS 133

5.6 Sensor Configuration analysis results; QUGS (Adaptive and Simple thresholding) 135

5.7 Loss Pattern, Architecture Selection, and Recurrent Novelty Detection analyses results; Yellow Frame 138

5.8 Sensor Configuration analysis results; Yellow Frame (Adaptive and Simple thresholding) 139

5.9 Time Complexity analysis; Yellow Frame 142

5.10 Detection Sensitivity result; Yellow Frame 142

List of Abbreviations

FAR False alarm ratio

TAR True alarm ratio

1-CG One-class joint Gaussian distribution

2-CGMM Two-class Gaussian mixture model

AE Autoencoder

ANN Artificial neural networks

AUC Area under the receiver operating characteristic curves

CNN Convolutional neural network layer

DR Dimensionality reduction

FFT Fast Fourier transform

GAN Generative adversarial network

ICA Independent component analysis

L-ReLU Leaky rectified linear unit

LSTM Long Short-Term Memory units

MAC Modal Assurance Criterion

ML Machine learning

PCA Principal component analysis

PDF Probability density function

PSIL Positive Shear Inspection Load

ReLU Rectified linear unit

RNN Recurrent neural network

ROC Receiver operating characteristic

SGD Standard Gaussian distribution

SHM Structural health monitoring

SOV Sensor output validation

PDF is a function whose value at any given sample is a relative likelihood of that sample being close the that value.

SGD is a Gaussian distribution with the mean of zero and a unit standard deviation.

ML is a type of artificial intelligence that makes decisions without being explicitly programmed.

ANN are computing systems inspired by animal brains.

ICA is a method to break a compound signal to different sources.

MAC is a statistical indicator of mode shapes similarities.

DR is a transformation that represents the data in a space with a lower number of dimensions while preserving the variation in the data.

SHM concerns monitoring a system over time by analyzing its responses to detect possible abnormal states.

SOV identifies anomalous sensor outputs to be filtered out before entering SHM routines.

FFT is an algorithm to compute the discrete Fourier transform of a sequence.

PCA is a linear dimensionality reduction technique.

GAN is a class of machine learning frameworks consisting of two sub-neural networks, generator, and discriminator, trained while competing.

AE is an instance of deep dimensionality reduction techniques.

1-CG is a one-class joint Gaussian distribution.

2-CGMM is a mixture of two Gaussian distributions.

LSTM is an instance of deep neural network layers.

RNN is a class of artificial neural networks designed to interpret the sequential dependencies in data.

CNN is a class of neural network layers.

TAR equals the ratio of detected to the total number of novelties in a dataset.

FAR equals the ratio of false alarms to the total detection instances in a dataset.

ROC shows the accuracy of a binary classifier as its detection threshold varies.

AUC measures a binary classifier performance from 0 to 1, with 1 depicting the best performance and 0.5 denoting a random guess.

PSIL is a load that produces positive shear forces at all floors of a structure with usage in damage detection.

ReLU is an activation function in neural networks.

L-Relu is a rectified Linear unit with a non-zero slope in negative values.

Chapter 1:

Introduction and research gaps

Structural health monitoring is a pattern recognition process [108], utilized for the identification of structural states (*i.e.*, supervised) or detection of abnormal conditions (*i.e.*, unsupervised). Generally, pattern recognition is accomplished via either model-based or data-driven approaches. Data-driven approaches are also called Machine-Learning-based (ML-based) SHM. In model-based methods, the established model can be used in supervised, unsupervised, or semi-supervised settings [18, 96, 119], the same as in the data-driven strategy [90]. Any form of prior labeled information can be regarded as a supervised setting.

In model-based techniques, typically, a finite element model [31, 43], or physics-based representations, such as flexibility matrices [19] are fitted with the available data and continuously updated with the incoming data [45, 104]. With the model available, novelty detection and localization is feasible. The main obstacles associated with model-based approaches are complex optimization algorithms and the lack of a global optimization method for different structures [60]. Other complications include the need for numerous sensors to capture the local modal behavior of the system to be modeled [35]. Modal identification also requires expert supervision, with the results being vulnerable in the presence of noisy data [45, 104]. Furthermore, the structural model (*e.g.*, finite element) can only accurately detect novelties that fit the behavior of the established model, and others are either left undetected or

misclassified [13].

Data-driven approaches are alternatives to model-based methods in SHM. These approaches are model-free, requiring no configurational data from structures, *e.g.*, number of stories or load-resisting system of structures. The difference between supervised and unsupervised settings is the presence of labeled training sets. As discussed, in model-based approaches, damages that cannot manifest themselves in the model are undetectable. The same fact is valid for the supervised approach, in which the only detectable conditions are the classes in the labeled set. On the other hand, unsupervised approaches require no labeled data, superior to supervised and model-based approaches. Furthermore, in data-driven approaches, the employment of representation learning tools, *e.g.*, deep convolutional networks [2], showed promising results with noisy data. Representation learning success is due to damage-sensitive features being automatically selected, the most informative representations regarding the dataset, unlike the hand-crafted features, such as modes. Based on the discussion on model-based and data-driven approaches, data-driven techniques are selected herein. They are explored to identify possible obstacles and rooms for improvement using the capacities of deep learning techniques.

1.1 Supervised learning and research gaps

Supervised learning requires labeled training sets. With the labeled training set, numerous statistical-[34, 85] and artificial neural network-based (ANN) techniques [3, 117] apply. Such models require features to perform the detection. Previous to the introduction of the representation learning paradigm, features were mostly based on hand-crafted features, such as subspaces [109], modal information, or classical linear and non-linear dimensionality reduction techniques, such as PCA, Independent Component Analysis (ICA), or Kernel-

PCA [39, 67]. Autoencoders, as a form of neural network-based dimensionality reduction is heavily exploited in the recent literature [51, 57, 61, 70]. As discussed, except for the reduction in dimensionality that makes the data more manageable, such schemes can remove noise and environmental factors [25]. Recently, more sophisticated representation learning techniques, such as the 1D Convolutional neural network layers [2, 3, 12] showed promising results being employed on the raw time-domain data.

Supervised approaches coupled with DR techniques and representation learning approaches are mature regarding accuracy. The main obstacle in supervised methods is not the accuracy but the need for labeled training sets. Gathering enough data from different structural conditions, specifically if in operation, is costly and impractical in most cases. The presence of data samples from only two states of the undamaged and fully damaged system for performing the damage detection [3] is introduced in the literature, but obtaining fully damaged states is a big challenge in itself. Another solution is deciphering the distribution of low-sampled data classes and generating new data objects, called data augmentation. This idea is practiced in several other areas in the literature [10, 58].

1.2 Unsupervised learning and research gaps

Unlike the supervised approaches, Unsupervised techniques do not require labeled training sets. Data-driven approaches are model-free (*e.g.*, finite element models of structures), and no need for labeled data can make the unsupervised approach the most prominent tool in SHM. Investigating unsupervised SHM literature, four key challenges can be identified. These are the loss of information caused by dimensionality reduction (*loss of information*), case-dependency of feature extraction methods (*case-dependent feature extraction*), lack of a dynamic-class novelty detection (*non-dynamic novelty classes*), and the novelty detection's

sensitivity to user-defined detection parameters (*sensitivity to detection parameters*).

The first challenge is the *loss of information*. Application of DR techniques in unsupervised SHM problems is a common practice to address the curse of dimensionality [98], specifically for distance-metric-based threshold models (*e.g.*, clustering). However, unsupervised SHM aims to detect novelties that have not been witnessed before. That setting differs from supervised SHM with known classes; a DR model is trained on those classes, both intact and damaged, accompanied by clustering or other decision-making tools. DR techniques delete the non-informative, correlated, and redundant portion of their training set upon learning from the full distribution of possible classes. Still, it is currently not clear whether DR performs the same for the incoming data from unseen data populations and thus requires a comparative analysis with and without DR. Such comparative analyses aid researchers in selecting between Raw- and DR-based features to develop their SHM methods.

The second challenge, *case-dependent feature extraction*, refers to the non-scalability of feature extraction methods between different structures. Furthermore, such models need to be trained again from scratch for machine learning-based techniques and any new structure, which is time-consuming. Defining a simple, meaningful, and universal feature that requires no DR can solve this issue. However, the main challenge is not defining such a feature, which can be the FFT of signals, but the tools (*e.g.*, neural network architecture) to manage this feature in an unsupervised novelty detection problem.

The third challenge is *non-dynamic novelty classes*. In an unsupervised setting, enough data is available for the baseline class, *i.e.*, intact system; however, data from recently identified novel classes are scarce. Thus, tuning the detection system for continuing the novelty detection process is impossible unless sufficient data for the tuning is first captured. That is why in the literature, there are plenty of binary-unsupervised SHM studies [86, 94, 95, 107, 110]. In such studies, a novelty detector cannot discriminate between different classes

of novelties, and the result is whether the system is normal or abnormal. There are studies in the literature that tried to have a dynamic-class detection, such as Bouzenad Bouzenad et al. [20], Rharã de Almeida Cardoso de Almeida Cardoso et al. [30], and Entezami Entezami and Shariatmadar [35]. But these methods have other issues due to the lack of information explained in the fourth challenge.

The fourth challenge is the *sensitivity to detection parameters*. User-defined (detection) parameters, such as the number of time-series data object to undergo the detection process in each detection iteration (*i.e.*, detection window length) [30], detection thresholds for having a dynamic number of clusters [20], or the order of autoregressive models [35] can result in different detections with the change in their hyper-parameters. One of the reasons behind such behavior is the lack of data instances to make the detection robust to user-defined detection parameters.

1.3 The need for a general SHM approach and associated challenges

Based on what was discussed, the most suitable ML-based SHM approach is unsupervised, as it requires no labeled data or structural model. Still, transforming an unsupervised SHM technique into a general approach faces several challenges, some of which were discussed beforehand, *i.e.*, loss of information due to DR and case-dependent feature extraction. These challenges make current approaches case-dependent and non-generalizable. In such cases, the novelty detection process cannot go online for new structures unless sufficient information from the baseline condition is captured and feature extraction routines are trained. Hence, a general model is unachievable unless designed to be, first, prior-information-free, and

second, function with raw features. On top of that, sensor planning plays a key role in the performance of damage detection techniques. Numerous optimization studies focus on selecting the best sensor placements to capture the behavior of structures [29, 112, 118]. Those optimization routines suffer from the same issues of hand-crafted features or what can be called a rule-based representation of data (*i.e.*, structure-dependent). For instance, the optimization method that works for a bridge may fail for a frame with several stories [60]. Hence, sensor location sensitivity must also be mitigated to obtain a general SHM framework.

1.4 Research contributions

Based on the research directions mentioned above, this study offers the following contributions to the ML-based SHM.

1. Employing a deep learning scheme to address data imbalance in SHM and SOV datasets. Proposing a general novelty detection approach that works for both SHM and SOV problems is the key concept, as data imbalance is simultaneously addressed for both SHM and SOV.

2. A thorough identification of current challenges in unsupervised SHM and how they impact novelty detection accuracy. We propose a framework that addresses the identified obstacles via deep learning.

3. Establishing a general unsupervised SHM approach. The proposed approach must be prior information-free, with raw features, and at the same time, robust to sensor configuration of structures. The proposed technique can be labeled as the first of its kind.

These contributions are published/submitted in the following journal papers, which form the basis of the following chapters of this dissertation.

- **First Contribution:**

  Mohammad Hesam Soleimani-Babakamali, Roksana Soleimani-Babakamali, and Rodrigo Sarlo. A general framework for supervised structural health monitoring and sensor output validation mitigating data imbalance with generative adversarial networks-generated high-dimensional features. *Structural Health Monitoring*, page 14759217211025488, 2021. doi: https://doi.org/10.1177/14759217211025488. URL https://doi.org/10.1177/14759217211025488

- **Second Contribution:**

  Mohammad Hesam Soleimani-Babakamali, Ismini Lourentzou, and Rodrigo Sarlo. Does the curse of dimensionality apply to unsupervised shm? investigating the trade-off between loss of information and generalizability to unseen structural conditions. In *Proceedings of the Thirteenth International Workshop on Structural Health Monitoring (IWSHM), March 15-17, 2022 (formerly December 7-9. 2021)*, pages 644–651. DEStech Publications Ltd., 15-17 Mar 2022, Stanford, CA, USA

- **Third Contribution:**

  Mohammad Hesam Soleimani-Babakamali, Reza Sepasdar, Kourosh Nasrollahzadeh, and Rodrigo Sarlo. A system reliability approach to real-time unsupervised structural health monitoring without prior information. *Mechanical Systems and Signal Processing*, 171:108913, 2022. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2022.108913. URL https://www.sciencedirect.com/science/article/pii/S0888327022000991

- **Fourth Contribution:**

  Mohammad Hesam Soleimani-Babakamali, Reza Sepasdar, Kourosh Nasrollahzadeh,

Ismini Lourentzou, and Rodrigo Sarlo. Toward a general unsupervised novelty detection framework in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 2022. doi: https://doi.org/10.1111/mice.12812. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12812

1.4.1 Research attribution

- **Mohammad Hesam Soleimani-Babakamali**, Roksana Soleimani-Babakamali, and Rodrigo Sarlo. A general framework for supervised structural health monitoring and sensor output validation mitigating data imbalance with generative adversarial networks-generated high-dimensional features. *Structural Health Monitoring*, page 14759217211025488, 2021. doi: https://doi.org/10.1177/14759217211025488. URL https://doi.org/10.1177/14759217211025488

  Mohammad Hesam Soleimani-Babakamali: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing –original draft, Writing–review & editing.

  Roksana Soleimani-Babakamali: Software, Visualization.

  Rodrigo Sarlo: Project funding, Writing –review & editing, Supervision.

- **Mohammad Hesam Soleimani-Babakamali**, Ismini Lourentzou, and Rodrigo Sarlo. Does the curse of dimensionality apply to unsupervised shm? investigating the trade-off between loss of information and generalizability to unseen structural conditions. In *Proceedings of the Thirteenth International Workshop on Structural Health Monitoring (IWSHM), March 15-17, 2022 (formerly December 7-9. 2021)*, pages 644–651. DEStech Publications Ltd., 15-17 Mar 2022, Stanford, CA, USA

  Mohammad Hesam Soleimani-Babakamali: Conceptualization, Methodology, Soft-

ware, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing –original draft, Writing–review & editing.

Ismini Lourentzou: Conceptualization, Writing–review & editing.

Rodrigo Sarlo: Project funding, Writing –review & editing, Supervision.

- Mohammad Hesam Soleimani-Babakamali, Reza Sepasdar, Kourosh Nasrollahzadeh, and Rodrigo Sarlo. A system reliability approach to real-time unsupervised structural health monitoring without prior information. *Mechanical Systems and Signal Processing*, 171:108913, 2022. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2022.108913. URL https://www.sciencedirect.com/science/article/pii/S0888327022000991

  Mohammad Hesam Soleimani-Babakamali: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing –original draft, Writing–review & editing.

  Reza Sepasdar: Software, Writing– review & editing.

  Kourosh Nasrollahzadeh: Methodology, Writing–review & editing.

  Rodrigo Sarlo: Project funding, Supervision, Methodology, Writing–review & editing.

- **Mohammad Hesam Soleimani-Babakamali**, Reza Sepasdar, Kourosh Nasrollahzadeh, Ismini Lourentzou, and Rodrigo Sarlo. Toward a general unsupervised novelty detection framework in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 2022. doi: https://doi.org/10.1111/mice.12812. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12812

  Mohammad Hesam Soleimani-Babakamali: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing –original draft, Writing–review & editing.

  Reza Sepasdar: Software, Writing– review & editing.

Kourosh Nasrollahzadeh: Methodology, Writing–review & editing.

Ismini Lourentzou: Methodology, Writing–review & editing.

Rodrigo Sarlo: Project funding, Writing –review & editing, Supervision.

Chapter 2:

A general framework for supervised structural health monitoring and sensor output validation mitigating data imbalance with generative adversarial networks-generated high-dimensional features

Mohammad Hesam Soleimani-Babakamali, Roksana Soleimani-Babakamali, and Rodrigo Sarlo. A general framework for supervised structural health monitoring and sensor output validation mitigating data imbalance with generative adversarial networks-generated high-dimensional features. *Structural Health Monitoring*, page 14759217211025488, 2021. doi: https://doi.org/10.1177/14759217211025488. URL https://doi.org/10.1177/14759217211025488

Abstract

This study proposes a novelty-classification framework that applies to structural health monitoring (SHM) and sensor output validation (SOV) problems. The proposed framework has simple high-dimensional features with several advantages. First, the feature extraction method is extensively applicable to instrumented structures. Second, the high-dimensional

features' utilization alleviates one of the main issues of supervised novelty-classifications, namely, imbalanced datasets and low-sampled data classes. The recurrent Neural Networks are employed for the classification of high-dimensional features. Furthermore, Generative Adversarial Networks are trained with low-sampled data classes' high-dimensional features for generating new data objects. The generated data objects are combined with the initial training set for improving classification results. The proposed framework is studied on two SHM and SOV datasets. The SHM dataset has twenty-one data classes, with a total test accuracy of 99.60% compared to another study with 88.13% accuracy. The SOV classification shows improved results with a mean accuracy of 96.5% compared to three other studies with mean accuracy values of 93.5%, 92.97%, and 71.1%. Furthermore, the integration of GAN's generated data objects with low-sampled classes improved those classes' mean F_1 score from 44.77% to 64.58% and from 73.39% to 90.84% on SOV and SHM case studies, respectively. The integration of GAN-generated data objects with the initial low-sampled data classes for accuracy improvement shows more potential in the SHM dataset than the SOV case, which can be due to the signal pattern-based labeling logic of SOV datasets.

2.1 Introduction

As rapid urbanization increases the demands on the construction and maintenance of engineering structures [65], sensor-driven structural monitoring is becoming a popular strategy to preserve resources, protect human lives, and reduce overall maintenance costs [88]. This paper will focus on two standard monitoring tasks, namely, structural health monitoring (SHM) [108], and sensor output validation (SOV) [113]. For this discussion, SHM will be defined as a tool for identifying structural damages, whereas SOV is defined as a tool for faulty sensor signal identification. Both SHM and SOV can be performed model-based or

data-driven with machine learning-based techniques in a supervised, unsupervised, or semi-supervised manner.

Supervised methods are practical tools for damage detection, localization, and severity measurement [21], whereas unsupervised techniques are not suitable for damage localization since no labeled data is available. In addition, supervised methods, unlike the model-based approaches [45, 104] can have good robustness to noisy data [115], a common issue in SHM and SOV. However, gathering data from different damage scenarios is a requirement for supervised techniques. Data samples from normal structural behavior are plentiful, but data from damaged conditions or anomalous sensor outputs are rare. In the presence of data from both normal and damaged status, those classes can have vastly different numbers of data objects, resulting in imbalanced datasets with low-sampled classes of data. The imbalanced datasets may result in over-fitting to the normal data, which is in the majority, and poor generalizability.

Feature extraction is the starting point of both supervised and unsupervised techniques. To avoid issues with the curse of dimensionality [98], many SHM and SOV techniques greatly reduce the sensor output dimensions in the feature extraction step. For example, principal component analysis (PCA) and subspace-based techniques which transform structural responses into few subspaces are commonly used for SHM [17, 109, 114] and for SOV problems [33, 36]. However, linear subspace techniques such as PCA cannot capture the non-linearity in SHM or SOV problems [67]. The non-linear dimensionality reduction can be achieved by utilizing deep-learning networks such as autoencoders [70]. These approaches are case-dependent, and the network should be trained from scratch for each new structure, which is a time-consuming procedure. Furthermore, non-linear feature extraction steps with desirable results for a structure may produce poor results for another structure [2].

By employing neural network architectures that can handle high-dimensional data objects,

classification with simple feature extraction steps becomes possible. Recently in SHM, simple feature extraction steps such as having the signals' FFT in a matrix format or the raw sensor output signals were used with 2-D [70], and 1-D convolutional layers [3]. In the SOV case, raw signals were transformed to images and classified via deep neural networks [16, 100].

The previously mentioned studies in SHM performed the analysis in situations where enough data was available. On the discussed SOV studies [16, 100], imbalanced datasets were present. However, the number of data objects in those studies was high, even for the least populated class. Hence, down-sampling of all other classes to have the same number of data objects equal to the class in the minority is utilized to improve the classification results. However, in general, and specifically for SHM, the low-sampled class population is deficient (*e.g.*, a few samples or data objects); hence, other classes cannot be under-sampled for improving the classification accuracy.

This study aims to define a general feature extraction step and utilize proper neural network architectures for the classification. Furthermore, a general approach for handling imbalanced datasets with low-sampled classes of data for both SHM and SOV is established. These tasks are not trivial since SHM and SOV have different labeling logic for data classes. Labeling in SHM datasets is determined by changes in the monitored system's underlying physics. In contrast, labeling in SOV datasets is based on arbitrary signal patterns (*e.g.*, square wave or dead channel). Hence, SHM and SOV generally have different classification approaches. For example, sensor data reconstruction methods [33, 74] are designed explicitly for SOV and do not apply to SHM. In employing the same deep neural network approaches, the feature extraction step is still different for SHM and SOV. In SOV, the features are mostly taken as images of sensors' outputs [16, 61]. In SHM, however, features are the outputs from specific neural network layers, such as the autoencoders.

The approach in this paper leverages general, high-dimensional features in both the time and

frequency domains. These high-dimensional features require specific neural network layers to handle the pattern recognition task, such as convolutional or recurrent neural network (RNN) layers. In this approach, each feature is assumed to be a meaningful sequence. The Long Short-Term Memory (LTSM) architecture is utilized to carry out the classification. Furthermore, a specific kind of deep generative model, namely, Generative Adversarial Networks (GAN) [37], is employed to aid the classification accuracy of low-sampled classes of data in an unsupervised manner. In GAN, a generator is trained from a bounded probability density function to mimic the training data. In this study, one-dimensional GANs are used on the low-sampled classes. The primary goal is to demonstrate how the combination of LSTMs with high-dimensional features can address different types of SHM and SOV classification tasks with comparable or superior accuracy compared to other recently applied methods, as well as mitigating the issue of low-sampled classes by merging the dataset with GAN's generated data objects. It should be noted that, among deep networks, recurrent neural networks (*e.g.*, LSTM networks) and GANs are scarce in the structural monitoring literature [13]. Thus, this paper's results are also intended to encourage the further employment of these techniques to solve problems in the field.

In the following sections, we first present the summary for two benchmark classification case studies addressed with this approach, one in SHM and another in SOV. Subsequently, the proposed features are explained, and the sequential format of extracted features is discussed. The generalized neural network and GAN architectures are then defined. Finally, the step-by-step procedure for performing the two datasets' classification and introducing GAN-generated data objects to the training set is presented. The results are discussed and compared to some of the previous literature that used the same datasets.

2.2 Case Studies

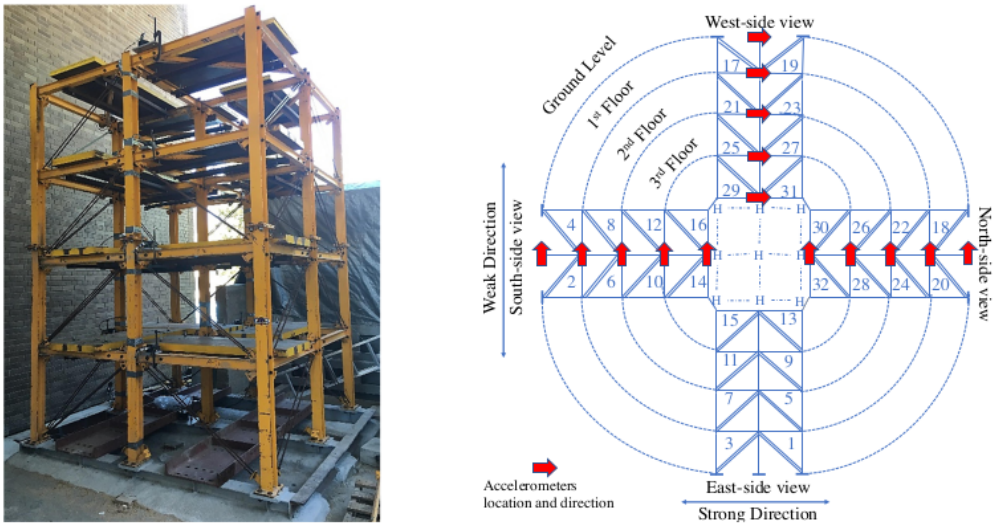
Two experimental case studies of instrumented structures are used to evaluate the proposed methodology: Yellow Frame and Sutong Bridge datasets to evaluate the proposed framework for SHM and SOV problems, respectively. A summary of both case studies' structures, a brief explanation of acquisition systems, and a summary of both datasets are also explained in this section.

2.2.1 Yellow Frame

Yellow Frame [63] is a four-story, one-third scaled modular steel frame established at the University of British Columbia, as shown in (Fig. 2.1a). The frame's elements, such as beams, masses, and braces, can be manipulated to simulate different damage scenarios. Yellow Frame is considered a benchmark for testing the state-of-the-art SHM approaches. Further details on Yellow Frame can be found in Mendler Mendler et al. [63].

The frame response is measured via fifteen accelerometers with a sampling frequency of 1000 Hz over four different days. Further instrumentation details are explained in Allahdadian [4]. A dataset with twenty-one data classes is selected for the proposed framework's evaluation (DC1-DC21). The induced damages descriptions and data summary is shown in Table 2.1 with the frame's modular element labeling depicted in Fig. 2.1b. Studying imbalanced datasets with low-sampled damage classes is of this study's interest to demonstrate the GAN's potential in improving classification accuracy. In this dataset, the difference between the number of data objects in different classes is not high. Hence, in the results section, a low-sampled class is synthesized for the implementation of GANs.

Damage localization and severity measurements were performed on the same damage config-



(a) (b)

Figure 2.1: Yellow Frame: (a) structure [4] (b) brace labeling, and sensor locations. Used with author's permission.

uration of Yellow Frame by Bernagozzi Bernagozzi et al. [18]. Bernagozzi Bernagozzi et al. [18] utilized a modal-flexibility-based approach under the Positive Shear Inspection Load (PSIL) for the classification. This method is exclusively for shear-planar frames, generalized to this 3-D non-planar frame, with the damage localization results compared to the proposed methodology's classification in the results section.

2.2.2 Sutong Bridge

Sutong Bridge is a densely instrumented cable-stayed bridge in China, producing vast amounts of data from various implemented sensors (*e.g.*, accelerometers, thermometers, and strain-gauges). Sutong Bridge dataset used for evaluating the proposed framework is captured in January of 2012 (one month-long) by 38 accelerometers. The dataset contains seven classes of normal and abnormal sensor output patterns. The accelerometers' placement is shown in

Table 2.1: The dataset summary for Yellow Frame, including various classes of damage simulated by removing certain bracing components, with labels shown in Fig. 2.1b.

Damage class	Removed brace ID (No.)	length (sec)	Damage class	Removed brace ID (No.)	length (sec)
DC1	None	668	DC12	DC11 + (17, 19, 25, 27)(I)*	1003
DC2	2,4(II)**	1100	DC13	DC8 + (1, 3, 17, 18)(I)	1053
DC3	DC2 + (18, 20)(II)	1000	DC14	(10, 12)(II)	1047
DC4	DC3+ (1, 3, 17, 19)(II)	830	DC15	DC14 + 21(II), 23(I)	1147
DC5	DC2 + (17, 19)(II)	1002	DC16	(21, 23)(II)	1147
DC6	DC2 + (18, 20)(I)	830	DC17	(7-8, 21, 22)(I)	666
DC7	2(II)	1101	DC18	(5, 6, 7, 8, 21, 24)(I)	559
DC8	(2 , 4)(I)	713	DC19	DC18 + (7, 8, 21, 22)(I)	665
DC9	(25, 27)(I)	1050	DC20	DC19 + (5, 6, 23, 24)(I)	559
DC10	(29, 31, 8, 6)(I)	1170	DC21	(6, 8)(II), (21, 22, 23, 24)(I)	659
DC11	(21, 23 ,29 , 31)(I)	1170			

* One brace is removed
** Both braces are removed

Fig. 2.2 with further information on this dataset presented in Li and Ou [54].

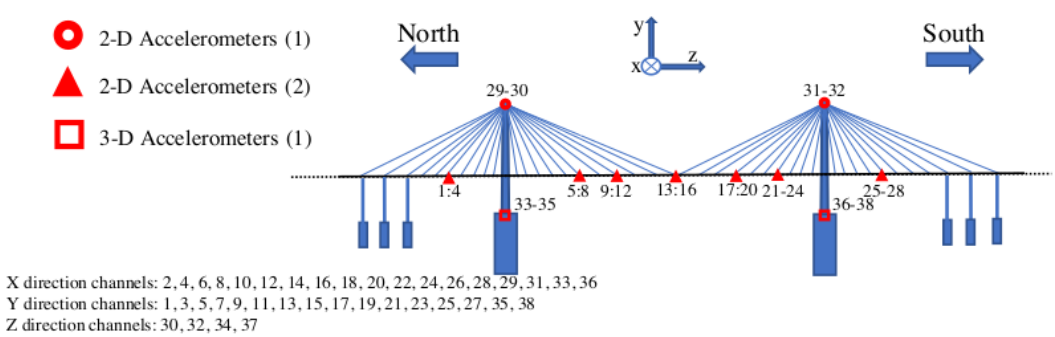


Figure 2.2: Sutong Bridge instrumentation [100].

The sensors outputs were divided into seven categories based on the descriptions presented in Table 2.2. The dataset has one-hour-long acceleration signals as data objects. The dataset

summary and examples of each class data object are reported in Table. 2.2, and Fig.2.3, respectively. According to the dataset summary, this dataset is imbalanced with low-sampled classes of PC4 and PC7. Hence, GAN-generated data objects can be tried to improve the classification accuracy by merging the generated data objects with the training set. There are two studies [16, 100] that worked explicitly on this dataset, however, with more data points (*i.e.*, longer acquisition period). Another study employed the same dataset [11], to perform the classification with the shapelet transformation [56]. The proposed framework's results are compared to those three studies in the results section.

Table 2.2: The dataset summary for Sutong Bridge, including abnormal patterns' definitions.

Pattern class	Description ([100])	Data objects
Normal (PC1)	The pattern is symmetrical; frequency response has peaks	13575
Missing (PC2)	Most/all of the time response is missing	2942
Minor (PC3)	Relative to normal sensor data, the amplitude is very small	1775
Outlier (PC4)	One or more outliers appear in the time response	527
Square (PC5)	The time response is like a square wave	2996
Trend (PC6)	The data has an obvious trend and peak item in both domains	5778
Drift (PC7)	The vibration response is non-stationary, with random drift	679

2.3 Methodology

This section first defines the general high-dimensional features used for both SHM (*i.e.*, damage detection) and SOV (*i.e.*, signal pattern classification). Generalized neural network architecture is then presented for SHM and SOV problems. The high-dimensional feature makes it possible to improve classifier performance by having a GAN trained on those features. Finally, we present a GAN architecture that generates new data objects based on the

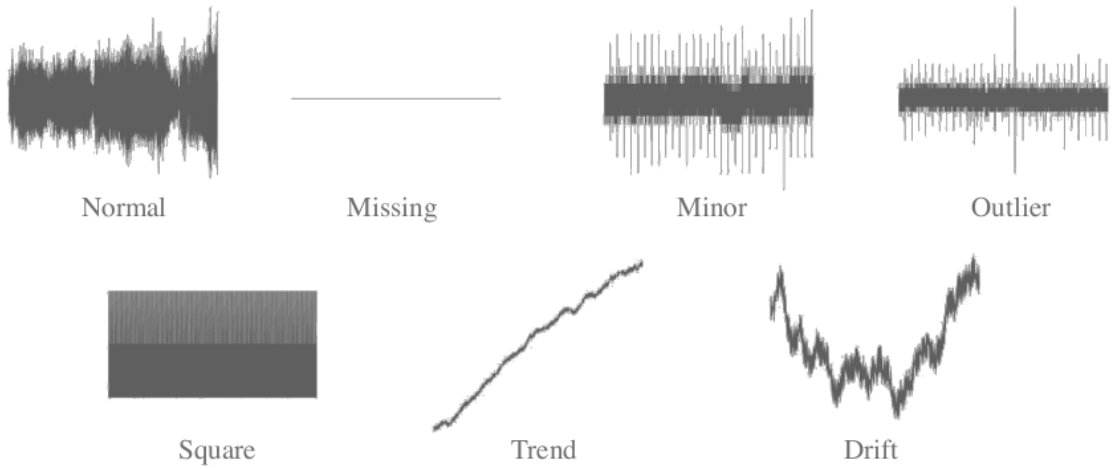


Figure 2.3: Seven classes of sensor output patterns for Sutong Bridge.

low-sampled classes.

2.3.1 Sequential high-dimensional features and RNNs

This study argues the possibility of having the time- and frequency-domain data from the raw input signals as meaningful sequences and utilizing sequence-deciphering neural networks for the classification. Fast Fourier Transform (FFT) and time-domain amplitudes of sensor signals are related amplitudes that follow each other in a particular order. The order can define different states of the system or the presence of a novelty.

Aside from striving to have a general extraction method, the resulting features should also be designed to be compatible with the 1-D GAN. Generally, having features' values to oscillate in a fixed range (*e.g.*, 0-255 for images) is beneficial for training GANs. The feature extraction method shown in Fig. 2.4 was developed based on this logic. Before the feature extraction occurs, it is beneficial to scale the signals to a sensible range, such as 0-10. The captured accelerations are very small; thus, the scaling can improve the classification

process. In the proposed extraction method, N is the number of data channels, T is the number of acceleration data points in each data object. In the proposed extraction method, the sensor output is divided into two features: the time-domain feature (F_I) and the frequency-domain feature (F_II). The F_I feature results from some simple arithmetic, such as taking the absolute value. The F_II feature is the half-spectrum magnitude of the fast Fourier transform of a signal divided by its mean value and bounded between zero and ten. Having a bounded FFT cannot capture the time-series amplitudes, which is crucial for SOV problems and is the reason for having the F_II feature. Finally, both F_I and F_II features are merged into a 3-D matrix, F_III, for the classification.

2.3.2 General neural network architecture for SHM and SOV classification.

The proposed neural network architecture, shown in Fig. 2.5, is utilized for both SOV and SHM. One of the architecture's flexible parameters is N (*i.e.*, number of channels), which specifies the number of branches in the proposed neural network. The number of channels for an SOV problem is one since the novelty detection is performed for each channel separately. On the other hand, the number of channels can be any number for SHM. The input of the network (*i.e.*, a matrix) has $2N$ channels. The first N channels the F_I features, and the rest are the F_II features. F_III has a 3-D matrix format so that each channel's F_I and F_II features become extractable with 2-D convolutional layers in the general architecture. Based on the number of channels, the Attach layer output can be high dimensional; therefore, more than one fully connected layer (FCL) might be required to capture the network's best accuracy in such cases. All data processing and network training are done in MATLAB R2020a.

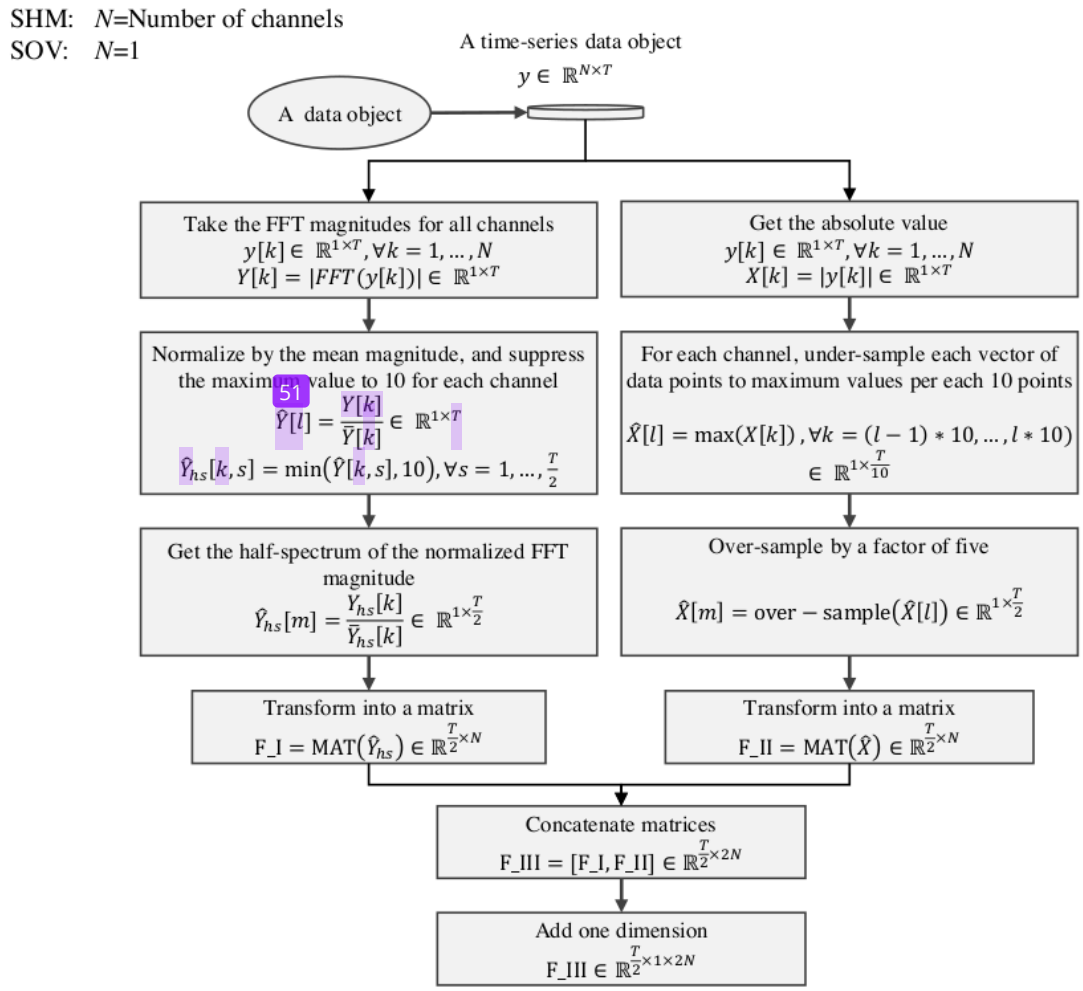


Figure 2.4: The feature extraction methodology, N is the number of data channels.

2.3.3 Network validation

The general architecture presented in Fig. 2.5 has only the number of hidden units of (Bi)LSTM layers as the main tunable element. Hence, for the validation process, six RNN candidates containing three LSTM and three Bi-LSTM units with 100, 200, and 400 hidden units are considered. The validation process is a repeated single holdout procedure. The

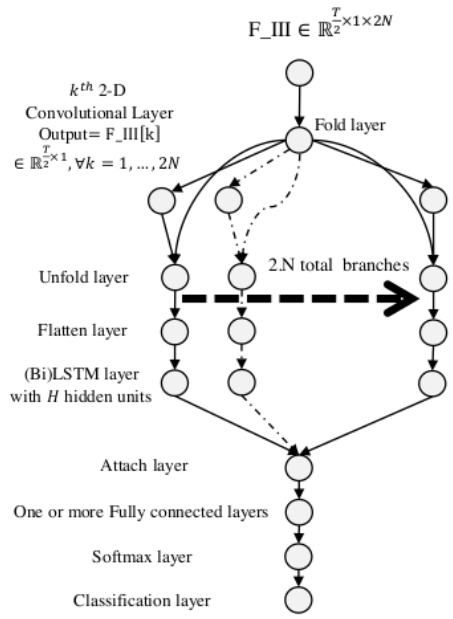


Figure 2.5: General network architecture.

dataset is divided into the training, validation, and testing set with sizes of 10%, 10%, and 80% of the dataset, respectively. Each architecture is trained ten times, and the validation accuracy is the metric for selecting the most suitable (Bi)LSTM layer. Finally, the chosen architecture is trained only on the training set (*i.e.*, 10% of the dataset), allowing for more robust testing of the model on the test set (*i.e.*, 80% of the dataset). The resultant test-to-train ratio of 8 is much higher than the conventional test-to-train ratio of 0.25 (*i.e.*, 80% train-validation and 20% test). This creates a more challenging case for the network and is used to demonstrate that RNNs can capture sequential patterns (*e.g.*, spectral line dependencies in the F_I feature) with few training data objects and extrapolate to a large population of unseen test data. Furthermore, the very large test-to-train ratio rigorously examines the generalizability of the framework's pattern recognition ability for both SHM and SOV problems. For all networks' training phases, the training options in Table 2.3 are

applied unless otherwise mentioned. The F_1 score is utilized as the accuracy metric for the imbalanced datasets.

Table 2.3: The neural network's training options.

Optimizer	Squared gradient decay factor	Initial Learn rate	Mini-batch Size	Learn rate drop period	Learn rate drop factor
ADAM [49]	0.99	3e-4	24	3	0.5000

2.3.4 Imbalanced datasets and GAN's network architecture

The negative impact of imbalanced datasets and low-sampled classes can be alleviated by generating new data objects from the training set with high variability for letting the network better interpret the data. Another possible solution is over-sampling the low-sampled classes by copy-pasting them to have the same number of data objects as the most populated class. Over-sampling is tried on all pieces of training on both case studies in this study. Over-sampling can reduce the over-fitting issue and improve the classification accuracy to some extent; however, the idea of adding the GAN-generated data objects to the main training set before the over-sampling takes place is the core of the proposed methodology for accuracy improvement.

The same GAN architecture is proposed for both SHM and SOV problems, presented in Fig. 2.6. The main layers in both Generator and Discriminator are fully connected layers. The Discriminator and Generator have four layers, in which the last activation for the Discriminator and Generator is the Sigmoid and the rectified linear unit (Relu), respectively. As mentioned before, the F_I and F_II features are exclusively designed to be compatible with the 1-D GAN. One of those feature characteristics is that F_I and F_II features are independent. The F_II feature is scaled between zero and ten. The oscillation characteristics

of the time-series data objects are removed by getting the absolute value of time-series data objects in the F_II feature. Hence, the features are independent. As a result, the GAN's training can be performed separately for F_I and F_II features. The trained GAN's Generator is used to generate new data objects (*i.e.*, F_I and F_II features) from low-sampled classes. Some good ideas are taken from [53] for the GAN implementation in MATLAB.

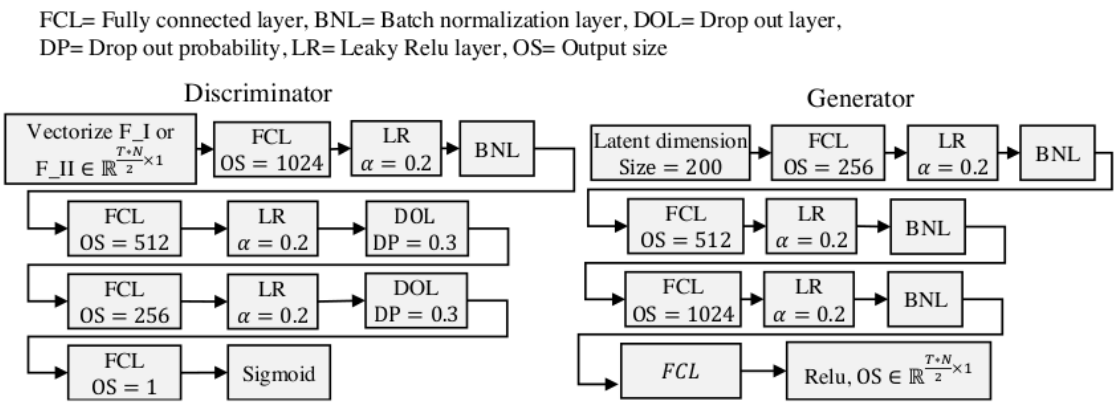


Figure 2.6: Proposed Generator and Discriminator architectures.

2.3.5 Capped GAN generated data objects

Regardless of the GAN's training epochs, the GAN-generated data objects may be unreliable. There might be some part of the latent dimension for which the trained network was not trained adequately to generate realistic data objects. As a result, a discriminative logic must be designed to distinguish between the proper and non-proper GAN-generated data objects. In this study, features are selected in a bounded manner: the F_I feature cannot be greater than ten, and the F_II feature oscillates in a predictable range (*i.e.*, its oscillation range on the training set). The maximum of those features' values in each channel is measurable and can be used to have capped generated data objects. In the proposed capping procedure, generated F_II features with a maximum value greater than 1.2 times the training data's

maximum value are removed. For the F_I features, the ones with a maximum value of less than twelve are further reduced to ten. The capping method can also compensate for a non-well-trained GAN, as there is no rule to conclude a GAN's training. As a result, more stable results can be achieved with the capping method regardless of how well-trained GAN is.

2.4 Yellow Frame's results and discussion

In the following subsections, first, the damage classification is performed on Yellow Frame. Results are then compared to the Bernagozzi *et al.* study. Furthermore, by reducing the number of data objects in one of the classes, a low-sampled class is synthesized. A GAN is trained with those class data objects. The generated data objects are merged with the initial training set for possible accuracy improvement for the low-sampled class.

2.4.1 Yellow Frame classification

A time-series data object duration of one second (*i.e.*, $T = 1000$) is chosen. The reason behind such a short window length is to have a reasonably high number of data objects for the classification. Furthermore, many data objects allow for synthesizing a low-sampled class for the GAN's implementation. Moreover, a high classification accuracy with such a short window can show how versatile are the proposed features and architecture for a dataset with twenty-one classes of data. For the feature extraction (Fig. 2.4), first, the scaling factor is set to be 150 by inspecting a portion of the training data objects. The feature extraction is then carried out on the scaled data objects with an example of the resulting features shown in Fig. 2.7. Each of the features is a matrix with N channels; hence, all data channels are

offered as a single image in Fig. 2.7.

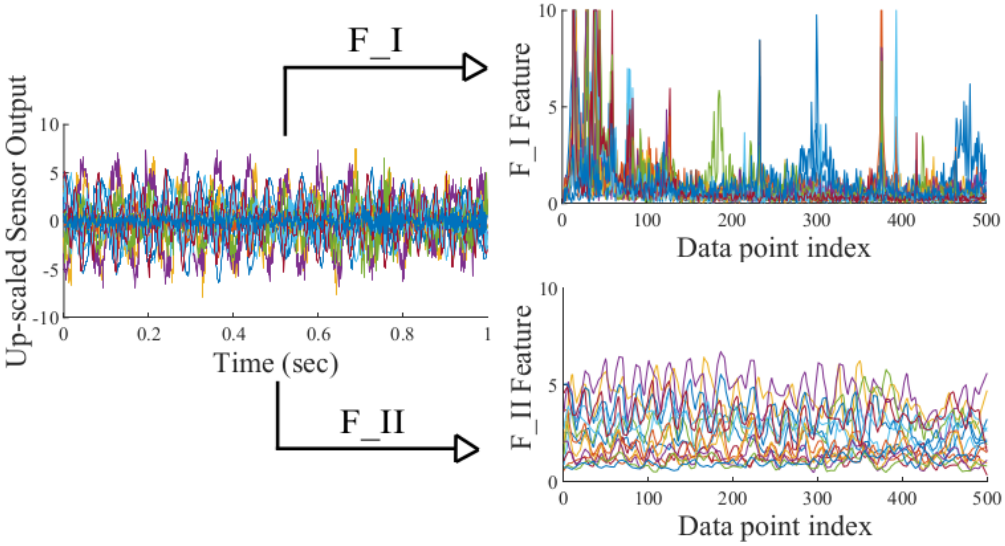


Figure 2.7: A sample F_I and F_II features of a data object, shown for all fifteen data channels.

2.4.2 Network validation and testing results

Yellow Frame dataset is about damage detection and localization; hence, the training, validation, and testing data objects must not be taken randomly due to data leakage. Accordingly, those data objects are taken in chronological order. The LSTM with 200 hidden units passed the validation phase. Even though the initial epoch number for the validation process is ten, the features are so informative that networks reached an acceptable accuracy in two epochs, causing overfitting with more epochs. Hence, two epochs are used for the rest of the Yellow Frame networks' training.

The selected network is trained twenty times, with mean and standard deviation (std) of the test F_1 scores of those twenty trained models presented in Table 2.4. The test's mean

confusion matrix is shown in Fig. 2.8. The overall accuracy is 99.60%, and F_1 scores are high for all classes, resulting from only one-second data objects. Short data objects are noisy, indicating the proposed features and architecture robustness to noisy data. Furthermore, the assumption about features to have sequential attributes and the application of RNNs on those features is justified.

The results are compared to Bernagozzi Bernagozzi et al. [18] to provide context for the test accuracy values of this study. The overall testing accuracy of Bernagozzi Bernagozzi et al. [18] is equal to 88.13% compared to 99.60% for this study. The proposed method has other advantages over Bernagozzi Bernagozzi et al. [18], such as no need for experts to select vibrational modes. Furthermore, the PSIL method applied in Bernagozzi Bernagozzi et al. [18] requires the presence of accelerometers on all stories, unlike the proposed classification method. However, including different formations of sensors that may not cover all stories can be tested in future studies for the proposed method.

The reason for the accuracy improvement is a combination of high-dimensional features coupled with the RNNs ability to recognize high-dimensional features' sequential patterns. In the method utilized by Bernagozzi Bernagozzi et al. [18], the first five modes were considered. The conversion of the raw structural vibration signals to a small set of modal properties is an example of a large dimensionality reduction, resulting in the loss of information. The high dimensional features employed in this study are designed to maintain a large majority of the signals' information. Furthermore, the F_I feature (*i.e.*, a normalized FFT-based feature) is assumed to be a sequence in this study, with its sequential characteristics (*i.e.*, dependencies among spectral lines) linking to a specific structural condition. Hence, RNNs can understand the spectral dependencies among different spectral lines, analogous to temporal dependencies in time-series data, resulting in robust testing results, even with a test-to-train ratio of 8.

Table 2.4: F_1 score statistics, Yellow Frame classification on original dataset.

Damage class	Mean F_1 score ±std (%)	Damage class	Mean F_1 score ± std (%)	Damage class	Mean F_1 score ± std (%)
DC1	99.97±0.05	DC8	99.98±0.06	DC15	98.67±1.11
DC2	99.95±0.14	DC9	99.98 ±0.03	DC16	99.88±0.44
DC3	99.92±0.08	DC10	98.63 ±1.25	DC17	99.57±0.71
DC4	99.94±0.08	DC11	98.76 ±1.10	DC18	99.69±0.25
DC5	100.0±0.00	DC12	99.95 ±0.07	DC19	99.93±0.17
DC6	99.98±0.03	DC13	100.0 ±0.00	DC20	99.92±0.20
DC7	99.91±0.08	DC14	98.07±1.39	DC21	99.99±0.03

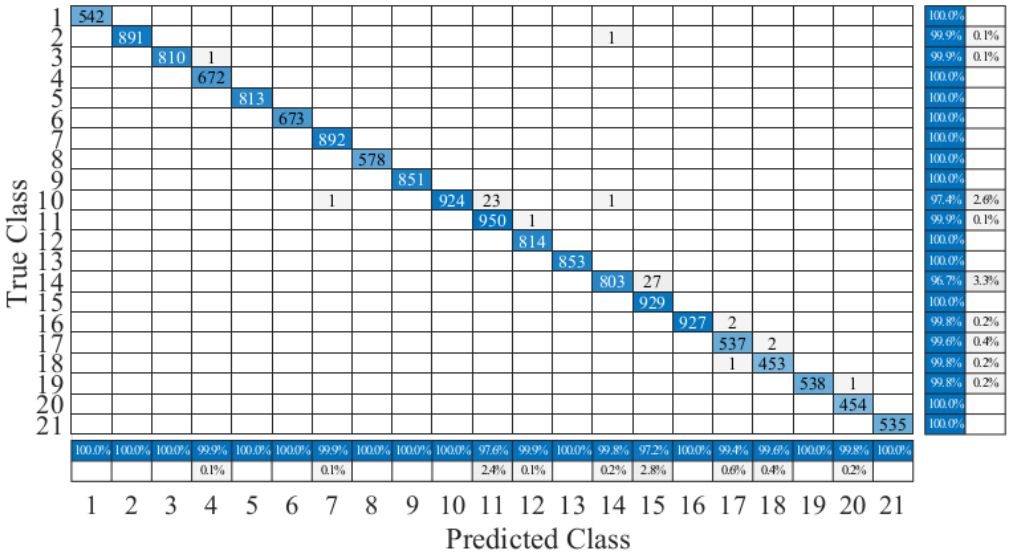


Figure 2.8: The mean Confusion matrix, Yellow Frame classification on original dataset.

2.4.3 Synthesized low-sampled class and GAN-generated data objects

The standard Yellow Frame dataset classes are all well sampled. Therefore, for synthesizing a low-sampled class, DC9 is under-sampled from 105 data objects to 10 data objects. The

extreme reduction is applied because, even though the Yellow Frame structure is not simple, the features are so informative that only such a reduction can affect the classification accuracy by a noticeable amount. The selected network (*i.e.*, LSTM with 200 hidden units) is trained twenty times on the under-sampled training set, and the mean F_1 score of DC9 drops to 73.39%. The GAN network shown in Fig. 2.6 is employed for possible improvement in the low-sampled class's classification accuracy. As discussed, the F_I and F_II features are independent and separate GANs are trained for those features.

For DC9, the GAN's training loss for both Discriminator (DL) and Generator (GL) are shown in Fig 2.9a and 2.9b, respectively. The GAN training process is an open-ended procedure in which the user must decide whether the training is enough or not. The training loss trend can identify a good Generator, in which its reduction or oscillation around the same value while the Discriminator's loss is increased or remained constant can be a reliable stopping criterion. DL and GL losses are computable from Eq. 2.1, and 2.2, in which OF and OR are arrays of outputs from the Discriminator for the generated and the training data objects (real data), respectively. Due to the real data's low dimensionality, the batch size is set to be ten, and both GAN's training of F_I and F_II features are concluded in 5000 epochs.

$$DL = -\text{mean}(\log(OR + \epsilon) + \log(OF + \epsilon)) \tag{2.1}$$

$$GL = -\text{mean}(\log(OF + \epsilon)) \tag{2.2}$$

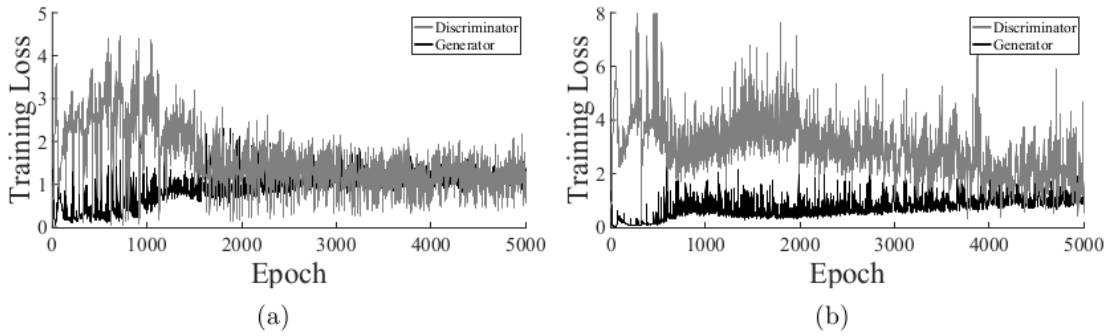


Figure 2.9: The DC9 GAN's training loss for (a) the F_I feature (b) the F_II feature.

2.4.4 Classification improvement via capped GAN-generated features

A group of ten capped generated F_I, and F_II features are shown in Fig. 2.10a, and 2.10b, respectively. Three scenarios are tested to investigate the effects on classification accuracy based on different ratios of real data objects (*i.e.*, the initial training set) and generated data objects merged. Furthermore, the effect of capping on the resulting accuracy values is studied. The scenarios contain 1) a mixture of ten real and capped generated data objects, R-1-C ("1" stands for the merging ratio of one), 2) ten real, and one-hundred capped generated data objects, R-10-C, and 3) ten real and one-hundred non-capped generated data objects, R-10-NC. Five groups of generated data objects are mixed with the primary training data. The five merged datasets are trained four times, resulting in twenty trained classifiers for each scenario. Due to the large confusion matrix (*i.e.*, twenty-one classes of data), only the F_1 score statistics for four classes among different scenarios are shown in Table 2.5.

According to Table 2.5, with capped generations, the improvement is observable in the low-sampled class (*i.e.*, DC9), while no accuracy reduction happens in other classes. The fact that there is no reduction in other classes' F_1 score is important. It shows that the capped

generated data objects' variations are not so high to resemble other data classes and vary enough to result in a better classifier. However, with the non-capped generations, the class itself and other classes have a reduction in accuracy. The accuracy reduction shows that the generated data objects are much different from the real data objects and resemble other classes' features without the capping.

Table 2.5: The mean F_1 score ±std for different merging scenarios of GAN-generated DC9 data objects with the main training set.

Data class	Low-sampled training set	R-1-C	R-10-C	R-10-NC
DC1	88.02 ±6.47	95.84 ±4.38	97.08 ±3.91	74.46 ±11.1
DC4	99.89 ±0.06	99.89 ±0.09	99.89 ±0.09	99.88 ±0.08
DC9	73.39 ±8.38	90.84 ±4.46	91.82 ±3.77	18.22 ±21.0
DC10	94.89 ±2.24	96.84 ±2.06	96.95 ±3.23	89.30 ±5.98

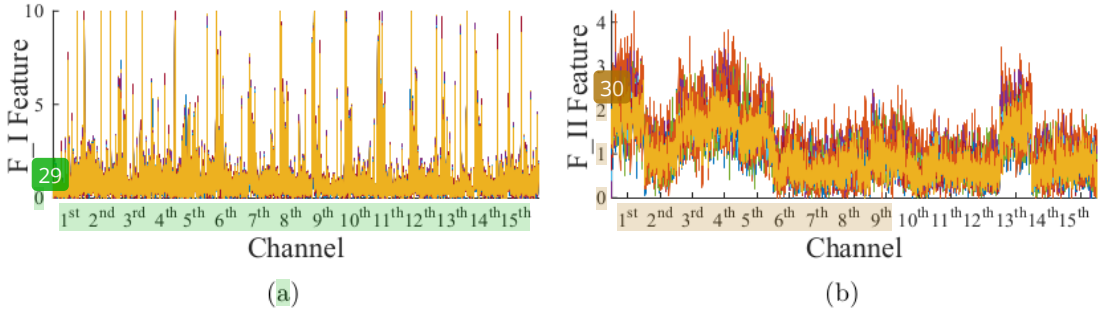


Figure 2.10: GAN-generated data objects' features for DC9 (a) the F_I feature(b) the F_II feature.

2.5 Sutong Bridge's results and discussion

Abnormal sensor output detection (*i.e.*, SOV) is performed on the Sutong Bridge dataset. The classification results are discussed after the most suitable network is chosen in a vali-

dation process, and the frameworks' step-by-step results are discussed. The results are then compared to three other studies on the same bridge's SOV problem. Finally, the GAN-generated data objects are implemented to improve the low-sampled class's accuracy (*i.e.*, PC4).

2.5.1 Network validation and testing results

The selected network has the general architecture as presented in Fig. 2.5 with two branches and one FCL with an output size of seven (*i.e.*, the number of classes). The Bi-LSTM layer with 200 hidden units is selected in the validation process. The finalized net is trained twenty times, with the mean and standard deviation values of the resulting F_1 scores for each class presented in Table 2.6. The mean confusion matrix is shown in Fig. 2.11. The most important class to be identified in SOV problems is the normal class of data, and in the next levels are the different faulty signal patterns. Normal class importance is due to avoiding false alarms in the decision-making processes with the signals labeled as normal by SOV. Hence, the normal class classification accuracy is of greater importance to evaluate the classification results. In the Sutong Bridge dataset, the normal class is PC1. The classification results show an F_1 score of 98.35% for this class. Among the two low-sampled classes (*i.e.*, PC4 and PC7), the PC4 accuracy suffers from its data objects being misclassified as other classes rather than others to be misclassified with this class. The PC7 class has a better F_1 score than PC4, and mostly all its misclassifications are with the PC6 data objects.

The results are compared to three studies: Tang Tang et al. [100], Bao Bao et al. [16], and Arul and Kareem [11]. The first two used a data set with an extended acquisition period (larger set of labeled data), while the last used the same data set as in this study (3% of the extended set). It is more probable for the smaller dataset to have low-sampled classes; hence,

the GAN-generated data objects can be used for possible improvements in the classification accuracy. As mentioned, the normal class of signal pattern (*i.e.*, PC1) is the most vital class to be classified correctly in SOV problems and based on Table 2.6, results indicate an improvement in the PC1's accuracy. PC2 (*i.e.*, missing signals) also shows 100% accuracy, in which the scaling factor might have a generous contribution to the sharp distinction between this class and the other classes. PC5 (*i.e.*, square-signals), PC6 (*i.e.*, trend), and PC7 (*i.e.*, drift signals) are also manifesting improvements on classification accuracy. For PC5, PC6, and PC7 classes, the difference between this study's accuracy values and the best of those studies is 7%, 6.4%, and 12%, respectively. PC3 (*i.e.*, minor signal) shows the same accuracy as the first two studies' best. However, a sharp reduction in the accuracy of the PC4 (*i.e.*, outlier signal) is observed compared to all other studies.

In Arul and Kareem [11], the reported accuracy is based on a test-to-train ratio of one, while the test-to-train ratio of this study is eight. Having a higher test-to-train ratio presents a more challenging scenario. Furthermore, it can show the informativeness of the proposed features and architecture if it results in a better testing accuracy than Arul and Kareem [11]. Among the seven data classes, five classes have better accuracy than that study, while PC3 has slightly less accuracy. However, a decline is observed for the PC4 classification accuracy compared to Arul, and Kareem [11]. The lower PC4 accuracy compared to all those three studies is due to this class having only fifty-two data objects for the training phase. PC4 has the potential for GAN-generated data objects to be merged to evaluate the possible improvements in its classification accuracy. Although the full dataset is not publicly available, based on the improvement in all classes, good accuracy is expected for PC4 if the proposed method is applied to a dataset with more data objects of this class.

Table 2.6: F_1 score statistics, Sutong Bridge classification.

Pattern class	This study's Mean F_1 score ±std (%)	Tang et. al.[100] Mean F_1 score ± std (%)	Bao et. al. [16] Mean F_1 score ± std (%)	Arul and Kareem [11] F_1 score (%)
PC1	98.35±0.68	96.50±0.40	78.3±2.50	95.00
PC2	100.0±0.00	98.90±0.70	98.0±0.30	100.0
PC3	87.23 ±1.99	86.10 ±0.90	44.5 ±4.00	91.00
PC4	44.77±4.71	81.50±1.90	73.9±1.00	67.00
PC5	99.35±0.05	92.30±1.30	58.3±5.00	96.00
PC6	97.40 ±0.27	91.00 ±0.60	87.3 ±1.50	94.00
PC7	81.30±1.39	69.40±1.85	57.4±2.30	66.00
Accuracy	96.50%	93.50%	71.10%	92.97%

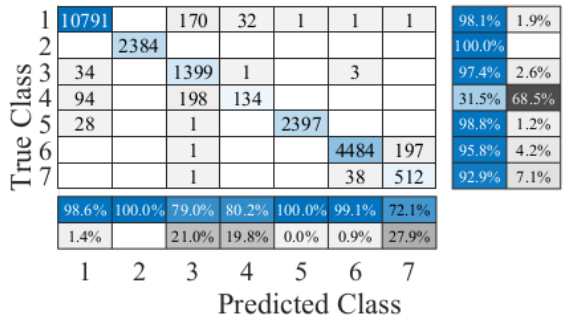


Figure 2.11: The mean confusion matrix, Sutong Bridge classification.

2.5.2 GAN-generated data objects for the low-sampled class, PC4

The GAN network in Fig. 2.6 is utilized to generate data objects mimicking PC4. As discussed previously, two different GAN networks are trained to generate data objects for the F_I and F_II features. Since the data is high-dimensional (36000×1), the inputs for the GAN network (*i.e.*, real data) are down-sampled by a factor of ten for faster GAN training. For PC4, the training loss value for both the Discriminator (DL, Eq. 2.1) and the Generator

(GL Eq. 2.2) are shown in Fig 2.12a, and 2.12b, respectively, with a batch size of fifty-two.

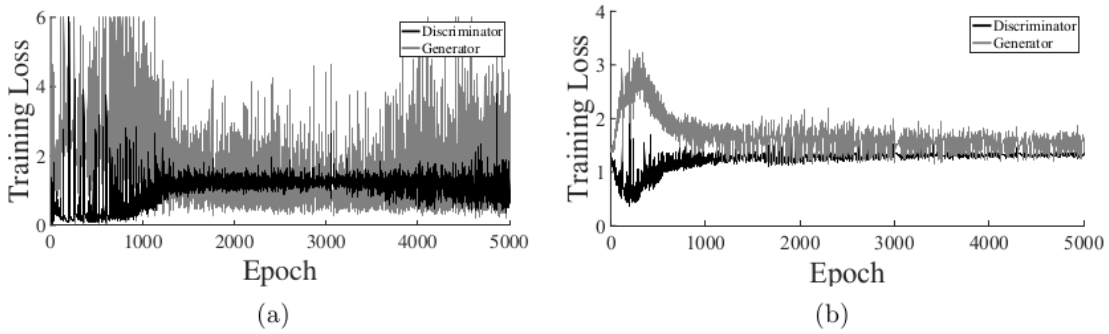


Figure 2.12: The PC4 GAN's training loss for (a) the F_I feature (b) the F_II feature.

2.5.3 Classification improvement via capped GAN-generated features

With the trained GAN, a group of fifty capped generated F_I and F_II features are shown in Fig. 2.13a, and 2.13b, respectively. The generated features indicate an adequate variation level based on the figures, and no mode-collapse failure of GAN is observable for the generated features. Due to under-sampling real data for the GAN's training phase, the generated data objects are 3600×1. Hence, the generated data objects must be over-sampled to 36000×1, which can be achieved through linear interpolation.

Following the same procedure taken in Yellow Frame, three scenarios, R-1-C, R-26-C, and R-26-NC, are examined. The resulting test statistics for those scenarios (*i.e.*, F_1 scores) are shown in Table 2.7. For conciseness, only PC4 results are shown in Table 2.7. The test's mean confusion matrix is shown in Fig. 2.14, in which the effect of integrated GAN-generated data objects on other classes' accuracy is observable. Improvement is seen on both capped versions in Fig. 2.14a and Fig. 2.14b. However, the non-capped generated data objects reduced the classification accuracy in the desired class and other classes (Fig. 2.14c),

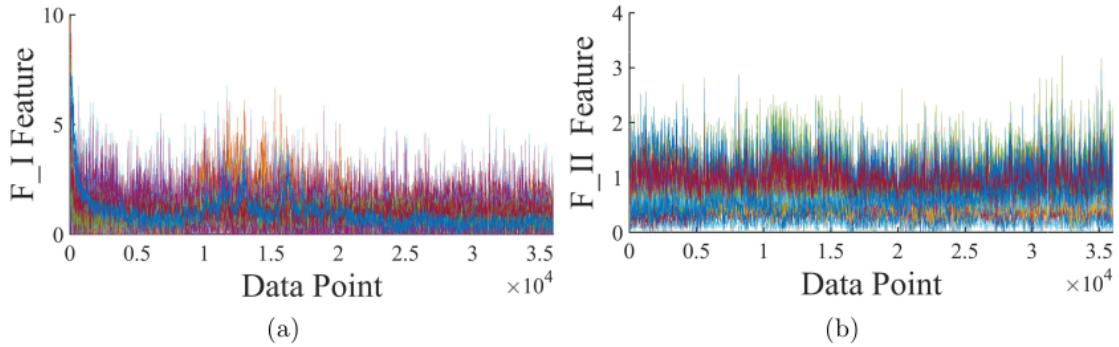


Figure 2.13: GAN-generated data objects' features for PC4 (a) the F_I feature(b) the F_II feature.

implying the importance of applying the capping method to the GAN-generated data objects.

Table 2.7: The testing accuracy for different merging scenarios of GAN-generated data objects with the main training set of PC4.

Scenario	Original Dataset	R-1-C	R-26-C	R-26-NC
Mean F_1 score ±std (%)	44.77±4.71	64.58±4.14	60.67±5.21	16.08±11.92

2.6 Limitations of GAN-generated samples for the studied SHM and SOV datasets

Regardless of the trained classifier's ability to classify the test data objects correctly, the possibility of a test-like object's generation by the GAN's Generator is studied in this section. For the studied datasets, the GAN's potential to generate data objects similar to the test set for the SHM's F_I feature and SOV's F_II feature is illustrated in Fig. 2.15 and Fig. 2.16, respectively. In Fig. 2.15, and for the Yellow Frame case, the first ten data objects (*i.e.*, ID from 1 to 10) are in the GAN's training set, and the data object with the ID of 617 is

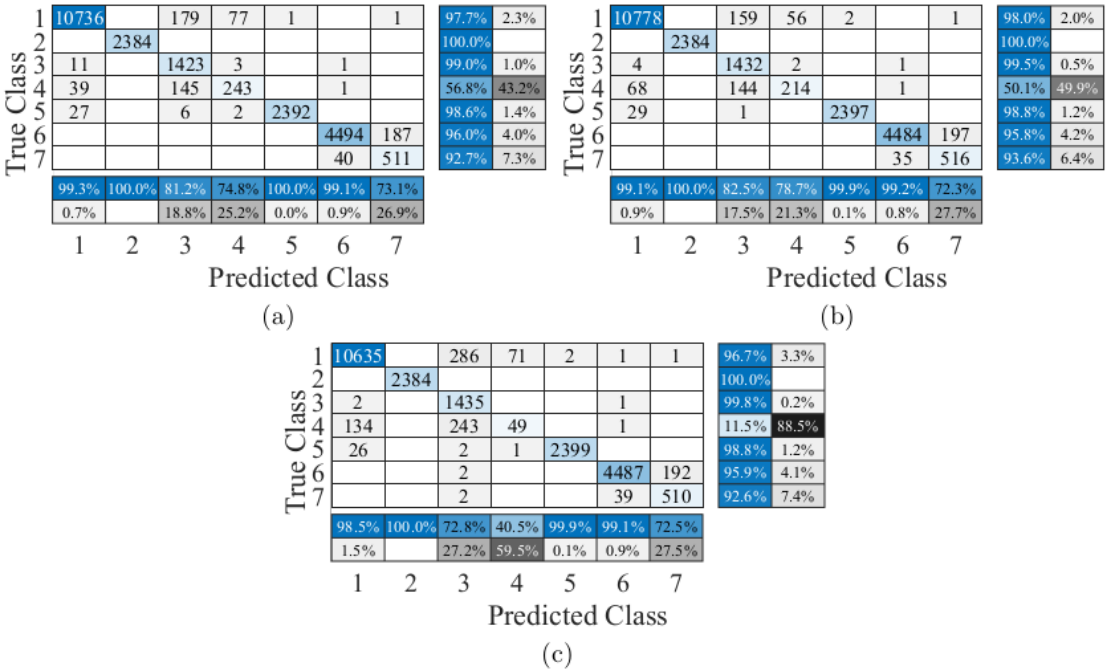


Figure 2.14: The mean classification confusion matrices for different scenarios (a) R-1-C (b) R-26-C (c) R-26-NC.

in the testing set. Based on Fig. 2.15, the test data object's FFT from 0 to 117 Hz is a mixture of the tree training data objects, in which the flat part comes from the ID of 7, and the exaggerated peak at about 100Hz is from the other two. Furthermore, the three peaks on all training data objects from around 200 to 400 Hz have the middle peak as the lowest; however, the Generator can introduce different peaks' intensities to have the second one as the highest peak. In contrast, in Fig. 2.16, and based on the sample training data objects, the Generator cannot generate the depicted test data object with the ID of 43. This trend may reveal why in the SHM case, the accuracy raised to about 92% with the GAN-generated data objects while it raised to 60% in the SOV dataset. Although on both SHM and SOV, the mean F_1 score raised about 20%, reaching the accuracy of 90% is more notable in the SHM case. For the studied datasets, it is more probable that the Generator in the SHM

problem generates real-like data objects than the one trained for the SOV problem. However, to expand this idea on SHM and SOV's general cases, more studies are required with diverse datasets.

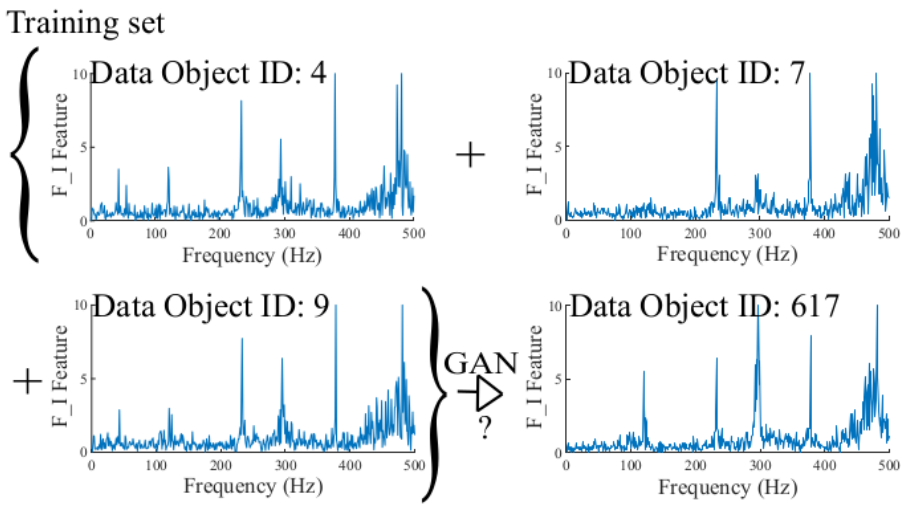


Figure 2.15: Test resembling data object's F_I generations in the SHM problem, DC9.

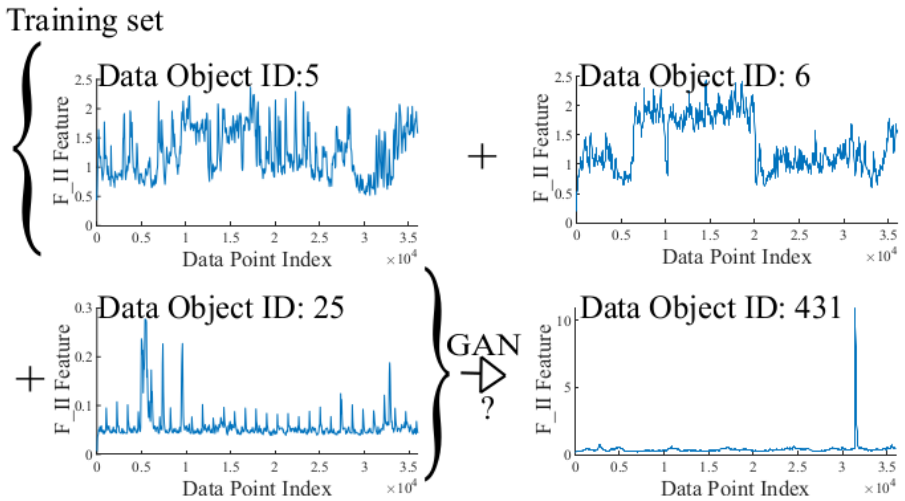


Figure 2.16: Test resembling data object's F_II generations in the SOV problem, PC4.

2.7 Conclusion

This study explores the current challenges in supervised approaches for SHM and the sensor output validation (SOV) frameworks. The main two difficulties are case-dependent feature extractions with considerable dimensionality reduction and the presence of imbalanced datasets with low-sampled classes. In this study, a general classification approach is proposed to alleviate such problems for both SHM and SOV simultaneously. A simple high-dimensional feature extraction method applicable to both SHM and SOV problems is defined. Recurrent neural networks (LTSMs) are utilized to perform the classification. The GAN architecture and its generated data objects are used to improve the low-sampled data class classification. The proposed framework is applied to two datasets: Yellow Frame (SHM) and Sutong Bridge (SOV).

Aside from having a general feature extraction method, the introduced features can lead to accurate classification results coupled with proper neural network layers, the RNNs. The SHM classification results, with twenty-one different damage classes, aside from having high F_1 scores for all classes (mostly above 99%), surpassed Bernagozzi Bernagozzi et al. [18] accuracy values on the same damage configuration with the mean accuracy of 99.6% compared to the accuracy of 88%. For the SOV problem, with a dataset suffering from low-sampled classes, compared to two other studies on an expanded version of this study's dataset, the proposed framework surpassed six out of seven classes of data in classification accuracy. Furthermore, it exceeded a recent study (on the same dataset) [11] that utilized shapelet transformation for the classification in five out of seven data classes. However, this study used a higher test-to-train ratio, eight versus one, which presents a greater challenge. The much higher test-to-train ratio and better accuracy in five out of seven data classes show the informativeness of the proposed features and the use of RNNs to perform pattern recognition

on these features. In SHM, appending GAN-generated data objects to a low-sampled class increased the mean F_1 score from 73.29% to about 91.82%. In SOV, the improvement in the mean F_1 score for a low-sampled class (*i.e.*, PC4) was from 44.77% to 63.84%.

It is also shown that unchecked GAN-generated data objects can contain misleading features resulting from a not well-trained network for a part of the latent dimension. The capping method introduced in this study shows good potential for removing those badly-generated data objects. Without capping, we observed large decreases in the mean F_1 scores, from 73.4% to 18.2% and from 44.8% to 16% for SHM and SOV datasets, respectively. The GAN-generated data objects showed more capacity for improving classification accuracy on the SHM case rather than the SOV case. This phenomenon can be partly due to the different labeling logic of physical- and pattern-based in SHM and SOV datasets; however, a more general conclusion requires further datasets to be studied.

Chapter 3:

On effectiveness of dimensionality reduction for unsupervised structural health monitoring anomaly detection

Mohammad Hesam Soleimani-Babakamali, Ismini Lourentzou, and Rodrigo Sarlo. Does the curse of dimensionality apply to unsupervised shm? investigating the trade-off between loss of information and generalizability to unseen structural conditions. In *Proceedings of the Thirteenth International Workshop on Structural Health Monitoring (IWSHM), March 15-17, 2022 (formerly December 7-9. 2021)*, pages 644–651. DEStech Publications Ltd., 15-17 Mar 2022, Stanford, CA, USA

Abstract

The curse of dimensionality (CD) brings difficulties in pattern recognition problems, such as those found in structural health monitoring (SHM). Dimensionality reduction techniques (DR) make data more manageable by reducing noise and non-informative portions. There exists a trade-off between CD and the loss of information due to the application of DR. Even though in supervised SHM, DR techniques are shown to be effective, for unsupervised

SHM, the trade-off must be assessed due to the unknown data population of novel classes. This study assesses the trade-off concerning a novel method working with a raw frequency-domain feature, the fast Fourier transform (FFT). Different DR techniques are applied to the initial FFT-based feature to assess the trade-off, and detection results are compared. The results indicate that the loss of information can have detrimental effects, such as lowering the detection accuracy by 60% for the autoencoder-based DR. The accuracy reduction is present for all different DR techniques applied in the study; however, regularization lessens the accuracy decrements. This phenomenon indicates the assumption that novelties show themselves in less-vary portions of the baseline condition to be not true.

3.1 Introduction

Structural health monitoring (SHM) is a pattern recognition process [108], identifying novel states with sensor data. Data structures in SHM are mostly high-dimensional, especially with numerous channels. High-dimensional data brings difficulties such as less meaningful distances or a sparse search area, known as the curse of dimensionality (CD). CD and noise make pattern recognition a challenge as innate features of SHM datasets [104]. Dimensionality reduction (DR) can address several problems related to CD [32] as well as the noisiness in data [14]. However, dimensionality- reduction causes loss of information. A trade-off exists between the impact of the loss of information and the benefits of DR on detection accuracy. Several factors impact the trade-off, one of which is whether the analysis is supervised or unsupervised.

In supervised problems, several data classes exist, each of which represents a data population. Those populations define the search space of the problem. Hence, a trained DR technique upon representative training sets can remove redundant and/or uninformative data from

the universal population. In the SHM field, the application of DR in supervised SHM problems has been heavily studied for both structures and machining tools using techniques such as autoencoders [89, 111]. In the unsupervised approach, DR performance is heavily affected by the "generalizability of training sets". As discussed, DR strives to delete the less informative portion from their training set, representing a population. If that population is not representative of all possible observations in the future, then it is not a generalizable set. Numerous studies in the literature have addressed the unsupervised SHM problem but with features resulting from DR techniques. These studies can be divided into two main categories, (1) binary-clustering [79] and (2) dynamic-class clustering [30]. These studies make the fundamental DR assumption: novelties to show themselves in the less-varied portions of the baseline or current state.

Regarding the discussed concerns, the contribution of this paper can be summarized to finding answers for two critical questions in unsupervised SHM frameworks: (1) which one is more detrimental; the curse of dimensionality or applying dimensionality reduction techniques? (2) Is it valid to assume that novelties are discernible on less-varied portions of the baseline state?

3.2 Methodology

An unsupervised SHM technique with high-dimensional or raw inputs as features is needed to assess the trade-off above and assumption. Even though the use of representation learning tools (*i.e.*, deep architectures) is reported in the supervised SHM literature [2], such techniques are scarce for unsupervised SHM. The raw feature-based technique, proposed by **Soleimani-Babakamali** et al. [102], is introduced in this section, based on Generative Adversarial Networks [37], called the GAN-ensemble technique in this study. Several

modifications to the GAN-ensemble technique are introduced in this section to attain such scalability.

3.2.1 GAN-ensemble technique

The modified GAN-ensemble method can be summarized in Fig. 3.1. In this method, T_L incoming data objects are selected to train a GAN model to generate data objects online. The inputs in SHM problems are in the format of acceleration or strain data points with a sampling frequency and N unique channels. D_L dictates how many accelerations/strain data points are merged to construct time-series data objects. The GAN-ensemble technique has two features; (1) high-dimensional and (2) low-dimensional. The first feature is called F and is computed as the half-spectrum FFT of the input signal normalized by its mean value and capped to 10. The second feature in the initial GAN-ensemble technique is the quartiles of energy in F and is omitted as a low-dimensional feature.

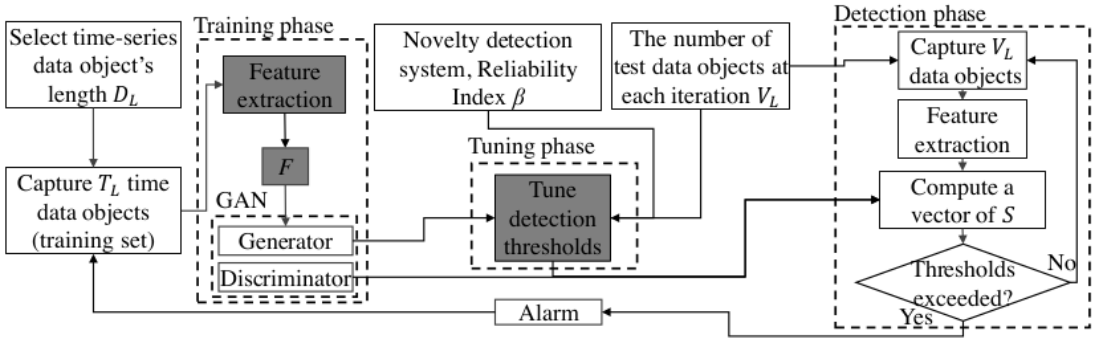


Figure 3.1: The modified GAN-ensemble approach; shaded blocks are the modified modules compared to the initial approach proposed in Soleimani-Babakamali et al. [92].

Training, Tuning, and Detection phases

GAN models are constructed from two neural networks, namely, generator ($\mathcal{G}$) and discriminator ($\mathcal{D}$. The objective function for this model is for the generator to generate "fake" data objects to be similar to the GAN's training set (*i.e.*, "real" data), such that the discriminator cannot capture the difference. The trained GAN provides two neural networks, one to detect novelties and one to implicitly decipher the input data distribution. By assuming the ultimate layer of the discriminator to be a sigmoid function, the novelty detection score for an observation x, S, can be defined as

$$S(x) = -log\mathcal{D}(x). \tag{3.1}$$

The sigmoid function outputs a value between 0 and 1; consequently, S is between 0 and positive infinity with the 0 to indicate the most probable normal data (*i.e.*, no novelty). In the detection and tuning phase, the parameter V_L dictates how many data objects are to be assessed in each iteration of the detection phase. V_L results in a vector of S scores, $\boldsymbol{S}$, from which a limit-state function [92] is defined as

$$P_{20}\boldsymbol{S} - T > 0. \tag{3.2}$$

In this formula, $P_{20}\boldsymbol{S}$ denotes the 20th percentile of the S score vector. This limit-state function fails when the load (*i.e.*, $P_{20}\boldsymbol{S}$) surpasses the resistance (*i.e.*, T), indicating the presence of a novelty. A trained generator and an analogous limit-state function are needed to tune the resistance. The analogous limit-state is defined as

$$P_{80}S(\mathcal{G} - \text{generated}) - T < 0, \tag{3.3}$$

in which $P_{80}S$ represents the 80th percentile of $\mathcal{G}$-generated data instances S scores. The difference between the two functions is that the analogous detector detects no-novel data in the GAN-generated data objects while the initial limit-state function detects novelties in the upcoming data stream; the two detectors are complementary. Monte Carlo histogram sampling can solve the analogous system (*i.e.*, defining T) with a target reliability index of β.

3.2.2 Bias-free comparison

The ratio of parameters in generator and discriminator and optimization parameters are kept constant to have a dimension-bias-free comparison. Moreover, since reliability analysis, from which T is extracted, can be biased towards the discriminator's scores for unseen data objects, instead of a direct T extraction from the observation histogram (*i.e.*, Monte Carlo histogram sampling), the mean value of threshold distribution (*i.e.*, observation histogram) is mapped to the running average of the incoming scores of unseen data objects. T is updated, as shown in Fig. 3.2.

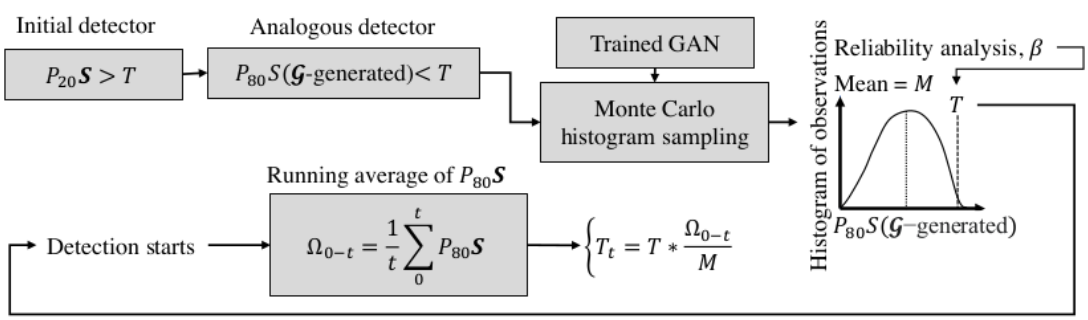


Figure 3.2: Updated T.

3.2.3 Dimensionality reduction techniques

This study applies three dimensionality reduction techniques: PCA, Kernel-PCA, and autoencoders. The PCA method applies to the covariance matrix of its input data, performing a basis transformation with orthogonal projections estimating the covariance matrix. The Kernel-PCA first transforms input data into a higher-dimensional space with a kernel Φ to be linearly separable. Then, similar to PCA, the covariance matrix is built, and the projections are performed. Autoencoders are deep learning-based DR techniques. The reduction process is performed by two sets of neural networks, namely, Encoder and Decoder, acting as shown in Fig. 3.3. The trained Encoder transforms the data into lower dimensions, *e.g.*, start with d_i neurons and end with d_t neurons. These techniques are further explained regarding the studied dataset.

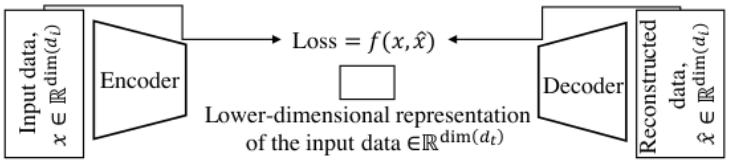


Figure 3.3: Updated T.

3.3 Experimental details

Fabricated at the University of British Columbia, the Yellow Frame structure contains several benchmark datasets for SHM studies [63]. The dataset used in this study has the damage type of brace removal, with 21 data classes. The brace and sensor planning is shown in Fig. 3.4 with the induced damages reported in Table 3.1.

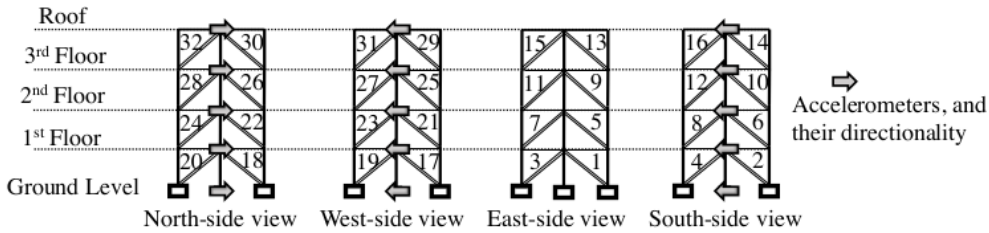


Figure 3.4: Sensor and brace planning for Yellow Frame.

Table 3.1: Yellow Frame dataset summary on accordance with (Fig. 3.4).

Damage class	**Removed brace ID**	**Damage class**	**Removed brace ID**
FDC1	None	FDC2	2,4(II)†
FDC3	FDC2+(18,20)(II)	FDC4	FDC3 +(1,3,17,19)(II)
FDC5	FDC2+(17,19)(II)	FDC6	FDC2+(18,20)(I)‡
FDC7	2(II)	FDC8	(2 , 4)(I)
FDC9	(25,27)(I)	FDC10	(29,31,8,6)(I)
FDC11	(21,23,29,31)(I)		

† One brace is removed.
‡ Both braces are removed.

3.4 GAN architecture and training optimizer

The LSTM-based architecture (Fig. 3.5) is channel-based, in which the data from each channel enters a separate LSTM to be analyzed. Hence, the DR techniques are applied to channels. Based on N and D_L of 1000, a channel dimension of 500 results. By naming the initial dimension of each channel (*i.e.*, 500) as d_i, three dimensions of 100, 50, and 10 are selected to be the target reduced dimensions, d_t. The Kernel-PCA technique is applied with the polynomial kernel of the third degree.

Two fully connected autoencoder architectures (AE) are proposed, shown in Fig. 3.3. The more complex autoencoder named AE-I has more chance to overfit than the simpler one, *i.e.*,

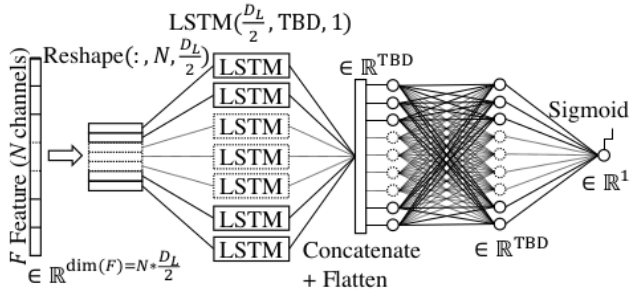


Figure 3.5: LSTM-based discriminator, LSTM (i, h, n), has an input size of i, n stacked layers, and a hidden size of h.

AE-II. The ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 0.0001 is used for their training. Among the autoencoders, the L2 regularization is applied to AE-I, with $\lambda = 0.01$ and $\lambda = 0.025$ to assess the effect of overfitting. Furthermore, a detection scenario with only three sensors, acting as a one-fifth dimensionality reduction, is also carried out to have a bias-free comparison. Three sensors are selected on the first three floors on the frame's west side (Fig. 3.6).

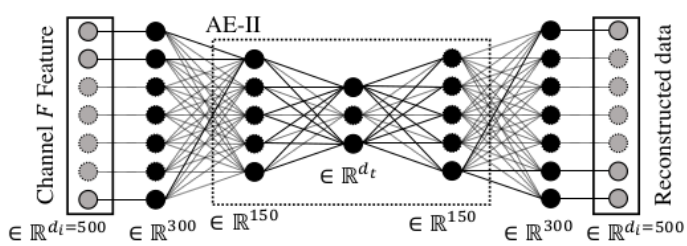


Figure 3.6: AE-I and AE-II architectures.

3.4.1 Comparison

The most important characteristic of the comparison is to be unbiased. A trained GAN contains different pairs of discriminators and generators, equal to its training iterations. The detection results may vary from one combination of discriminator-generator to another

combination. Hence, a fair comparison must be performed to alleviate or cancel the effect of this variation. Thus, a Monte Carlo-based comparison, called "Stochastic Analysis", is practiced. Furthermore, the detection results may differ for a different order of data classes. For example, if detection starts from FDC1 to FDC11, different detections may happen compared to other orders. Hence, for each combination of generator and discriminator (*i.e.*, one detection), classes are permuted, resulting in a total of $11! = 39,916,800$ permutations. The metric for the permutation detection is defined as TAR, the mean number of detected novelties in all permutations divided by 10, with the maximum possible value of 1.

3.4.2 Results and discussion

The Stochastic Analysis results for the PCA-based DR compared to the Normal data and 3-sensor scenario is shown in Fig. 3.7a. Both Normal and 3-sensor data are intact with no DR applied to them. The first notable observation is that the dimensionality reduction, *i.e.*, loss of information, results in lower accuracy metrics. Furthermore, TAR is lower for smaller d_t. All PCA-based detection accuracies are lower than the 3-sensor scenario, showing the reduced accuracy caused by the loss of information. The same analysis is performed for the Kernel-PCA-based DR, with the results shown in Fig. 3.7b). The same trend of accuracy decrement is observable; however, two interesting patterns are discernible. First, the more complicated Kernel-PCA-based reduction with d_t of 100 has lower accuracy than its PCA counterpart, and second, the lowest d_t of 10 has higher accuracy than d_t of 100. This observation is related to the second question, which is about the non-generalizable characteristic of SHM problems. A lower d_t acts as regularization in more complex DR algorithms while it causes more information loss. The higher accuracy in the case of more information loss is a strong indicator of novelties not impacting the less-variable portions of the baseline condition but rather are observable in more noise-like portions of the baseline, which are removed by more

overfitting.

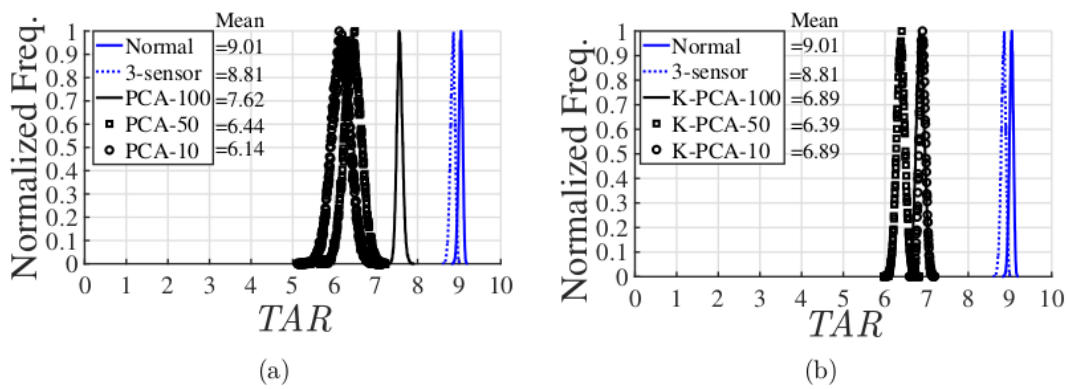


Figure 3.7: Stochastic Analysis: (a) PCA-based DR, (b) Kernel-PCA-based DR.

The results for the autoencoder-based DR (Fig. 3.8) show the same trend. First, detection accuracy with DR is less than normal and 3-sensor scenarios. Moreover, for the AE-I architecture (*i.e.*, the complex autoencoder), the lower the d_t becomes, the higher the accuracy results. Adding regularization can further increase the accuracy; however, there exists a risk regarding regularization. Regularization's effect on accuracy improvement is unpredictable, as having high regularization leads the system to have even more unfavorable results. Selecting the most suitable λ in L2 regularization requires a cross-validation scheme, which is not the case in unsupervised detections. For example, for d_t of 10, λ of 0.01 increases the accuracy; however, λ of 0.025 decreases it. The simpler Autoencoder, AE-II, also has higher accuracy than its AE-I counterpart.

3.5 Conclusion

Due to data interpretability complications caused by the curse of dimensionality (CD), dimensionality reduction (DR) techniques are widely used in unsupervised structural health monitoring (SHM). However, DR causes loss of information. This study investigates two

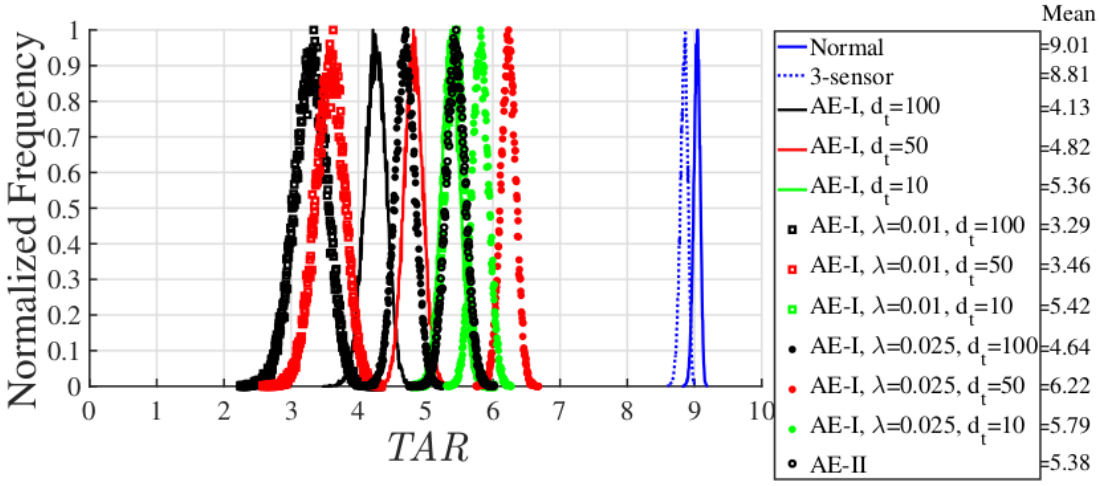


Figure 3.8: Stochastic Analysis; autoencoder-based DR.

fundamental questions regarding SHM: (1) which is more detrimental, CD or loss of information due to DR? (2) the assumption of novelties to affect less-varied portions of baseline data is true?

To answer these questions, and based on the literature, a technique capable of handling raw frequency-domain features is selected. Detection results from raw and DR-based features are contrasted. DR techniques are applied, such as Principal Component Analysis (PCA), Kernel-PCA, and autoencoders. The comparison is carried out on the Yellow Frame dataset with the following observations. First, the impact of CD on novelty detection accuracy is less than the loss of information due to DR techniques. The overfitting of DR to the baseline condition is as detrimental as the loss of information. The regularized autoencoder with the target dimensionality of 10 results in better accuracy than the non-regularized autoencoder with the target dimension of 100 (*i.e.*, less loss of information). The reason is the non-generalizability of SHM problems. The novelties are not observable in less variable parts of the baseline condition, which is the main assumption of using DR.

Chapter 4:

A system reliability approach to real-time unsupervised structural health monitoring without prior information

Abstract

Unsupervised techniques have gained much attention in the last decade as the most practical real-time structural health monitoring (SHM) approach. However, there are still obstacles to robust real-time health monitoring among the proposed unsupervised methods in the literature. These barriers include loss of information from dimensionality reduction, case-dependency of feature extraction steps, lack of dynamic-class novelty detection approaches, and detection results' sensitivity to user-defined detection parameters. This study introduces an unsupervised real-time SHM method, addressing the above four obstacles simultaneously. Furthermore, the proposed technique requires no prior information from structures before going online as a step towards having a general SHM technique. No prior information is common for newly detected novelties, preventing the establishment of detection baselines. Without the baselines, dynamic-class novelty detection cannot take place. Hence, while solving the dynamic-class novelty detection hindrance with Generative Adversarial Networks

(GAN), the framework can be adjusted to be prior-information-free. Online generations of data objects with GAN from the incoming data stream is the key feature that addresses the obstacles above. A mixture of low- and high-dimensional features are used to train multi-ensembles of GAN and one-class joint Gaussian distribution models (1-CG). A novelty detection system of limit-state functions based on GAN and 1-CG models' detection scores is constructed. The Resistance of the limit-state functions is tuned to user-defined detection parameters with the GAN-generated data objects through reliability analysis. The tuning makes the detection results robust to the user-defined detection parameters. The proposed novelty detection framework is applied to two standard SHM datasets to illustrate its generalizability: Yellow Frame and Z24 Bridge. All different damage categories are identified with low sensitivity to the initial choice of detection window length by employing the proposed dynamic and static baseline approaches with few or no false alarms.

4.1 Introduction

Cities are a mixture of different structures, such as buildings, bridges, and other physical infrastructure. Incorporating Structural Health Monitoring (SHM) into such a vast infrastructure network is vital to save resources, reduce repair costs with timely maintenance, and save human lives. This task can be accomplished via inexpensive and highly accurate sensors and robust novelty detection frameworks. A novelty in a structure is a change in the acquired sensors' data resulting from damage in the system or a shift in the structure's response due to environmental factors.

Generally, the SHM task is accomplished via either model-based or data-driven approaches. In model-based approaches, typically, a finite element model or modal flexibility matrices [19] are fitted or updated, based on experimental sensor data [45, 104]. By regularly updating a

model, possible damages and their locations can be discovered. However, these methods rely on complex optimization algorithms and dense sensor deployments for capturing many mode shapes, including local modes, [35]. Furthermore, the need for the presence of an expert and the vulnerability of detection results to noisy data [45, 104] make this type of approach unsuitable for real-time SHM with noisy data. Moreover, missing a novelty is probable if the model's physics cannot capture its behavior [13]. Some improvements have been achieved by assuming probabilistic model parameters [35, 120]; however, the proposed methods are mostly validated versus numerical simulations and laboratory testings and need further field experiments to prove their validity [120].

Data-driven approaches, including supervised, unsupervised, and semi-supervised strategies, are alternatives to the model-based techniques. Their use of prior data can distinguish supervised and unsupervised approaches. Supervised methods have labeled datasets, regardless of techniques employed in their feature extraction steps. On the other hand, unsupervised methods have unlabeled datasets or no prior data. A typical example of a supervised method is applying subspace identification while using the angles between subspaces as damage indicators [109]. Another example is the use of 1-D or 2-D convolutional neural network layers for damage classification [3, 115], shown to work well even in noisy situations. However, supervised methods are not well-suited to the real-time SHM of structures. Gathering enough data objects of different structural conditions from operating facilities is costly and impractical in most cases. Some efforts have been made to alleviate this issue, such as having only two classes of an undamaged and fully damaged system for performing the damage detection [3]. Unfortunately, obtaining fully damaged states is a big challenge in itself. Another solution to the labeled data scarcity problem is to employ deep generative models to generate data objects from minority classes, which has gained attention in recent years. GANSO [83] is a method developed based on Generative Adversarial Networks (GAN) and

vector Markov Random Fields. This method can improve the supervised classification of data classes in highly imbalanced data sets in different areas, including classifying sleep stages using electrocardiographs as time-series data. Exclusively for supervised SHM, GAN-based augmentation techniques have shown improvements for the classification of datasets with imbalanced classes in studies, such as Zareapoor *et al.* [116], and Soleimani-Babakamali *et al.* [102]. However, GAN-generated data objects could contain misleading generated data objects. Such a phenomenon should be considered while mixing the initial data classes with the generated data objects.

Nevertheless, supervised data augmentation strategies still require *some* labeled data, making unsupervised techniques the best-suited to real-time SHM because of no prior, labeled information or assumptions of abnormal structural states. However, current unsupervised approaches still have several challenges, particularly when applied across multiple structures. Based on a review of state-of-the-art studies in the unsupervised SHM field, we have identified four key challenge categories that limit the scalability of these techniques: loss of information from dimensionality reduction (*loss of information*), case-dependency of feature extraction methods (*case-dependent feature extraction*), lack of a dynamic-class novelty detection (*non-dynamic novelty classes*), and the novelty detection's sensitivity to user-defined detection parameters (*sensitivity to detection* parameters), each of which is described below.

The first challenge is the *loss of information*. Most unsupervised learning studies involve large dimensionality reduction (DR) to avoid the curse of dimensionality [98] specifically for clustering. However, in an unsupervised novelty detection framework, the future damages are generally unseen. DR techniques can ensure the discarded information is not important for describing their training set and the population it represents. However, it may not work for an unseen data object from another population (*i.e.*, an untrained novelty). Some specific deep neural network architectures, such as one or two-dimensional convolutional

layers, RNNs, or deep fully-connected layers, have been shown to classify novelties (*i.e.*, supervised) with good performance with little to no DR [2, 3]. Such deep architectures must be trained with large-scale data [97] while having a loss function to be optimized. In a supervised approach, the loss function is constructed from the classification accuracy, which is not the case in unsupervised techniques with no labeled data. Hence, to take advantage of such deep architectures, novel methods that employ them in an unsupervised setting with unsupervised-based loss functions are of interest. GAN models have a loss function that enables the training of such deep networks unsupervised. However, as further explained in the forthcoming sections, GAN is not solely selected for the discussed matter, and its other capabilities have been amongst the reasons for its selection. 2-D GAN was applied in unsupervised anomalous sensor output validation [61]. In that study, sensor outputs (*i.e.*, time-series data) are transformed into Gramian Angular Field images to train a GAN model. The GAN's generator is then embedded as the decoder of a deep autoencoder network, with the autoencoder's reconstruction loss as the novelty detection criterion. In SHM, on the other hand, 1-D data are of more interest (*i.e.*, frequency or time-domain data series). The 1-D GAN idea for constructing lost sensor data [52], and the classification improvement of low-sampled damage scenarios [91, 116] are implemented in SHM.

The second challenge, *case-dependent feature extraction*, refers to the limitations in applying the feature extraction steps (*i.e.*, high or low-dimensional) that are trained or defined for a specific case study to a new structure or across multiple facilities. For example, it is impractical to train deep auto-encoders [59, 70, 107, 107], deep Boltzmann machines [75], or Kernel-based PCA approaches [50] on all the structures in a network. Applying those techniques to a new structure from scratch is time-consuming since they should be trained to extract new relevant features. Defining a simple, meaningful extraction technique can aid its generalization to any new structure. Still, such a simple extraction technique may need

special tools (*e.g.*, neural network architecture) to handle the resulting features that can be high-dimensional.

The third challenge is *non-dynamic novelty classes*. In the unsupervised SHM literature, the most practiced method is to construct a detection baseline based on previously recorded data on the normal behavior of structures. That baseline is later used to have a binary classification of normal and abnormal states. Having one-class novelty detection or support vector machines are examples of this approach[86, 94, 95, 107, 110]. For having more than one cluster of novelties, the user may define a static number of clusters before the detection starts [23, 24]. However, one-class models cannot record and save different damage scenarios, and the correct number of classes is unknown at the start of the detection phase. A major benefit of a dynamic-class novelty detection approach is that it can distinguish between different novelties. Hence, if the same novelty returns in the future, it recognizes it without alarming the presence of a new novelty. Bouzenad *et al.*, in a semi-supervised approach [20], defined a threshold for the distance of a new test data object from available clusters to increase the number of clusters (*i.e.*, dynamic-class novelty detection) in case of the threshold being exceeded. However, the lack of a probabilistic framework to tune such thresholds can reduce its effectiveness in practice, as different thresholds could generate more or fewer clusters than the ground truth clusters. The first approach, a binary classification, can turn into a dynamic-class novelty detection approach if there is enough data of any new novelty it faces. This way, whenever a novelty is detected, and with adequate data samples from it, a new baseline can be constructed to detect the next novelty from the current novel status of the system. Nevertheless, in a real-settings, gathering enough data from any novelty that occurs may require months of observations to gather a dataset as comprehensive as the one employed for the normal behavior, which is unrealistic.

The fourth challenge is the *sensitivity to detection parameters*. User-defined (detection) pa-

rameters, such as the number of time-series data object to undergo the detection process in each detection-iteration (*i.e.*, detection window length) [30], or detection thresholds for having a dynamic number of clusters [20], can result in a variety of clusters, or alarms contradictory to the ground truth (*i.e.*, false alarms). The detection window length is negatively correlated to the novelty detection power and positively correlated with having less novelty detection while the system is intact (false alarms) [30, 84]. In studies dealing with the detection window length, different lengths were studied [30, 35, 84], and the one with the best detection result was introduced as the best candidate. Nevertheless, tuning the detection window length is impractical in a real-time detection process with unseen data, as it must be defined before the analysis initiation.

4.2 Research Significance

In the current unsupervised SHM literature, certain studies address some of the four mentioned challenges; however, the framework introduced in our study addresses all of them simultaneously. Furthermore, while solving these challenges, it becomes possible to propose a general unsupervised real-time SHM technique that requires no prior information about a structure, labeled or unlabeled. Such a technique is a step towards a general unsupervised novelty detection framework in SHM that can be applied to different structures of interest with few modifications. This characteristic is advantageous in monitored cities with numerous distinct structures with no prior data available. The philosophy and significance behind our approach are explained in the following points.

- The *loss of information* challenge can be solved by having general domain-specific features or raw signals as inputs without handcrafted or DR-extracted features. This study solves this challenge by having the half-spectrum fast Fourier transform (FFT)

of signals as a high-dimensional feature. The half-spectrum FFT can be considered the raw frequency-domain information of signals. Employing high-dimensional features, or even raw signals, are practiced in the supervised literature [2, 116]. However, employing high-dimensional features without DR, such as the FFT of all channels that can be of a dimension of tens of thousands, is scarce in the unsupervised novelty detection literature. We contribute a novel approach to handle such high-dimensional features in an unsupervised setting via the use of GANs.

- The *case-dependent feature extraction* challenge is also a major consideration in choosing the FFT of input signals as features. Taking the FFT of signals is a universal approach in signal processing that is not specific to monitored structures. By avoiding significant DR on the FFT features, we also avoid a bias towards certain structures. No case-dependent feature extraction schemes can be considered a step towards having a general SHM framework applied to different structures or across multiple structures.

- The non-dynamic novelty classes issue is addressed with the application of GAN. As mentioned in the introduction section, baseline methods, with prior data, showed promising results dealing with binary unsupervised SHM problems, even with universal and simple models. Examples included one-class support vector machines [107], K-nearest neighbors [86], or single thresholds and linear regression models [94, 95]; however, they cannot distinguish between different types of novelties. Upon encountering a novelty and with enough data, another baseline is tunable. This study proposes a GAN-based framework to generate a diverse dataset for each detected novelty from the incoming data stream. Apart from data generation, GAN comprises a binary classifier (*i.e.*, discriminator) to define a new baseline for each encountered novelty, building a dynamic, multi-class detector in a fully online fashion. In this system, novelties can be distinguished from each other in an unsupervised setting.

- The *sensitivity to detection* parameters is addressed by applying a well-established probabilistic approach: reliability analysis. A key contribution of our framework is that detection window length and thresholds are linked based on reliability analysis, something scarce in SHM literature. Recently, reliability-based techniques were applied in a model-based SHM approach by assuming physical model parameters as random variables, incorporating the probability of damage detection and false alarms to define the minimum detectable damages [64]. In our study, since GAN generates new data objects from a latent dimension (*i.e.*, a probability density function), sampling-based reliability techniques such as Monte Carlo histogram sampling can be applied. This technique can account for user-defined detection window length, tuning the detection thresholds to reach a pre-defined reliability index while abiding by a precept. To the authors' best knowledge, this is the first application of reliability-based techniques in a data-driven SHM context (as opposed to the model-based approach in [64]).

- The final contribution of the framework is the concept of *prior-information-free monitoring*, which arises from the combination of the various techniques discussed above. We mean a method that does not demand any data before going online by a prior-information-free approach. The major benefit of prior information is for the numerical model to observe and learn the data fluctuations caused by normal environmental and operational variations in operational structures. Such fluctuations aid the model in distinguishing between natural and novelty-caused variations. The need to observe data fluctuation is addressed by the variation of GAN-generated data instances with no prior data. It generates vast amounts of data objects from the incoming data stream, which are not seen before. The proposed method is not model-based, so no prior information about the structure itself is required and is unsupervised, functioning with no labeled data from different structural states.

The proposed novelty detection framework is tested on two datasets, Yellow Frame with twenty-one classes of normal and damaged states and Z-24 bridge on its last month before demolition, with multiple damage scenarios. In what follows, both datasets are explained, and then different parts of the methodology are introduced.

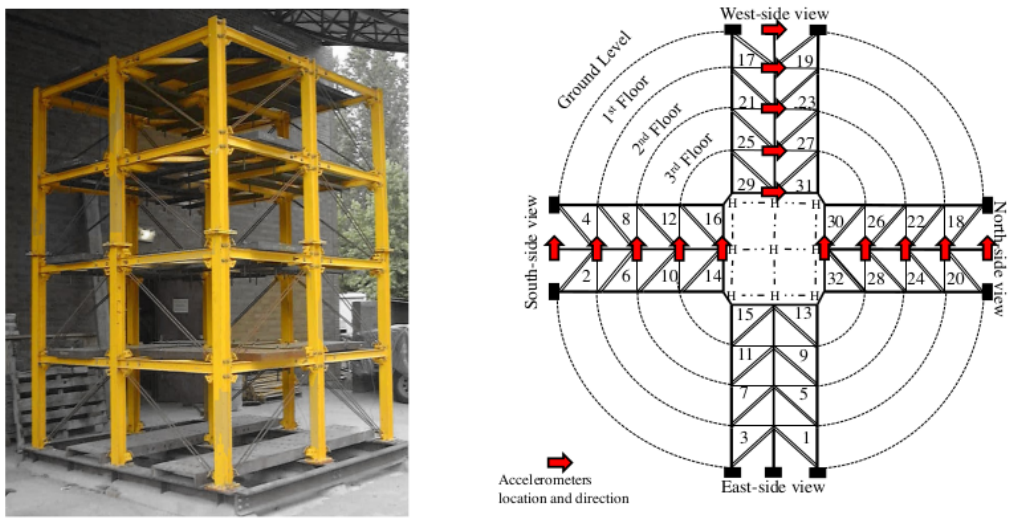
4.3 Case studies

For real-time SHM, and specifically for evaluating how the proposed framework works, experimental datasets including multiple damage scenarios with defined damage initiation points are essential. In this section, two such datasets are introduced: Yellow Frame and Z24 Bridge. A summary of both datasets' structures, a brief description of their acquisition systems, and the datasets summaries are provided. Both datasets exhibit real SHM problems with all possible difficulties, such as noise in the data. The outcomes of this study for these two datasets must be compared with competitive studies to highlight the improvements. For the selection of competitive studies, two aspects are to be considered. First, they must employ the detection in an unsupervised manner, and second, they must investigate the same datasets for a meaningful comparison. The competitive studies for the comparison are introduced in this section.

4.3.1 Yellow Frame

Yellow Frame [63] is a one-third modular four-story steel frame established at the University of British Columbia, as shown in (Fig. 4.1a). Modular frame components (*i.e.*, braces, beams, and masses) can be removed to simulate diverse damage scenarios. There are fifteen accelerometers on the structure, capturing the ambient vibration response with a sampling

frequency of 1000 Hz. Readers are referred to Mendler *et al.* [63] and Allahdadian [4] for details of the structure and instrumentation. The unsupervised real-time novelty detection results of Yellow Frame are compared with Rharã de Almeida Cardoso *et al.* [30]. A dataset with seven data classes of Yellow Frame with the same damage type (*i.e.*, brace removal) was studied unsupervised in that study.



(a) (b)

Figure 4.1: Yellow Frame (a) structure (in color) (b) brace labeling and sensors locations.

For this study, a benchmark dataset with twenty-one different structural configurations is selected, as presented in Table 4.1. This dataset includes one "normal" class and twenty damage classes, simulated by removing different structure braces. The table also shows the fundamental frequency (*i.e.*, fist-mode frequency) of different classes. Those frequencies are obtained with ARTeMIS software [1] with the Enhanced Frequency Domain Decomposition method [21]. The Yellow Frame dataset was checked to have no malfunctioning sensors during the data acquisition. However, a sensor output validation (SOV) algorithm must be present in more general cases before the data stream enters the novelty detection framework. SOV problems are well studied, and not only can be used to detect malfunctioning sensors [16,

61, 102], but also to reconstruct damaged or lost sensors' data [52].

Table 4.1: The dataset summary for Yellow Frame, including various damage classes simulated by removing certain bracing components, with labels shown in Fig. 4.1b.

Damage class	Removed brace ID (NO.)	F.F (Hz)*	Damage class	Removed brace ID (NO.)	F.F (Hz)
FDC1	None	7.62	FDC12	FDC11 + (17, 19, 25, 27)(I)**	6.77
FDC2	2,4(II)***	5.79	FDC13	FDC8 + (1, 3, 17, 18)(I)	6.40
FDC3	FDC2 + (18, 20)(II)	4.76	FDC14	(10, 12)(II)	6.41
FDC4	FDC3+ (1, 3, 17, 19)(II)	4.88	FDC15	FDC14 + 21(II), 23(I)	6.52
FDC5	FDC2 + (17, 19)(II)	5.38	FDC16	(21, 23)(II)	6.87
FDC6	FDC2 + (18, 20)(I)	5.40	FDC17	(7-8, 21, 22)(I)	6.98
FDC7	2(II)	7.19	FDC18	(5, 6, 7, 8, 21, 24)(I)	6.58
FDC8	(2 , 4)(I)	6.85	FDC19	FDC18 + (7, 8, 21, 22)(I)	6.04
FDC9	(25, 27)(I)	7.29	FDC20	FDC19 + (5, 6, 23, 24)(I)	4.92
FDC10	(29, 31, 8, 6)(I)	6.90	FDC21	(6, 8)(II), (21, 22, 23, 24)(I)	5.40
FDC11	(21, 23 ,29 , 31)(I)	7.18			

* Fundamental mode frequency
** One brace is removed
*** Both braces are removed

4.3.2 Z24 Bridge

Z24 Bridge overpassed the A1 highway between Bern and Zürich in Switzerland. There are two datasets on Z24 Bridge; a long-term and a short-term dataset. The long-term dataset was captured over a year, while the short-term dataset was acquired over a month. Long-term monitoring was mostly employed in the literature [94, 95] to build a baseline from the intact bridge to detect the damaged state of the bridge over the short-term monitoring. The long-term monitoring has a variety of ambient temperatures from around -10 to 40

degrees of Celsius. The temperature below zero could have substantial effects on the bridge's vibrational characteristics [71], and should be accounted for carefully during the baseline definition. However, the temperature above 0 degrees Celsius did not have much of an effect [71]. During the short-term monitoring, the ambient temperature varied around 10 to 35 degrees of Celsius. Furthermore, 17 different data classes are captured successively over a month with a mean time interval of two days between the data acquisitions. Consequently, since we detected damages by only employing the short-term dataset, the temperature is not a decisive factor. In fact, we purposely avoided using the long-term data set for baseline training to demonstrate the effectiveness of the prior-information-free approach.

A schematic of the bridge is shown in Fig. 4.2. This study's Z24 Bridge dataset consists of vibration and environmental monitoring of the bridge with various exerted damages before its demolition with a sampling rate of 100 Hz. Further information on the bridge and the sensor setups can be found in Reynders and De Roeck [77]. The dataset summary, including different types of damages, is shown in Table 4.2. In this table, the fundamental mode's frequency is also reported for different damage scenarios, demonstrating the global effect of damages on the system. Among the 17 data classes, three can be regarded as reference measurements (*i.e.*, intact system): BDC1, BDC2, and BDC8. In BDC8, an extra part is added to the bridge, and the monitoring continues from that point. Since only one reference measurement is enough for novelty detection, BDC2 is taken as the starting point, and BDC8 is skipped to comply with the order of data. According to Table 4.2, there are seven damage types with varying intensities, for a total of 14 damaged classes. One example is the lowering of the pier in various distances ($PL_{20} - PL_{95}$).

The original dataset contained sixteen accelerometers with different orientations (*i.e.*, vertical, horizontal, or transverse channels) on the bridge resulting in numerous data channels. To ensure that anomaly detection was limited to structural damages and not instrumentation

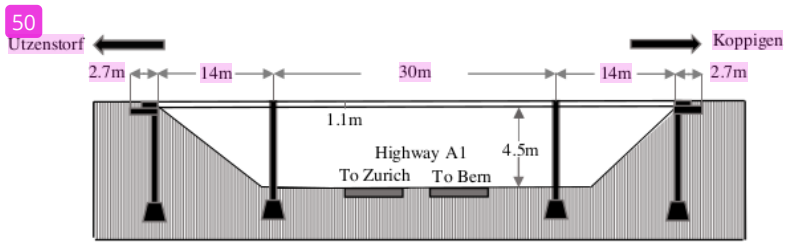


Figure 4.2: Z24 Bridge [78].

failures (*e.g.*, dead or square wave channels), we manually reduced the number of channels to those with no observable defects during the fifteen data classes, resulting in a total of eighteen channels. The selected channels are reported in Table 4.3, with the sensors' labels explained in the Z24 dataset [78].

Table 4.2: The dataset summary for Z24 Bridge, including various global and local damage scenarios.

Data class	Description [77]	Fundamental mode frequency (Hz) [22]
BDC1	First reference measurement (Ref_1)	(Not used)
BDC2	Installation of pier settlement system (Ref_2)	3.88
BDC3	Lowering of pier, 20 mm (PL_{20})	3.87
BDC4	Lowering of pier, 40 mm (PL_{40})	3.86
BDC5	Lowering of pier, 80 mm (PL_{80})	3.76
BDC6	Lowering of pier, 95 mm (PL_{95})	3.69
BDC7	Lifting of pier, tilt of foundation (T_F)	3.84
BDC8	Second reference measurement (Ref_3)	(Skipped)
BDC9	Spalling of concrete at soffit, $12m^2$ (SC_{12})	3.87
BDC10	Spalling of concrete at soffit, $24m^2$ (SC_{24})	3.86
BDC11	Landslide of 1 m at abutment (LS)	3.85
BDC12	Failure of concrete hinge (F_{CH})	3.85
BDC13	Failure of 2 anchor heads (F_{2A})	3.85
BDC14	Failure of 4 anchor heads (F_{4A})	3.84
BDC15	Rupture of 2 out of 16 tendons (TR_2)	3.85
BDC16	Rupture of 4 out of 16 tendons (TR_4)	3.83
BDC17	Rupture of 6 out of 16 tendons (TR_6)	3.83

Table 4.3: Selected channels' labels, Z24 Bridge dataset.

Label [77]	Sensor	Label [77]	Sensor	Label [77]	Sensor
Setup 1	R2V	Setup 2	106V	Setup 2	304V
Setup 2	R1V	Setup 3	R1V	Setup 4	116V
Setup 4	314V	Setup 4	R2V	Setup 5	121V
Setup 5	319V	Setup 5	R1V	Setup 5	R2V
Setup 6	126V	Setup 6	R2V	Setup 7	131V
Setup 8	136V	Setup 8	R2V	Setup 9	R1V

Currently, to the authors' best knowledge, there are no dynamic-class, unsupervised, real-time novelty detection studies on this dataset to evaluate and compare results. With the comprehensive long-term Z24 dataset, most studies have focused on binary novelty detection between the long- and short-term datasets, with the latter having the damaged states. From the most recent state-of-the-art binary classification-based studies on Z24 Bridge, the results of Wah *et al.* [95] is employed for the comparison. It is worth mentioning that in this study, the proposed detection algorithm starts from the reference class (*i.e.*, BDC2, FDC1) and then continues to damage classes in the same order that those classes were captured (*e.g.*, FDC2 after FDC1). Having the same order of data in the data acquisition and novelty detection can result in unbiased detection results.

4.4 Methodology

A general overview of the proposed unsupervised real-time SHM methodology is depicted in Fig. 4.3. The method consists of three main phases: training, tuning, and detection. Before starting the training phase, the user should select D_L, which determines how many data points (*e.g.*, acceleration points) represent a single *time-series data object*. In addition, the user should select T_L and the detection parameter V_L, indicating the number of time-series

data objects used in the training and detection phases, respectively. As will be discussed, the methodology is designed to be robust to the choice of V_L.

The training phase captures the first T_L time-series data objects from the incoming data stream (*i.e.*, online) and extracts FFT-based features, F_I and F_II, for training the GAN and one-class joint Gaussian (1-CG) models, respectively. The trained GAN discriminator and 1-CG models output a novelty score (probability) used for classifying new data objects. The GAN generator is used to synthesize new data for the reliability analysis in the tuning phase.

The tuning phase aims to determine appropriate novelty detection thresholds for the GAN discriminator and 1-CG scores. This tuning, a key novelty of this work, is vital for any method since arbitrary choices of parameters with a static threshold can lead to many false alarms or undetected novelties [20, 30, 84]. GAN models are trained on a limited number (T_L) of data objects' F_I features from the data stream. The trained GAN can generate new F_I features, from which F_II features are obtainable, and are utilized to create a dense distribution of scores. A reliable threshold can be selected from those distributions that can result in novelty detections that are not sensitive to (*i.e.*, independent from) user-defined detection parameters.

Finally, the detection phase tests each new set of V_L data objects for novelties by either a static or dynamic baseline approach. In the static baseline approach, the detection thresholds are tuned only once, while in the dynamic baseline approach, the thresholds are tuned each time a novelty is detected. Further details are provided in the following subsections.

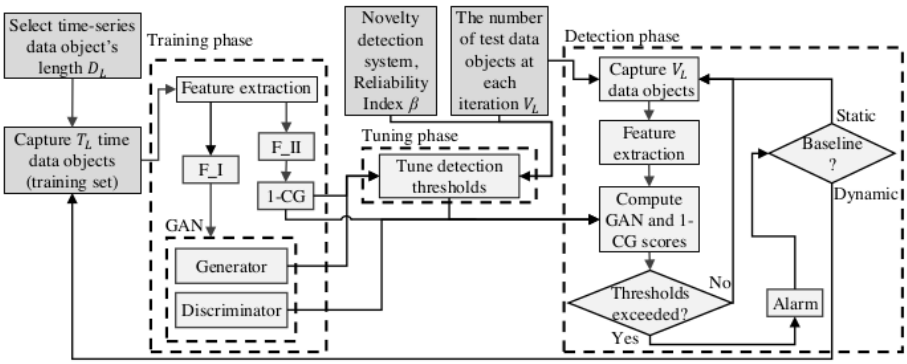


Figure 4.3: Unsupervised real-time SHM framework; shaded blocks represent user-defined parameters.

4.4.1 Parameter Selection

The novelty detection system requires four parameters to be defined by the user: D_L, T_L, V_L, and the reliability index, β. The reliability-based tuning framework is designed to make the detection results robust to the choice of V_L. The considerations for the selection of other parameters are explained in what follows. The only consideration for selecting D_L (number of data points per data object) is that the resulting FFT-based feature is high-dimensional. As explained later in Section 4.4.2, the half-spectrum FFT feature has a dimension of $D_L/2 \times 1$. However, since the utilized datasets are limited, D_L is selected to result in enough data objects for a meaningful statistical analysis of results. Each dataset's total number of detections is the total number of data points divided by $D_L * V_L$. The Yellow Frame dataset has 19,258,000 data points across all classes. With $D_L = 1000$, and $V_L = 10$, this results in 1925 total detection and a mean of 92 detections for each of the 21 classes. For the Z24 dataset, 1,036,800 data points are present. Choosing $D_L = 200$ and $V_L = 10$ results in 5184 detections. With 15 classes, the mean number of detections for the Z24 dataset is around 38. Hence, the size of the Z24 dataset limits the selection of D_L. However, in a real setting, D_L can be selected freely as long as it results in a high-dimensional F_I feature

(*e.g.*, several thousand).

For generic novelty detection of a structure or across multiple structures, T_L can be selected based on the monitoring duration, which can be several days or months. It is possible to use a dynamic T_L, in which the novelty detection framework is initiated with a short T_L, and it dynamically increased after a pre-defined period of novelty-free monitoring. However, for the limited datasets in this study, T_L must be selected to ensure that T_L is less than the number of data objects in the classes. The proposed methodology is unsupervised; however, the ground truth is used for assessing the method's accuracy. Since false alarms and how the proposed tuning method affects them are to be studied, values of T_L resulting in a high test-to-train ratio during the accuracy computations are desired. A high test-to-train ratio creates a challenging environment concerning the algorithm's sensitivity to false alarms. It can also illustrate the algorithm's sensitivity to novelties (true alarms) with limited training data. For Yellow Frame, $T_L = 100$ is selected, which results in a mean test-to-train ratio of 8 for different classes. For the Z24 dataset, a $T_L = 75$ results in a mean test-to-train ratio of 5 for data classes. Both test-to-train ratios are much higher than the conventional test-to-train ratio of 0.25 (80% train-validation, and 20% test).

The parameter V_L is defined as the number of time-series data objects monitored for each iteration of the detection phase. The proposed novelty detection framework tunes detection thresholds with the selection of V_L to reduce detection results' sensitivity to the user's choice of V_L. Hence, a range of values for V_L are selected for both datasets to assess the impact of the tuning stage.

The target reliability index, β, is a number to show the probability of failure for a defined goal in reliability analysis. The convention in reliability analysis is to set $\beta \geq 3$ to ensure a small probability of failure. We chose $\beta = 3$ for both tests, and such a choice is recommended as the default for any implementation, unless some relaxation of the false positive rate is

desired, for example, to reduce the false-negative rate. This parameter is further explained in the reliability section of the methodology.

4.4.2 Training Phase

The GAN and 1-CG models are trained with the features extracted from the first T_L time-series data objects in the training phase. Specifically, the GAN model is trained on the high-dimensional features (*i.e.*, F_I), and the 1-CG model is trained on the low-dimensional features (*i.e.*, F_{II}). In this section, first, the feature extraction method is explained. Then, GAN's architecture and 1-CG model are described. Two novelty scores are defined based on the trained GAN and 1-CG outputs for further utilization in the tuning and detection phases.

Feature Extraction

Based on the potential difficulties caused by the *loss of information* and the *case-dependent feature extraction* in real-time SHM, we implement a simple feature extraction method that balances high- and low-dimensional features, as shown in Fig. 4.4. The extraction is based on the fast Fourier transform (FFT) of the input signals from N channels. The first feature (F_I) is high-dimensional ($N*D_L/2\times1$), defined from half-spectrum FFT magnitudes of the input signals with maximum values suppressed to ten. As discussed in Soleimani-Babakamali *et al.* [102], limiting the features to vary in a fixed range is beneficial for the GAN's training. The second feature (F_II) is a reduced F_I representation, made of quartiles of vibrational energy in each time-series data object, which is implemented in prior studies [30]. Since the method is designed to be general and avoid time-averaging with specified windows, a Periodogram power spectral density (PSD) estimate is employed. In this method, the FFT

magnitudes of a signal are normalized and squared to build the PSD estimates, which can extract vibrational energy quartiles. Both low- and high-dimensional features are utilized in the detection phase, and their effectiveness is compared.

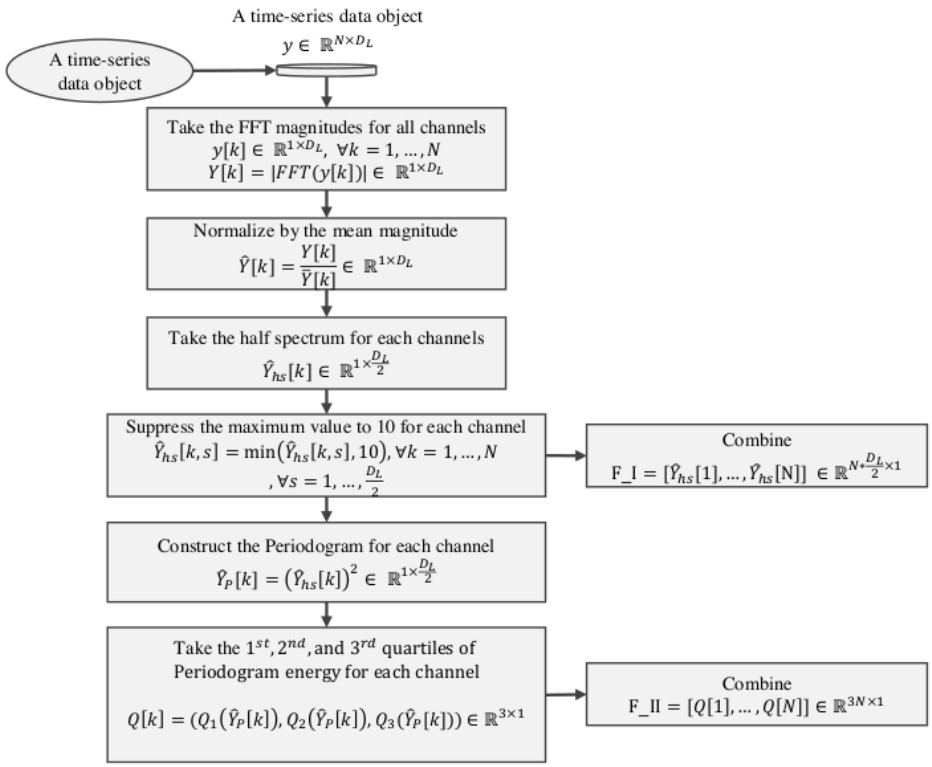


Figure 4.4: Feature extraction step, N is the number of data channels.

The GAN architecture

A GAN consists of two neural networks, a generator ($\mathcal{G}$) and a discriminator ($\mathcal{D}$). The GAN's training set, which is also called real data, is a group of data objects with no labels (*i.e.*, unsupervised) and distribution of P_{data}. As mentioned, the real data in this study is the time-series data objects' F_I features. $\mathcal{G}$ tries to mimic real data by sampling from a pre-defined probability density function (PDF), called the latent space (*i.e.*, P_{latent}). A

sample z is taken from P_{latent}, and through the neural network layers of $\mathcal{G}$, it takes the same dimension of real data (*i.e.*, $N * D_L/2 \times 1$). Real data is then labeled as "1", and $\mathcal{G}$-generated features are labeled as "0", and are called fake data. The ultimate goal of $\mathcal{G}$ is to generate fake data resembling the real data. At each iteration, a combination of real and fake data enters a binary classifier (*i.e.*, $\mathcal{D}$), whose goal is to accurately classify real data as 1 and fake data as 0. By assuming the last activation of $\mathcal{D}$ to be a sigmoid function, the training loss of $\mathcal{D}$ is defined as

$$\ell_{\mathcal{D}} = -\mathbb{E}_{x_{\text{F_I}} \sim P_{\text{data}}} \left[\log \mathcal{D}(x_{\text{F_I}})\right] - \mathbb{E}_{z \sim P_{\text{latent}}} \left[\log \left(1 - \mathcal{D}(\mathcal{G}(z))\right)\right], \tag{4.1}$$

where $x_{\text{F_I}}$ is a real data. This loss function penalizes $\mathcal{D}$ when fake data are classified as real or vice versa. Hence, $\mathcal{D}$ learns to classify input data as real or fake (i.e. a novelty) in an unsupervised manner. Meanwhile, $\mathcal{G}$ is trained with the backpropagation of its loss defined as

$$\ell_{\mathcal{G}} = -\mathbb{E}_{z \sim P_{\text{latent}}} \left[\log \mathcal{D}(\mathcal{G}(z))\right]. \tag{4.2}$$

This loss penalizes $\mathcal{G}$ if it is not able to generate "realistic" data objects. As a result, $\mathcal{G}$ learns to generate data objects that resemble the real data. In this study, $\mathcal{G}$ is used to generate data objects' features from the incoming stream of data, and simultaneously, $\mathcal{D}$ works as a binary classifier between baseline data (considered real) and novel data (considered fake) in the detection phase. The limitless generations from $\mathcal{G}$ are further used for the tuning phase.

The feed-forward neural network architectures of $\mathcal{D}$ and $\mathcal{G}$ are shown in Fig. 4.5 and Fig. 4.6, respectively. Since $\mathcal{D}$ acts as a binary classifier, the sigmoid function is set to be the last activation function in its architecture (Fig. 4.5). The latent space of $\mathcal{G}$ is defined as a normal distribution, with the latent dimension of 100. It is worth mentioning that in both datasets, $N * D_L/2$ value (*i.e.*, F_I dimension) is greater than the number of neurons in the

penultimate layer of $\mathcal{G}$, and users can change this layer's neurons to meet their problem.

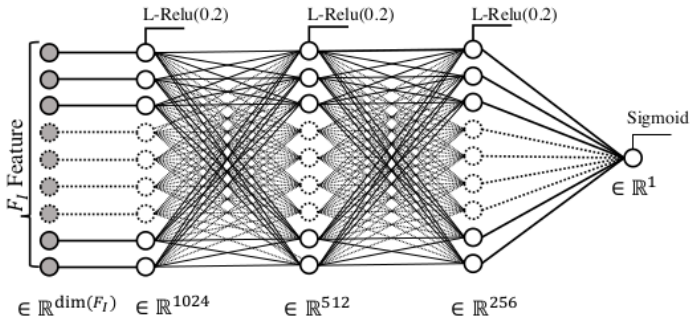


Figure 4.5: $\mathcal{D}$ architecture, L-ReLU(α) is Leaky RuLU activation with negative slope of α, and Dropout(p) is the dropout activation layer with a probability of p.

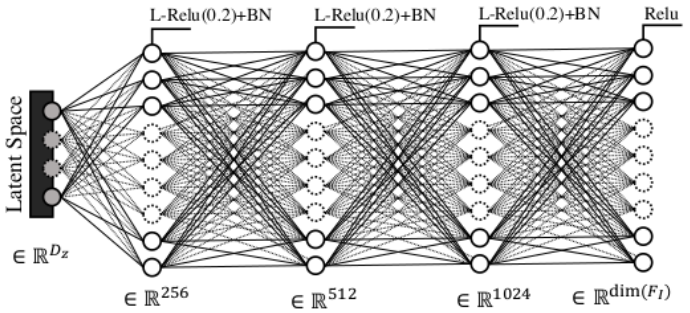


Figure 4.6: $\mathcal{G}$ architecture, L-ReLU(α) is Leaky RuLU activation with negative slope of α, and BN is the batch normalization layer.

The GAN training process is a zero-sum game, in which the gain for one player (*e.g.*, $\mathcal{D}$) is the loss for the other player (*i.e.*, $\mathcal{G}$). Such a game scheme finally reaches the Nash equilibrium [66] with the same losses for $\mathcal{D}$ and $\mathcal{G}$ ($\ell_{\mathcal{G}} = \ell_{\mathcal{D}}$). Equilibrium depicts a trained $\mathcal{G}$ which can mimic real data adequately by pushing $\mathcal{D}$ to randomly classify generated data objects as real or fake. In this study, and to propose a general method, the epoch for ending the GAN's training is set to be in the equilibrium zone of losses. During the GAN's training, equilibrium may fall after several epochs and then return, or the training starts to diverge from the starting point (*i.e.*, no equilibrium). Change of the employed optimizer's

parameters can aid in reaching an equilibrium. For this study, GAN's pieces of training are terminated after a continuous equilibrium or at the end of an equilibrium. The training batch size equals T_L (*i.e.*, the entire training set); hence, each epoch is equivalent to an iteration. The ADAM optimizer [49] is used for neural network training. The optimizer has three hyperparameters: learning rate, β_1, and β_2. In the introductory study of ADAM [49], the learning rate of 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ were suggested. In this study, and for employing ADAM for GAN models, $\beta_1 = 0.5$, and the learning rate of 0.0002 is decided [62]. However, the target epoch is where the losses are at equilibrium, and these parameters may need to be changed accordingly to reach that state.

The output of $\mathcal{G}$ is an $N * D_L/2 \times 1$ vector (*i.e.*, F_I dimension), which mimics the T_L training objects' behavior. The output of $\mathcal{D}$ for a F_I input feature, $\mathcal{D}(x_{\text{F_I}})$, is the output of a sigmoid activation layer. The higher this value, the higher the confidence of $\mathcal{D}$ that a new test data object belongs to GAN's training set. From this, we define the GAN's $\mathcal{D}$-based score, S_{GAN} for a test data object's F_I feature, as

$$S_{GAN}(x_{\text{F_I}}) = -\log_{10}(\mathcal{D}(x_{\text{F_I}})). \tag{4.3}$$

The output of $\mathcal{D}$ is a probability and is between 0 and 1, in which the probability of 1 indicates the *least* likelihood of having a novel data object. Taking the negative logarithm of that probability in S_{GAN} permits to work with higher real numbers and also reverses the behavior; thus, higher scores indicate a novelty (*i.e.*, $-\log(0) = \infty$).

1-CG architecture

The 1-CG model is a joint Gaussian distribution, which is aimed to fit the T_L data objects' F_II features. The 1-CG model's probability density function for a given $x_{\text{F_II}}$ is defined as

$$p(x_{\text{F_II}};\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{3N}{2}}|\Sigma|^{\frac{1}{2}}} exp(-\frac{1}{2}(x_{\text{F_II}}-\mu)^T\Sigma^{-1}(x_{\text{F_II}}-\mu)), \tag{4.4}$$

in which $x_{\text{F_II}}$ has $3N$ dimensions herein. The mean values of $x_{\text{F_II}}$'s dimensions are stored in μ, and the covariance values between those dimensions are stored in Σ. μ and Σ parameters are fitted with the Expectation-Maximization algorithm to the training F_II features. By having the fitted 1-CG model, the likelihood of an unseen data object to be from this distribution is measurable with the likelihood function. The likelihood function for one data object to be observed is the same as the probability density function:

$$\mathcal{L}(\mu,\Sigma|x_{\text{F_II}}) = p(x_{\text{F_II}};\mu,\Sigma) \tag{4.5}$$

Consequently, S_{1-CG} score for a new test data object can be defined as

$$S_{1-CG}(x_{\text{F_II}}) = \frac{-\log\mathcal{L}(\mu,\Sigma|x_{\text{F_II}})}{\frac{1}{T_L}\sum_{x\in x_{\text{F_II}_T}}(-\log\mathcal{L}(\mu,\Sigma|x))}, \tag{4.6}$$

in which, from a density-based anomaly detection viewpoint, the ratio of the negative log-likelihood of a test F_II feature's to the mean negative log-likelihood of the 1-CG model's training set (*i.e.*, $x_{\text{F_II}_T}$) is used for this score.

4.4.3 Tuning and detection phases

The trained GAN and 1-CG models produce two different scores, namely S_{GAN}, and S_{1-CG}, both of which examine the chance of a test data object to be an outlier (*i.e.*, novelty). A pre-defined novelty detection system of limit-state functions is tuned to the user-defined parameters (*i.e.*, V_L) in the tuning phase. The novelty detection system and the proper methods for tuning it are explained in further subsections.

Novelty detection using a GAN and 1-CG ensemble

Generally, a novelty or failure criterion can be defined with a limit-state function (g) as

$$g(T, S) = T - S, \tag{4.7}$$

which can be viewed as the difference between Resistance (T) and Load (S). A limit-state function fails whenever S is greater than T. In an SHM novelty detection problem, the scores obtained from GAN and 1-CG models can be seen as S. Thresholds, seen as T, define the limit of S in the baseline case and signal a novelty whenever $g < 0$ ($S > T$). Since there are two scores, several limit-state functions can be defined. Those functions can be series (*i.e.*, a failure of one results in total failure) or parallel (*i.e.*, both must fail so that the system fails). Such combinations create a detection system.

In this study, there are two considerations to define the detection system. First, S_{GAN} is desired to be more important since it is based on a high-dimensional feature. Second, two different detectors are to be defined: one to contain a more aggressive S_{GAN}-based limit-state function (*i.e.*, Limit-State-III) and the other to be milder (*i.e.*, Limit-State-I). A S_{1-CG}-based limit-state function (*i.e.*, Limit-State-II) is then added to the detector with the aggressive S_{GAN}-based limit-state function to regulate its detection sensitivity. As a result, the novelty detection system is defined as shown in Fig. 4.7 with two detectors and three limit-state functions. In this system, the S_{GAN}-based score is in the majority (*i.e.*, two out of three limit-state functions) and is in an absolute stand (*i.e.*, series limit-state functions). In this detection system, T_1, T_2, and T_3 are the detection thresholds. The system detects a novelty if Detector-I is failed (*i.e.*, S is over T in Limit-State-I) or Detector-II fails (*i.e.*, Limit-State-II and III to fail simultaneously). T_3 must be selected to be lower than T_1 to have the more aggressive S_{GAN} detector on Limit-State-III.

In general, S_{GAN} and S_{1-CG} are not scalars but rather a $V_L \times 1$ vector of scores for each detection window of V_L (Fig. 4.3). In this more general case, percentiles can be used to define the scalar load S. We use the median value (50th percentile) of the low-dimensional 1-CG scores ($P_{50}(S_{1-CG})$). However, since high-dimensional features tend to be noisier than low-dimensional data, the 20th percentile is selected for the GAN scores ($P_{20}(S_{GAN})$).

At this point, thresholds T_1, T_2, and T_3 could be defined manually to create a functional novelty detection system. However, a key aspect of our approach is that it leverages reliability analysis and the $\mathcal{G}$-generated features to robustly tune the thresholds based on the user's choice of detection parameters, explained in the next section.

Reliability-based threshold tuning

Since this is an unsupervised approach, except for the T_L time-series data objects, no more data is available to tune the thresholds. However, T_L training data objects data can be greatly augmented through the use of GAN-generated features, and thresholds are tuned by defining an analogous system shown in Fig. 4.8.

The central precept of the framework's tuning phase is that thresholds should be defined such that no detection to occurs for the GAN-generated features in the analogous system. This approach assumes that the variation in the GAN-generated features is similar to or greater than the variation of the unseen data objects' features belonging to the same class as the training set; thus, representing extreme cases. In the analogous system, the score percentiles are complementary (*i.e.*, P_{80} and P_{20}) to the main system since the main system is defined for detecting novelties; whereas, the analogous system is established to not detect novelties in GAN-generated features. However, as mentioned, for having a more aggressive Limit-State-III compared to Limit-State-I, T_3 is selected to be smaller (*i.e.*, P_{50} instead of

P_{80}). The $\mathcal{G}$-generated features are initiated from a standard Gaussian distribution and are random variables. Hence, for a system of limit-state functions with random variables, applying reliability-based analysis techniques to tune the thresholds to the desired reliability index from the $\mathcal{G}$-generated features is possible. As mentioned, the reliability analysis must be carried out on the analogous system shown in Fig. 4.8. A reliability index (β) indicates the failure probability of limit-state function(s). A higher β means a lower failure probability. Since the analogous system's failure should have a very low probability, a high β value is desirable. In this study, β of 3 is chosen as a suitable value, corresponding to a 0.0013 probability of failure for the analogous novelty detection system. Additional values of β are tested once for a constant V_L to demonstrate its effect on novelty detection in the results section. With a known target β for a single or a system of limit-state functions, the Reliability (*i.e.*, R) can be defined as

$$R = 1 - \Phi(-\beta), \tag{4.8}$$

in which Φ represents a standard normal distribution function. The notion of R can be used to define different limit-state functions' β based on the target β for the whole system. By assuming the novelty detection system's limit-state functions to be uncorrelated (for simplicity), the system's reliability can be summarized as

$$R_{system} = R_{EL-I}(R_{EL-II} + R_{EL-III} - R_{EL-II}R_{EL-III}). \tag{4.9}$$

Furthermore, the system can be tuned by assigning the same reliability (R_E) to all three limit-state functions, resulting in

$$R_{system} = 2{R_E}^2 - {R_E}^3. \tag{4.10}$$

By solving Eq. 4.10 for the β of 3, the resultant β for each limit-state function is 3.012.

For each limit-state function, Monte Carlo histogram sampling (MCHS) can be applied. In MCHS, S of the analogous novelty detection system limit-state functions (*i.e.*, percentiles of S_{GAN}, and S_{1-CG}) are sampled, and histograms of their values are constructed. Based on the acquired histogram, T (*i.e.*, thresholds) can be selected to satisfy the reliability index criterion. In this study, and based on the target reliability index, 5000 iterations are used in the MCHS; however, a higher β would require more iterations.

Dealing with misleading GAN generations

$\mathcal{G}$ generates features by sampling a random noise vector from a probabilistic distribution, which is a standard Gaussian distribution (SGD) herein. Numerous neurons, weights, and biases transform the random noise into realistic fake features. There is always the possibility that $\mathcal{G}$ is not trained well for all parts of that SGD, resulting in misleading generations. This phenomenon is shown and discussed in Soleimani-Babakamali *et al.* [102], and Zareapoor *et al.* [116], in which improperly generated samples could even downgrade the classification accuracy. Defining features in a bounded manner to "Cap" the generated data objects was practiced in Soleimani-Babakamali *et al.* [102]. The classification of generated data objects to augment the minority class with the generated data objects classified as that class's member was employed in Zareapoor *et al.* [116] as another method to defend against improper generations. There should be a filter to perform the same task on the MCHS-generated histograms of Loads in this study.

This study assumes that the trained $\mathcal{G}$ may generate two groups of features: informative and the misleading ones [102, 116]. Hence, a method must separate these two distributions from each other so that only the informative generated data remains. To achieve this goal, a Gaussian Mixture model with two Gaussian distributions (2-CGMM) is determined. By fitting this model to the generated features, the members in the Gaussian distribution with

the lower mean (*e.g.*, $\mathcal{D}$ scores) are assumed to be informative. The informative generated features resemble their training set; hence, the cluster with the lower mean value of scores is identified as the informative cluster. The generated features in the second Gaussian distribution are assumed to be non-informative if the second Gaussian distribution passes a pre-defined limit. The limit to throw away the second distribution is that the ratio of the higher to the lower mean values of distributions surpasses a limit in an iterative setting. The iterative procedure is explained in Alg. 1. The algorithm is defined to converge fast while avoiding excess data removal.

Detection phase

The designed novelty detection system (Fig. 4.7) can be tuned for all user-defined parameters; however, the most influential parameter is V_L, as it dictates the length of the vector of time-series data objects to be monitored on each iteration. Hence, in this study, different values of V_L are used for evaluating how the reliability-based tuning phase lowers the detection sensitivity to its selection. The detection phase can be initiated with a tuned novelty detection system, trained GAN, and 1-CG. The detection phase can have a static baseline (*i.e.*, one class) or a dynamic baseline. The only difference between the static and dynamic baseline approaches (Fig. 4.3) is that in the static one, the thresholds are tuned only once for the normal class (*i.e.*, the initial data stream). Damages are detected against the normal condition, without the potential to have discrimination among different damage scenarios. However, the dynamic baseline approach re-tunes thresholds after a novelty is detected, understanding and preserving different damage scenarios' characteristics.

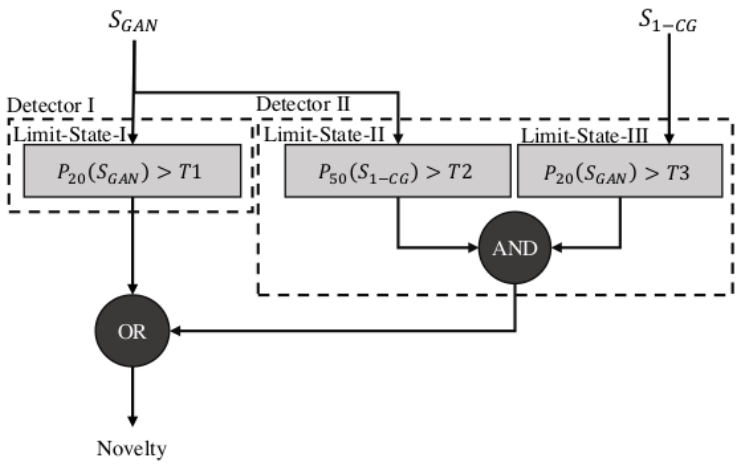


Figure 4.7: Novelty detection system.

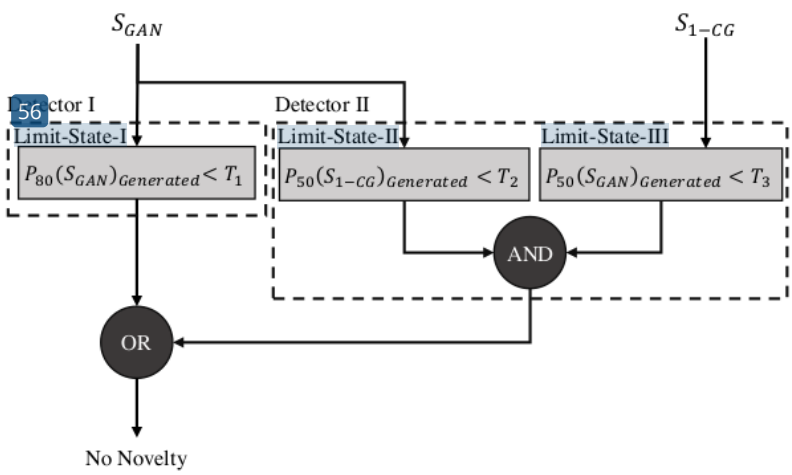


Figure 4.8: Analogous system.

Algorithm 1: 2-CGMM-based anomaly detection in score histograms.

Input: A Load histogram

Output: An outlier-removed Load histogram

```
i ← 1;
j ← 1;
while j==1 do
    Train a 2-class GMM on the Load histogram;
    if the second cluster mean value divided by the first cluster mean value is greater
      than i+1 then
        Delete the second cluster points;
        i ← i + 1;
    else
        j ← 0;
    end
end
```

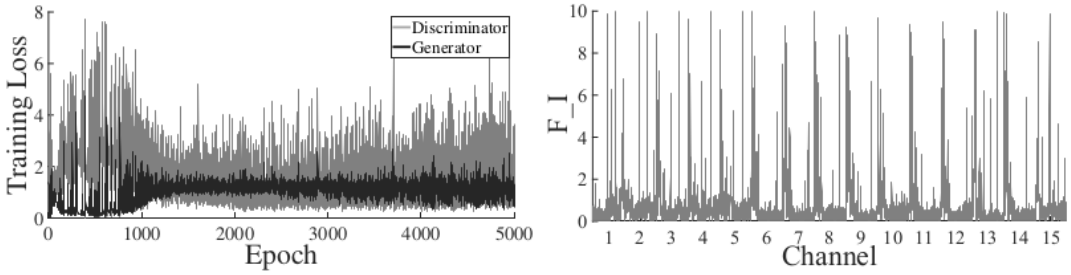
4.5 Yellow Frame results and discussion

This study aims to define a framework in which the novelty detection thresholds are tuned based on the user-defined parameters so that false alarms and novelty detection's accuracy become uncorrelated to the chosen parameters. For Yellow Frame, T_L is equal to one-hundred, and three values of ten, twenty, and forty are chosen for V_L, and the detection method is performed on the dataset. In what follows, results from each segment of the proposed unsupervised real-time SHM method (Fig. 4.3) are presented and discussed. Furthermore, and only for the case of $V_L = 20$, different β values are employed in the tuning

phase to evaluate its effect on the results.

4.5.1 GAN and 1-CG training, Yellow Frame

As discussed in section 4.4.2, the GAN's training can be terminated during an equilibrium (*i.e.*, $\ell_{\mathcal{D}} = \ell_{\mathcal{G}}$) or when the equilibrium breaks. The GAN's training loss and a generated F_I feature for class FDC1 are shown in Fig. 4.9a and Fig. 4.9b, respectively. Based on these results, the Yellow Frame training is concluded in 5000 epochs. The 1-CG model is also trained on the training set (*i.e.*, T_L data objects).



(a) (b)

Figure 4.9: Unsupervised real-time SHM framework results for FDC1 (a) GAN' training loss (b) a generated F_I feature.

4.5.2 Novelty detection system tuning for V_L

The MCHS of score percentiles (*i.e.*, Loads) for all three limit states of class FDC1 are shown in Fig. 4.10 (before applying the 2-CGMM outlier removal) and Fig. 4.11 (after applying the 2-CGMM). The variations in the scores are the desired fluctuations to have a robust detector, passing data fluctuations caused by factors, such as environmental changes or operational variations. MCHS results without outlier removal exhibit a considerable tail for $P_{50}(S_{1-CG})$ in Fig. 4.10b. Since the reliability analysis seeks extreme values (*i.e.*, high β), the

resulting detection threshold is large without the application of 2-CGMM when compared to Fig. 4.11b, resulting in few detections. The 2-CGMM algorithm clears those tails (*i.e.*, outliers). However, in the GAN score percentiles, the 2-CGMM implementation can result in no change in the Load histograms (Fig. 4.10a vs. Fig. 4.11a and Fig. 4.10c vs. Fig. 4.11c).

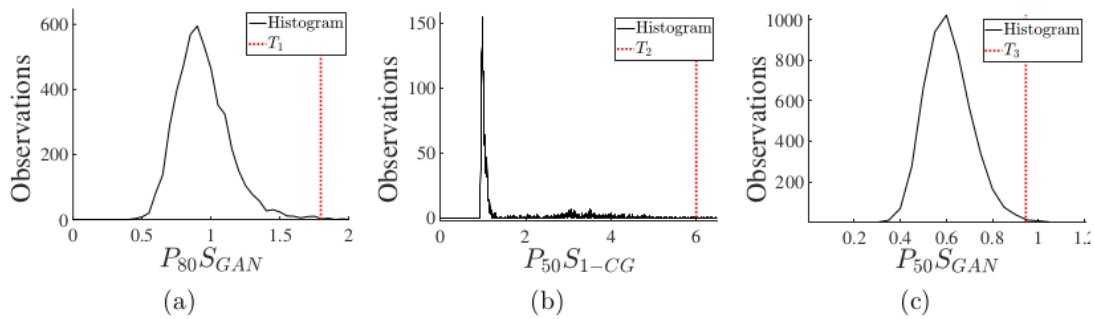


Figure 4.10: Load MCHS results of novelty detection system's limit-state functions (a) I (b) II (c) III.

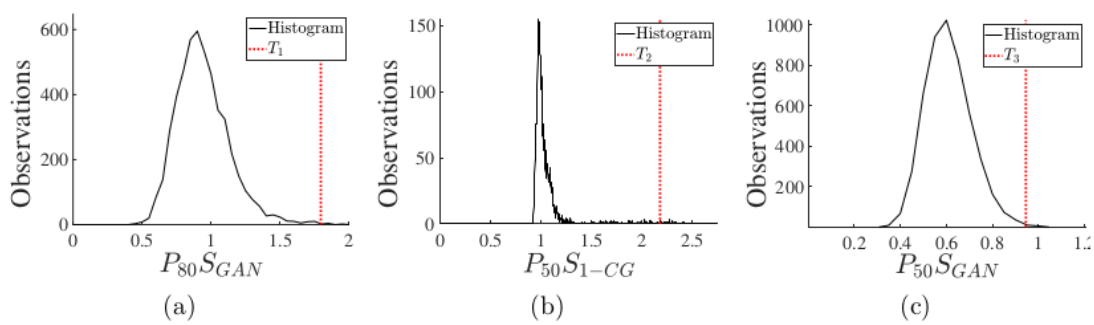


Figure 4.11: Load MCHS results (without outliers) of novelty detection system's limit-state functions (a) I (b) II (c) III.

4.5.3 Unsupervised real-time novelty detection results, Yellow Frame

With the determination of the novelty detection system's thresholds for the first class of data (*i.e.*, FDC1), V_L time-series data objects are analyzed using the novelty detection method each iteration. If a novelty is detected before the starting point of a new class of damage

minus V_L, the detection is labeled as a false alarm; otherwise, the detection can be on time or with delay, or novelties can be undetected. GAN and 1-CG training phases are performed in the dynamic baseline approach after a novelty is detected, and this cycle continues for the entire dataset. For the sake of enumerating the false alarms, if a false alarm is triggered, the analysis continues until the detection of a true novelty happens.

For the three different V_L values of 10, 20, and 40, the results of the unsupervised real-time SHM framework are shown in Fig. 4.12, Fig. 4.14, and Fig. 4.15, respectively. For $V_L = 10$, all damages are recognized without delay. There are three false alarms in 1818 vectors of time-series data objects. Hence, the false alarm ratio is equal to 0.165%. For $V_L = 20$, all damages are recognized without delay with one false alarm in 909 vectors of time-series data objects, equivalent to a 0.11% false alarm ratio. For $V_L = 40$, all damages are recognized without delay. There is one false alarm in 404 vectors of time-series data objects, equivalent to a 0.24% false alarm ratio.

The S_{1-CG} scores are noisy (Figs. 4.12, 4.14, and 4.15), as there are plenty of situations in which its detection threshold is exceeded. On the contrary, S_{GAN} is very sensitive to the presence of a novelty, mostly oscillating in a specific range and increasing sharply in the presence of a novelty. Such a trend indicates the potential of high-dimensional features for a robust novelty detection procedure. However, S_{1-CG} also plays an essential role in novelty detection as T_1 might be too high due to non-deleted outliers in Load histograms or the amount of alteration in the F_I's features of successive classes of data (*e.g.*, close damage scenarios). In such cases, S_{1-CG} can aid damage detection.

For $V_L = 10$ (*i.e.*, 1818 vectors of time-series data objects), the detection results for different β values are reported in Table 4.4. The false alarm ratios are expected to perceive the β variation's effect on the detection results in this table. It is worth mentioning that for higher β values, more simulations are required for the MCHS; hence, all MCHS are performed

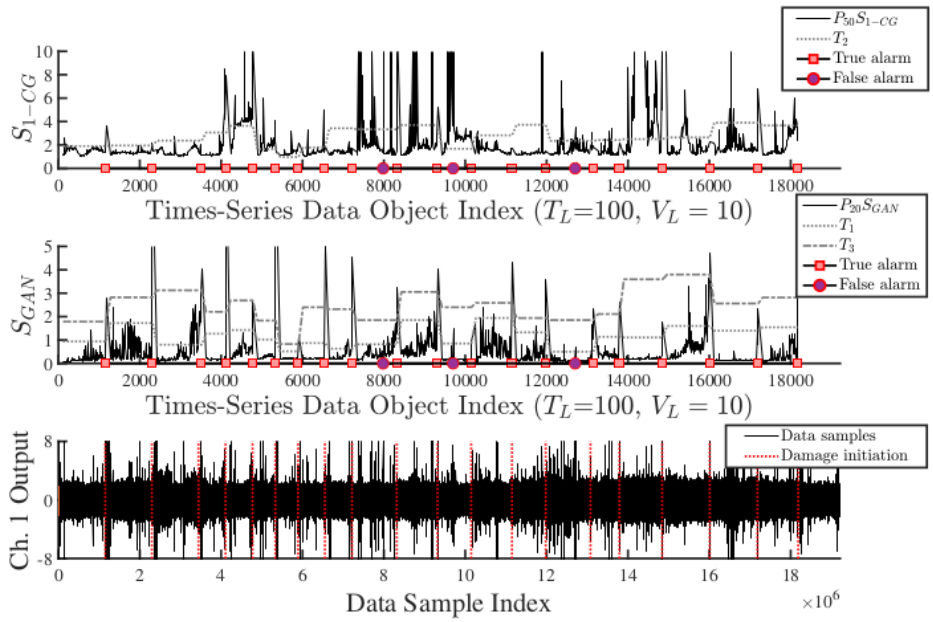


Figure 4.12: Dynamic baseline novelty detection results with V_L equal to ten, Yellow Frame.

with 125000 iterations for this section. Based on the main assumption in the tuning phase, β should be selected as high as possible, as the analogous system must not fail (*i.e.*, low probability of failure) for the GAN-generated features.

Based on Table. 4.4, low β values result in a higher false alarm ratio while higher β reduces that ratio. The detection framework's sensitivity to novelties is the most important parameter in the β selection. Increasing β reduces the false alarms but at the same time increases the detection thresholds, which in turn makes the novelty detection harder. The proposed high-dimensional features, and specifically the GAN's $\mathcal{D}$, are sensitive to novel data. As an example, in Fig. 4.14, the FDC2's normal data has S_{GAN} of around one, while at the point of novelty detection, it has a S_{GAN} of about six. S_{GAN} is a logarithmic score, which means that the $\mathcal{D}$'s score (*i.e.*, probability) for a novel data object is around million times smaller than a normal data object's score. This characteristic enables selecting a high β, ensuring

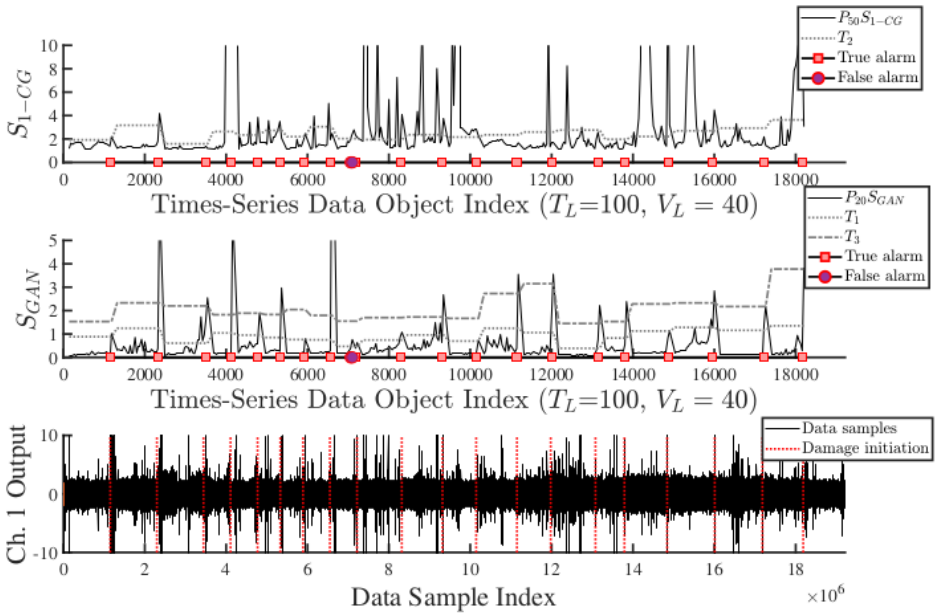


Figure 4.13: Dynamic baseline novelty detection results with V_L equal to forty, Yellow Frame.

less false alarm ratio while maintaining the novelty detection power. There are instances of delay in some data classes with high β values (Table 4.4). Due to the large thresholds caused by high β values, the detection may not occur at the point of the damage initiation; however, with some additional iterations and the fluctuation of scores, the damage is finally detected.

Table 4.4: Detection summaries for Yellow Frame with $V_L = 10$, and different β values.

β	False alarms	False alarm ratio (%)	Novelty detection results
1	90	4.95	All detected
2	20	1.10	All detected
3	3	0.16	All detected
3.5	3	0.16	All detected
4	3	0.16	All detected, Delay in FDC16
5	2	0.11	All detected, Delay in FDC16 and FDC7

The novelty detection is also applied in the format of static baseline, with $V_L = 20$. The

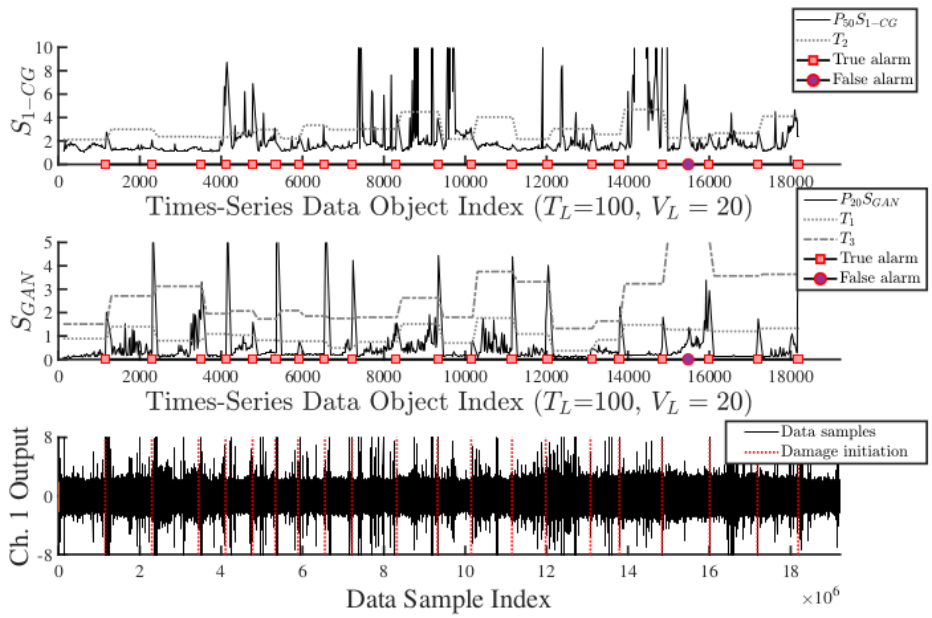


Figure 4.14: Dynamic baseline novelty detection results with V_L equal to twenty, Yellow Frame.

GAN and 1-CG models are trained only for the normal state, and the framework is used to detect all damage classes as a single novelty class. The result is shown in Fig. 4.16. Based on this figure, since only one true alarm at the beginning of each damage scenario is enough for that class to be considered detected without delay, all damages are identified with no false alarms. Although both static and dynamic baseline methods performed well, the dynamic baseline is of the main interest in this study as having a static number of classes in an unsupervised SHM is a drawback. However, in cases where binary classification is the main concern, the static baseline approach can be used considering the Yellow Frame datasets' reliable detection results with 21 data classes.

Receiver operating characteristic curve (*i.e.*, ROC curve) can be drawn for binary classifiers to assess their performance. Since the detections are based on baselines (*i.e.*, thresholds),

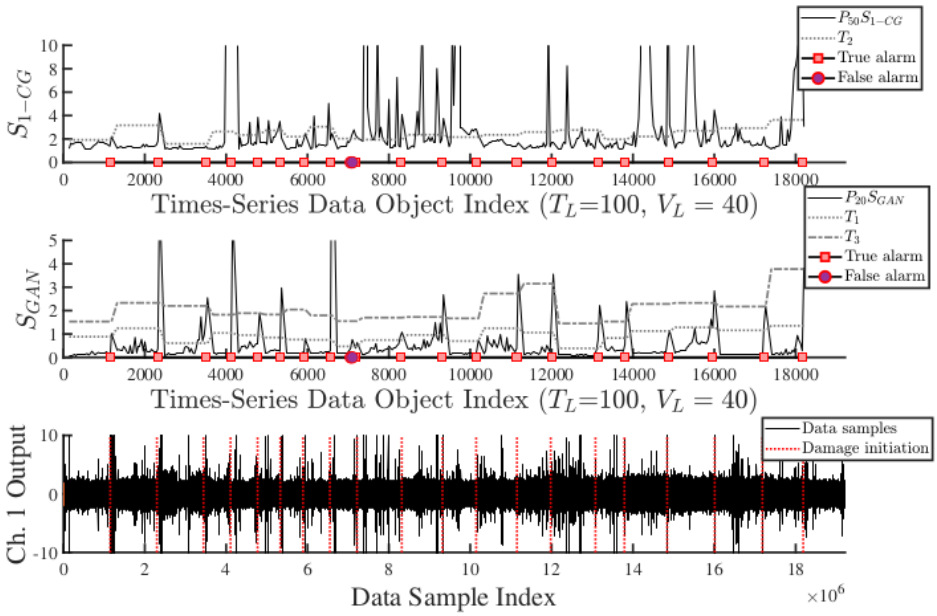


Figure 4.15: Dynamic baseline novelty detection results with V_L equal to forty, Yellow Frame.

ROC curves can be represented by altering the baseline's values. The proposed method can have both static and dynamic baseline strategies. Since all classifiers at different classes are tuned in the same way, and to assess the classifiers' performance in the presence of all damage classes, the ROC curve for Z24 and Yellow Frame datasets is drawn for their static baseline detection results. The area under ROC curves (*i.e.*, AUC), and its closeness to 1, can be used to evaluate the binary classifier's performance. This curve for the static baseline detection results of Yellow Frame is shown in Fig. 4.17. According to the figure, an AUC of 0.992 is obtained, which shows that the classifier maintains a high detection rate in low false alarm regimes. This ROC curve results from a baseline created from 100 data objects (*i.e.*, expanded by GAN) in a dataset with 20 different damage classes, which is notable. This performance is just for the static baseline approach of this method, and with the dynamic baseline process, novelties are also distinguishable in an unsupervised setting.

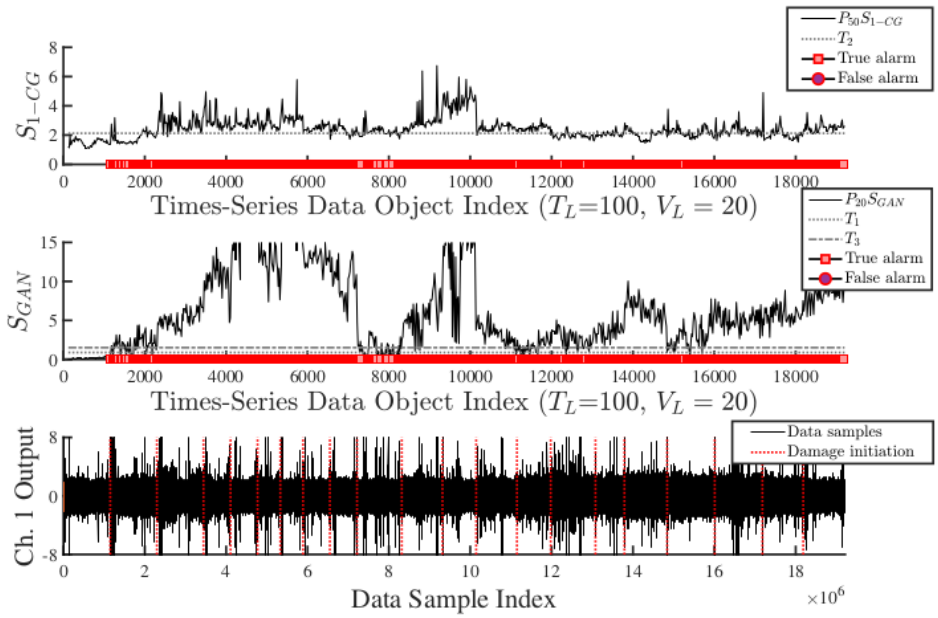


Figure 4.16: Static baseline novelty detection results with V_L equal to twenty, Yellow Frame. (in color)

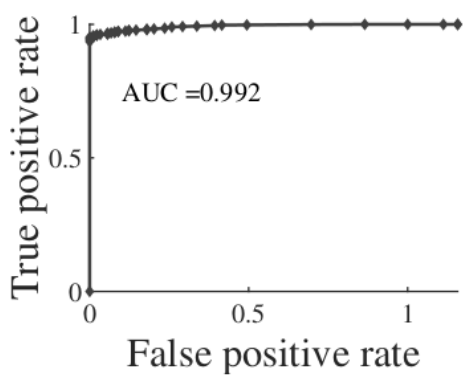


Figure 4.17: ROC curve of static baseline detection; Yellow Frame.

4.5.4 Literature comparison; Yellow Frame

Novelty detection results for Yellow Frame are compared to Rhará de Almeida Cardoso *et al.* [30] to highlight the novelties of the proposed method. The same damage type (*i.e.*, brace

removal) is practiced in both studies, with only seven damage classes in Rharã de Almeida Cardoso *et al.* [30]. The employed method in both studies is real-time and can distinguish between different novelties. The detection window length employed in Rharã de Almeida Cardoso *et al.* [30] is transformed to be in the format of utilized V_L in this study. The comparison is reported in Table 4.5 in the format of true and false alarm ratios. According to this table, in terms of both true and false alarm ratios, the method presented in this study is more accurate compared to Rharã de Almeida Cardoso *et al.* [30]. Moreover, the reliability-based tuning technique results in virtually the same true and false alarm ratios with different values of V_L. In contrast, in Rharã de Almeida Cardoso *et al.* [30], a change in V_L results in the discussed trade-off between detection power and the number of false alarms.

Table 4.5: Novelty detection comparison; Yellow Frame

Metric	This Study			Rharã de Almeida Cardoso *et al.* [30]		
	$V_L = 10$	$V_L = 20$	$V_L = 40$	$V_L = 5$	$V_L = 15$	$V_L = 50$
$TAR(\%)^*$	100	100	100	60.0	40.0	40.0
$FAR(\%)^{**}$	0.16	0.11	0.24	1.80	1.80	0.00

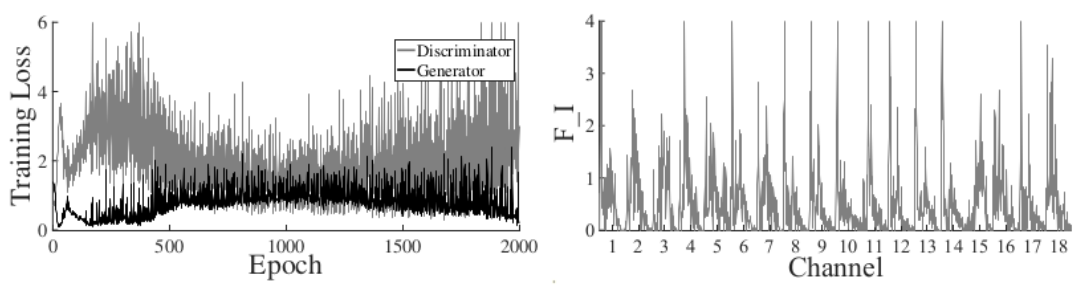
* True alarm ratio
** False alarm ratio

4.6 Z24 Bridge results and discussion

The Z24 Bridge dataset is utilized to evaluate the proposed method's generalizability to bridges. For this dataset, and to show the reliability-based unsupervised real-time SHM approach's capability of tuning the detection thresholds to avoid detection sensitivity to the user-defined parameters, three values of five, ten, and twenty-five are chosen for V_L. T_L is also equal to seventy-five. In what follows, results from each portion of the proposed unsupervised real-time SHM method (Fig. 4.3) are presented and discussed.

4.6.1 GAN and 1-CG training, Z24 Bridge

The Z24 data objects' F_I features have a dimension of 1800×1. As mentioned, this dimension is less than the Yellow Frame F_I features dimension (*i.e.*, 7500×1); hence, the GAN's training is concluded in 2000 epochs. In that epoch, the equilibrium breaks, and the analysis is stopped. The F_II feature for this dataset has the dimension of 54×1 and is extracted to train the 1-CG models. The GAN's training loss and a sample generated F_I feature for the BDC4 are shown in Fig. 4.18a and Fig. 4.18b, respectively. The 1-CG models are also trained with the training set to be used for tuning the detection thresholds.



(a) (b)

Figure 4.18: Unsupervised real-time SHM framework results for BDC1 (a) the GAN' training loss (b) a generated F_I feature.

4.6.2 Novelty detection system's tuning for V_L

The MCHS of Load for different novelty detection systems' limit states is performed to define the detection thresholds. All three Load histograms before and after the application of 2-CGMM (*i.e.*, outlier-detector) for BDC2 are shown in Fig. 4.18, and Fig. 4.19, respectively. The Z24 limit-state functions' Load histograms tend to be much noisier than the Yellow Frame histograms. Without the application of 2-CGMM (Alg. 1), the thresholds would be high, which increases the chance of missing novelties. Unlike the Yellow Frame case, all three

histograms have changed by applying 2-CGMM for this data class.

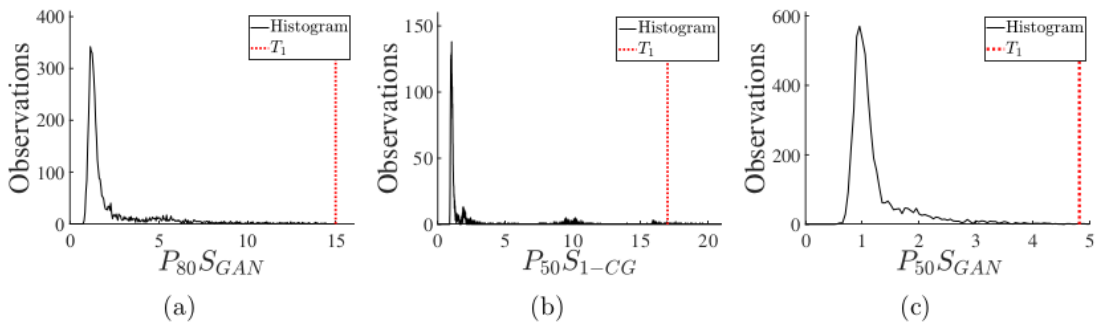


Figure 4.19: Load MCHS results of novelty detection system's limit-state functions (a) I (b) II (c) III.

4.6.3 Unsupervised real-time novelty detection results, Z24 Bridge

In what follows, results from each segment of the unsupervised real-time SHM (Fig. 4.3) are presented and discussed. Although Z24 has fourteen main damage classes, some are more severe versions of a preceding damage scenario. Following the same procedure as in the Yellow Frame case, a novelty is labeled as a false alarm if initiated before starting a new class of damage minus V_L. Both dynamic and static baseline approaches (Fig. 4.3) are applied, and results are discussed. The dynamic baseline detection results are shown in Fig. 4.20, Fig. 4.21, and Fig. 4.22 for V_L of 5, 10, and 25, respectively. The static baseline approach is also applied to the dataset with the results shown for $V_L = 10$ in Fig. 4.23.

With different V_L values, all seven distinct damage scenarios are detected in the dynamic baseline approach results. PL_{20} is distinguished from Ref_2 (*i.e.*, PL damage detection), T_F from PL_{95}, SC_{12} from T_F, LS from SC_{24}, F_{CH} from LS, F_{2A} from F_{CH}, and TR_2 from F_{4A}. All these damages are identified with their lowest intensity if they have several intensities. These seven main classes of damage are on different scales, global or local; hence, their detection shows the proposed framework's versatility in damage detection.

The results indicate that for some successive damage scenarios with different intensities, detection does not take place. This phenomenon is reasonable, as different tendon ruptures may change a specific part of the FFT of sensor signals, making it hard for GAN to capture the difference. However, this is not a shortcoming as all damage classes are identified. All different intensities are also identified from the initial undamaged class in the static baseline approach that can ensure (in this study) that the method can detect all damage classes, even with the same origin and different intensities from the normal state of the structure. Furthermore, some successive different damage-intensity categories are captured, such as the 40mm and 80mm from 20mm and 40mm of the pier lowering.

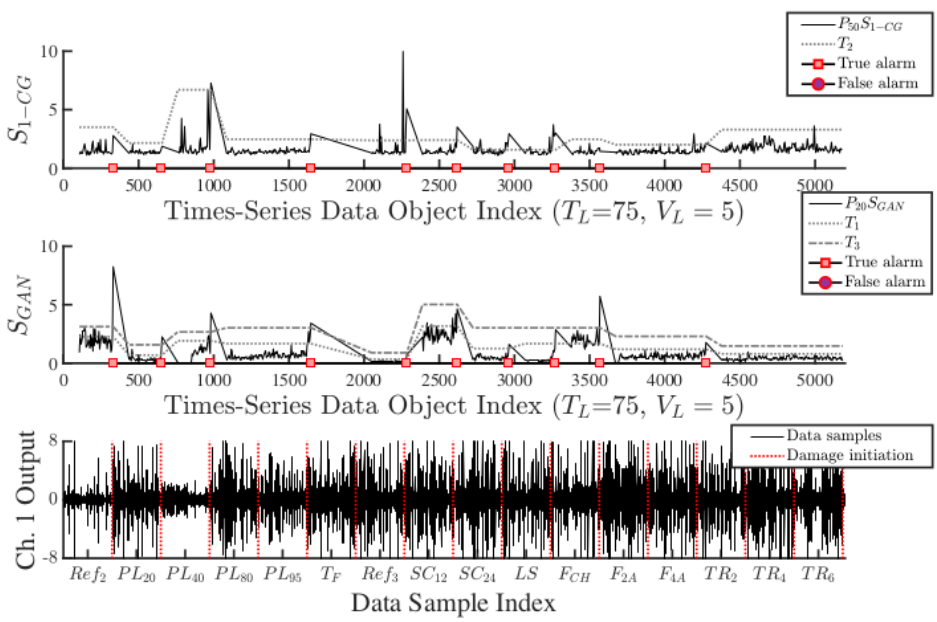


Figure 4.20: Dynamic baseline novelty detection results with V_L equal to five, Z24.

The effect of different β values on the detection outcome of Z24 Bridge with $V_L = 10$ is reported in Table 4.6. Based on this table and Table 4.4, two observations can be made. First, small β values result in a large number of false alarms, as they reduce thresholds.

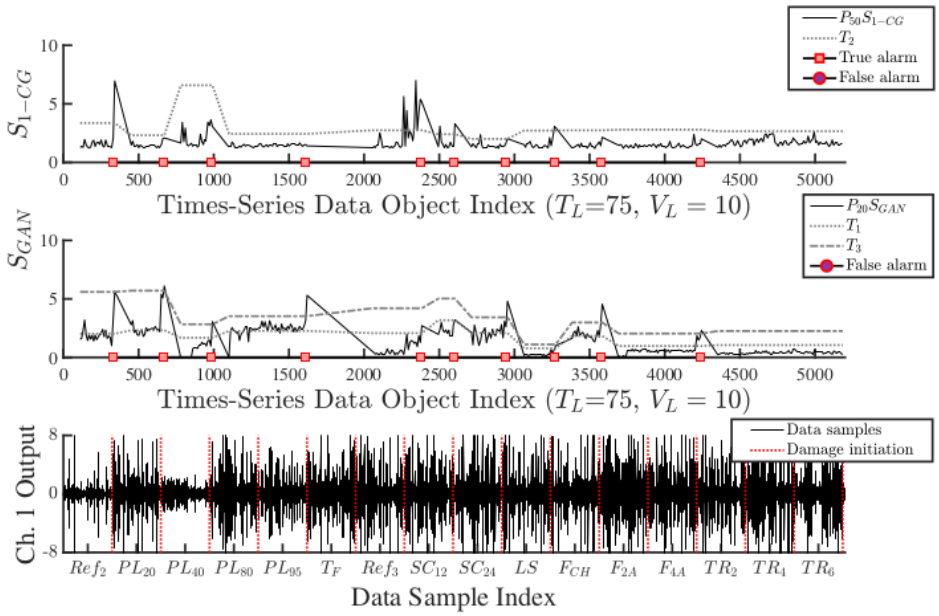


Figure 4.21: Dynamic baseline novelty detection results with V_L equal to ten, Z24.

However, this is simply for illustration purposes, as a low β is generally not recommended. Recall that the central precept of this study is that the analogous detection system must not fail for GAN-generated features. The effect of high β values is also observable in the table (as well as Table 4.4). Even the $\beta = 5$ does not reduce the true alarms in both datasets since the 2-CGMM outlier removal truncates the histograms. Hence, numerically, even the infinite β cannot pass the truncation points. As a result, there is a higher threshold limit, and increasing an already high β does not change the detection thresholds dramatically. There are standards to suggest values of β for structures, such as ISO 2394: 2015 [46]. Based on such codes, different β values from around 2 to 5 can be selected based on the possible safety requirements and other purposes. Due to the robustness of the proposed framework to high β values, a value of 3 or 3.5 is the recommended selection.

The ROC curve of the static baseline detection results for Z24 Bridge is constructed similarly

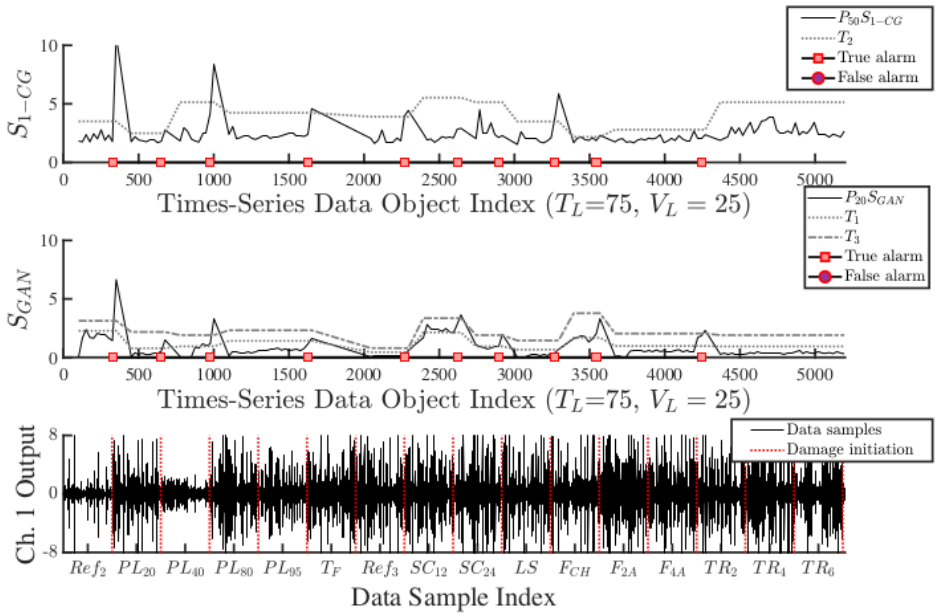


Figure 4.22: Dynamic baseline novelty detection results with V_L equal to twenty-five, Z24.

Table 4.6: Detection summaries for Z24 Bridge with $V_L = 10$, and different β values.

β	False alarms	False alarm ratio (%)	Novelty detection results
1	81	22.01	All detected*
2	6	1.63	All detected
3	2	0.54	All detected
3.5	0	0.00	All detected
4	0	0.00	All detected, Delay in T_F and TR_2
5	0	0.00	All detected, Delay in T_F and TR_2

* Different damage types, and not different severities

to Yellow Frame. This curve is shown in Fig. 4.24. AUC of 0.996 is obtained for this classifier, which shows the high detection rate of the classifier in low false alarm regimes. Such a high AUC is obtained with a baseline generated from only 75 samples. This performance is for the static baseline approach, and with the dynamic baseline method, different damages are also distinguishable in an unsupervised setting.

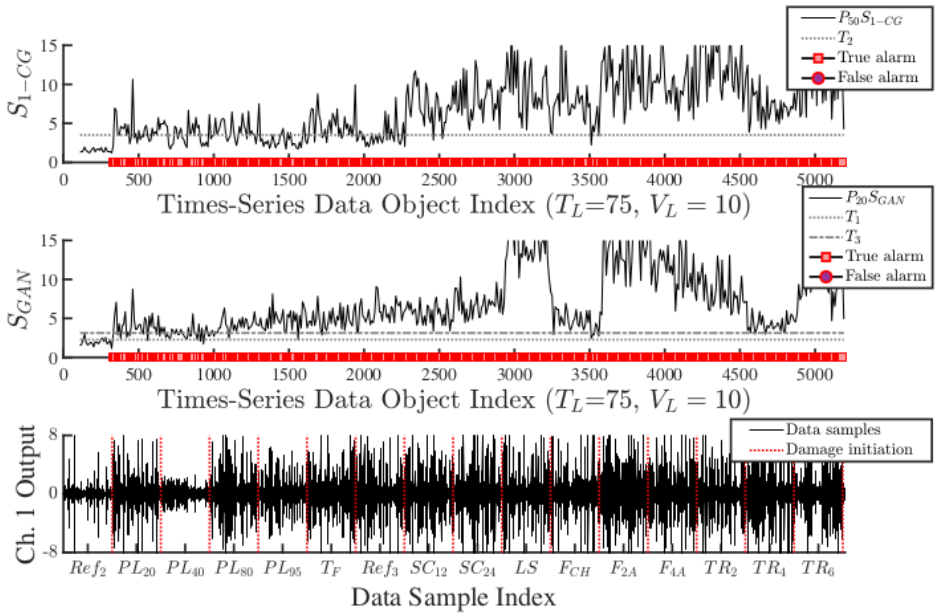


Figure 4.23: Static baseline novelty detection results with V_L equal to ten, Z24.

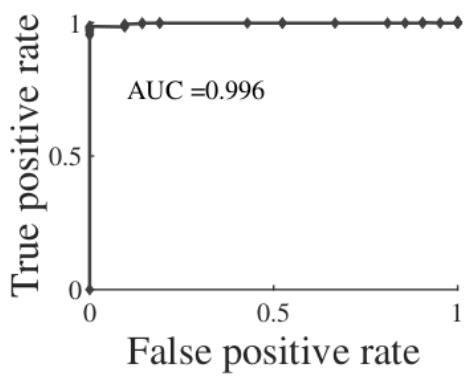


Figure 4.24: ROC curve of static baseline detection; Z24 Bridge.

4.6.4 Results comparison; Z24

As mentioned, no studies performed unsupervised real-time SHM with a dynamic-class novelty detection on Z24 Bridge. Results are compared to one of the most recent studies on Z24

Bridge with binary classification [95], which had the first four fundamental frequencies as features. A regression model was implemented on those frequencies to define baselines, and results were reported for different detection windows. For example, a 3-window detection indicates that three consecutive detection objects must exceed baselines to flag a novelty. Since the binary classification results are to be compared, the static baseline results are used (Fig. 4.23). In this case, a total of 4900-damaged data objects are present. With V_L of 10, there are 490 detection cases, from which 20 of them are not detected. Hence, the damaged-cases detection ratio is 95.9%. No undamaged data object is also classified as damaged, resulting in an undamaged-case detection of 100%. The same static baseline approach is performed for the two other V_L values for the comparison. The comparison is reported in Table 4.7. Based on this table, the proposed method has better detection performance, achieved using the first 75 data objects (*i.e.*, 75 seconds) from the data stream. In contrast, in the other study, the Z24 long-term monitoring dataset was employed to define a baseline (one year's worth of data objects). Hence, at least for this dataset, GAN-generated features can construct a baseline in a few minutes, which performs even better than a baseline established from a year of observations. Furthermore, only the static baseline results of the proposed methodology are compared, while the main potential of the proposed framework is to identify different damages and distinguish between them with the dynamic baseline method.

Table 4.7: Novelty detection comparison; Z24 Bridge

Detection (%)	This study			Wah *et al.* [95]		
	$V_L = 5$	$V_L = 10$	$V_L = 25$	1-window	2-window	3-window
Damaged cases	96.1	95.9	96.3	95.9	94.4	92.9
Undamaged cases	100	100	100	95.6	99.3	100

4.7 Computational complexity analysis

Since the analysis is real-time, computational times for different parts of the proposed framework are important. The whole analysis procedure is implemented in upgraded Google Colaboratory with an NVIDIA Tesla T4 GPU of 25.46 GB. The deep neural network architectures are implemented with Pytorch [69]. Matrix-based operations, such as Monte Carlo histogram sampling, are done with Numpy [42]. The heaviest computational portions of the framework are the neural networks' training and the Monte Carlo Histogram sampling. Since GANs are 1D, the analysis is relatively fast. For Yellow Frame with F_I's dimension of 7500×1, 5000 epochs are passed in 52 seconds. For Z24 Bridge, with 2500 epochs with a less dimension of F_I (*i.e.*, 1800×1), the analysis ends in 19 seconds. The mean duration of each data class is 9 and 6 minutes for Yellow Frame and Z24 Bridge, respectively. Consequently, the networks' training time is insignificant even for the studied time-limited datasets. There is more room for analysis time in a real setting since adequate damage detection can happen in the order of minutes or hours.

Regarding the Monte Carlo sampling, n F_I features are to be generated. In the case of T_1 and T_3 generations, the computation time is of relatively low complexity ($O(n)$) if we assume a constant generation time of $\mathcal{G}$ and constant classification time of $\mathcal{D}$ for each feature. However, the most computationally expensive task is T_2, since the quartiles of energy from each n generated data object must first be calculated. The analysis can be as long as several minutes for tuning T_2 for each novelty class for Yellow Frame with $n = 5000$. However, by exploiting the specific matrix-based syntaxes in Numpy and vectorizing rather than looping, T_2 computation time for each data class drops to about 5 seconds for Yellow Frame and far less for Z24. Hence, in a real-time setting, exploiting proper Numpy syntaxes and vectorizing the computation is essential.

4.8 Future work

The proposed method is a novel framework with many possible areas for improvement. Various network architectures for $\mathcal{G}$ and $\mathcal{D}$ can be investigated, with convolutional or RNN layers. The novelty detection system also has room for further research, as with the ensemble of GAN and 1-CG, different and more suitable systems can be employed. Furthermore, all GAN pieces of training are done without any supervision and with a fixed number of epochs. GAN has no specific concluding training epoch; hence, different training epochs (*i.e.*, training loss diagram patterns) can be checked for possible shortcomings or improvements in the results. Even if the thresholds are made robust with the variations in the GAN-generated data instances, different environmental or operational circumstances might trigger false alarms. The proposed model has the capability of having a dynamic-class novelty detection; hence, the prior trained GANs can be used to remember their associated novelties. If that capability is added, then in the case of such novel events, the model can recognize that class to avoid false alarm triggers. This scheme enables Continual Learning for the framework.

4.9 Conclusion

Investigating unsupervised, data-driven methods as the most suitable approach to perform real-time structural health monitoring has received much attention in the last decade. There are many improvements in the field, and this study simultaneously addresses four of the main obstacles which appear in some current approaches. The obstacles are loss of information from dimensionality reduction, case-dependency of feature extraction steps, lack of dynamic-class novelty detection, and the sensitivity to user-defined detection parameters on the nov-

elty detection results. Furthermore, this study strives to propose a framework for solving an unsupervised real-time SHM problem without prior information as a step towards having a general unsupervised SHM methodology. Two low- and high-dimensional FFT-based features apply to any instrumented structure (*i.e.*, having sensors installed) are defined. A multi-ensemble of joint Gaussian distribution (1-CG) models for the low-dimensional feature and GAN for the high-dimensional feature is established. A novelty detection system of limit-state functions is defined with the detection thresholds as Resistance and scores from trained GAN and 1-CG models as Loads. Using trained GAN's Generator, which generates features from a standard Gaussian distribution, the novelty detection system's thresholds can be tuned to user-defined parameters. The Monte Carlo histogram sampling approach is utilized for the tuning. For demonstrating the proposed method's generalization capacity, two datasets of a frame and a bridge are used. The Yellow Frame dataset has twenty-one classes, while Z24 Bridge has fifteen data classes. Based on the results, the following conclusions are made.

The framework's application in a static baseline setting resulted in no false alarms on both Yellow Frame and Z24 datasets, and all twenty damage classes in Yellow Frame and fourteen damage classes in Z24 were identified. A dynamic baseline approach identifies different novelties from the normal state and other damage classes (i.e., dynamic-class novelty detection). The dynamic baseline approach to Yellow Frame resulted in detecting all twenty changes in the structural condition. The false alarms were three, one, and two for V_L of 10, 20, and 40, respectively. The resulting false-alarm ratios show that the reliability-based thresholds' tuning made the detection results insensitive to user-defined detection parameters (i.e., V_L). The dynamic baseline approach's for Z24 resulted in the detection of all seven distinct damage classes. In cases of different intensities of damage, such as 95mm and 80mm lowering of the pier or between different numbers of ruptured tendons, the method could not catch

the difference. The reason might be that different intensity from the same damage scenario affects the FFT of signals in the same way. Hence, the detection models cannot tell those classes' differences. However, detecting different damage intensities is not as important as alarming new damage or damage following an intact condition. All the results originated from a simple feature-extraction step, i.e., the half-spectrum FFT. The simple extraction method can be applied to new structures without any training phase. The combination of framework characteristics is a vital step towards having a network of monitored facilities at large scales as a part of the future of smart cities. Key contributions of the work are outlined in the Research Significance section.

Acknowledgement

The authors would like to thank Prof. Carlos Ventura, Alex Mendler, and Saeid Allahdadian for providing the Yellow Frame dataset and associated images, and Dr. Giacomo Bernagozzi for providing us with the modal information of Yellow Frame. The Structural Mechanics Section of KU Leuven is gratefully acknowledged for providing access to the Z24 Bridge dataset.

Chapter 5:

Towards a general unsupervised novelty detection framework in structural health monitoring

Mohammad Hesam Soleimani-Babakamali, Reza Sepasdar, Kourosh Nasrollahzadeh, Ismini Lourentzou, and Rodrigo Sarlo. Toward a general unsupervised novelty detection framework in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 2022. doi: https://doi.org/10.1111/mice.12812. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12812

Abstract

This study proposes an unsupervised, online structural health monitoring framework robust to the sensor configuration, *i.e.*, the number and placement of sensors. The proposed methodology leverages Generative Adversarial Networks (GAN). The GAN's discriminator network is the novelty detector, while its generator provides additional data to tune the detection threshold. GAN models are trained with the Fast Fourier Transform of structural accelerations as input, avoiding the need for any structure-specific feature extraction. Dense, Convolutional (CNN), and Long Short-term memory (LSTM) units are evaluated

as discriminators under different GAN training loss patterns, *i.e.*, the differences between discriminator and generator training losses. Results show that the LSTM-based discriminators and the suggested threshold tuning technique to be robust even in loss patterns with overfitted discriminators, a probable outcome of limited training sets. The framework is evaluated on two benchmark datasets. With only 100 seconds of training data, it achieved 95% novelty detection accuracy, distinguishing between different damage classes and identifying their resurgence under varying sensor configurations. Finally, the majority-vote-ensemble of discriminator-generator pairs at different training epochs is introduced to reduce false alarms, improve novelty detection accuracy and stability.

5.1 Introduction

Machine learning techniques were always among the resolutions of Civil and Infrastructure engineering inquiries, from the utilization of gradient boosting machines to automatically detect cracks on concrete surfaces [28] to the application of neural networks. For instance, one-hidden-layer neural networks were employed for the estimation of concrete with restricted deep Boltzmann machines [76] or to estimate strain of columns in high-rise buildings from wind loads [68]. With the emergence of deep learning (*i.e.*, neural networks with two or more hidden layers) in 2012, it propagated rapidly to other disciplines, such as Civil engineering. The range of problems it solves can be as ubiquitous as the construction of bridges' 3D point clouds [44], or post-disaster preliminary damage assessment from aerial imagery [27]. Deep learning associated with the increasing availability of high-tech, low-cost, smart wireless sensors (*e.g.*, accelerometers) [5] is a novel field of research to solve current obstacles in structural health monitoring (SHM). Currently, in SHM, the focus is on fitting SHM tools to real-life structures [9]. In particular, the field is moving beyond supervised learning and

model-based techniques to real-time, semi-supervised and unsupervised SHM [24, 75, 80], demanding no labeled data.

Despite the impressive progress of unsupervised SHM methods, one major challenge unsupervised methods face is the lack of *generalizability to new (unseen) damages and structures*. For instance, sensor configurations vary heavily among different structures, while future damages are typically uncertain. Yet, current unsupervised SHM novelty detection approaches rely on (i) structure-specific feature extraction and (ii) considerable prior information from structures while facing difficulties in online discrimination and recall of varying damage classes (*i.e.*, dynamic-class novelty detection). Structure-specific feature extraction techniques refers to approaches, such as the estimation of finite vibrational modes [30, 73, 95], the use of hand-crafted or statistical features [6, 8], as well as deploying deep dimensionality reduction (DR) models [47, 59, 75] for feature extraction. The problem with hand-crafted features is that they do not generalize well across different systems [3]. Despite the availability of modal analysis techniques, even for complex systems [7, 55], their execution requires thorough fine-tuning for different structures, the same with the DR-based feature extraction. Additionally, modal identification of structures is heavily dependent on the sensor configuration of structures, *i.e.*, the number and location of sensors. Even though there is a separate line of research tackling the optimal sensor placement (OSP) problem [26, 112, 118], the non-scalability of optimization strategies [60] and the need for detailed structural models still present salient challenges.

Generally, novelty detection measures the divergence from a baseline condition, for which prior data are required. The need for prior data hinders large-scale monitoring, for instance, in smart cities with numerous structures. Moreover, the lack of data is the main obstacle in achieving a dynamic-class novelty detection. In dynamic-class novelty detection, novelties are identified and differentiated from each other. Binary SHM frameworks (*i.e.*, normal

and abnormal states of structures) [34, 95, 110] are studied in detail in the literature to the extent of performing satisfactory even under intense environmental variations [87]. A limited number of unsupervised SHM studies distinguish among various types of damages [20, 30, 84]. However, detection results are sensitive to the selection of user-defined detection parameters [20, 30] mainly due to the lack of data instances from the identified novel classes to tune thresholds.

High dimensional, domain-specific features obtained through simple feature extraction methods (*e.g.*, Fast Fourier Transform of a signal) can address the structure-specific feature extraction challenges if paired with the right detection architecture. In supervised settings, high-dimensional features have been used with 1D convolutional neural networks (CNN) [3, 15], 2D CNN [115], or with Long Short-Term Memory units (LSTM), a variation of recurrent neural networks (RNN) [102]. Specific target functions are required to employ such deep architectures in unsupervised settings. Generative Adversarial Networks (GAN) can handle unsupervised learning scenarios, *i.e.*, pure representation learning settings, and at the same time provides data augmentation for scenarios with limited training data available. In the SHM field, GANs have been applied to various tasks, including but not limited to lost signal reconstruction [52], faulty sensor identification [61], data augmentation for minority classes [83, 92, 116] and novelty detection [92]. In this work, we explore the aforementioned qualities of GAN to perform online, unsupervised damage detection with high-dimensional Fast Fourier Transform (FFT) of signals as features.

GAN is known to perform well in the equilibrium of discriminator and generator losses, *i.e.*, Nash equilibrium [99]. However, the equilibrium is not anticipated in a dynamic-class novelty detection setting (*i.e.*, lack of data from identified classes). Limited training data often lead to discriminator overfitting [48]. Thus, different GAN training loss patterns are included in this study, unlike previous attempts [61, 92]. Moreover, none of these prior works

proposed a methodology to identify the re-occurrence of novelties. The ability to recall past novelties is crucial in SHM as noticing their resurgence can aid maintenance plans and alarm prioritization. Furthermore, as discussed, the sensor configuration can significantly affect vibrational-mode-based novelty detection. This study proposes a methodology for reducing the sensitivity to sensor configuration by employing two mechanisms: (1) using raw data that, unlike modes, can capture the full spectra of each sensor, and (2) deep neural networks to automatically extract patterns on those high-dimensional features. Contributions of our work can be summarized as follows:

(1) We propose an SHM novelty detection framework that incorporates high-dimensional features with robust novelty detection regarding different sensor configurations. The proposed framework can reduce costs, installation time and can diminish the need for sensor-planning optimizations. The proposed GAN-based methodology requires no prior information on structures and is less sensitive to the number of employed sensors, substantially moving towards general unsupervised SHM techniques.

(2) Unlike binary novelty detection procedures, the methodology can distinguish between different novel classes; a dynamic-class approach. Moreover, it identifies the past novelties' resurgence. Thus, it mimics a dynamic-class clustering scheme with raw and high dimensional features (no dimensionality reduction).

(3) We provide exploratory analysis to select the best architecture for the discriminator and the best training-loss pattern of GAN to extract the trained generator and discriminator from it while accounting accuracy and less vulnerability to false alarms. The assessment of different loss patterns aids users' implementation of this framework, as terminating the GAN training process at a discernible, right point is vital for the framework's performance.

5.2 Methodology

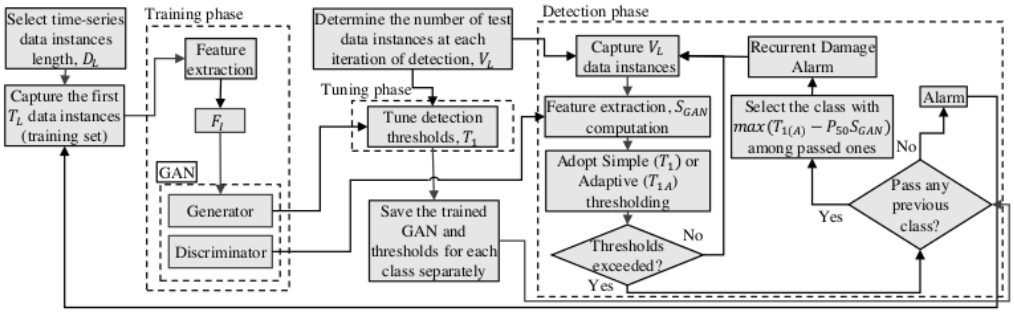


Figure 5.1: Unsupervised SHM framework

5.2.1 Unsupervised novelty detection

The proposed GAN framework has three main phases: Training, Tuning, and Detection, shown in Fig. 5.1 A GAN model consists of two neural networks, a discriminator $\mathcal{D}$ and a generator $\mathcal{G}$, trained online during the training phase. The trained GAN is further employed in both tuning and detection phases. The incoming data (acceleration in our case) must be divided into time-series data instances before the training phase for signal processing purposes. Time-series data instances are defined to have D_L total samples (*i.e.*, data points).

Training phase: The feature GAN models are trained with is designated as F_I, representing the concatenation of different channels' half-spectrum FFT (*i.e.*, amplitudes), normalized by each channel's mean value, and capped at 10. With N channels, F_1 has the dimension of $\frac{D_L}{2} \times N$. The trained $\mathcal{D}$ outputs a probability (*i.e.*, sigmoid activation) for a data instance's F_I feature (x_{F_I}) to be real or fake. From that output, the novelty detection score, S_{GAN}, is defined as

$$S_{GAN}(x_{F_I}) = -\log(\mathcal{D}(x_{F_I})). \tag{5.1}$$

Tuning and Detection phases: At each detection iteration, S_{GAN} scores of a batch of data instances with a length V_L is assessed (Fig. 5.1). That batch of data instances constitutes a vector of S_{GAN} scores, *i.e.*, $\vec{S}_{GAN}$. The median, an outlier-resistant centrality measure, is selected to define the representative value of that vector of scores, *i.e.*, $P_{50}(\vec{S}_{GAN})$. That representative value is then contrasted to a threshold in a simple threshold model to alarm novelties, defined as

$$T_1 - P_{50}(\vec{S}_{GAN}) > 0, \tag{5.2}$$

flagging a novelty when $P_{50}(\vec{S}_{GAN})$ exceeds the threshold T_1.

As discussed, the framework is deployed with no prior information, so no offline data is available to tune T_1. Instead, the $\mathcal{G}$ from the trained GAN model is used to synthesize a large amount of non-novel data instances. The threshold is tuned based on the principle that no novelties must be identified from the generated data instances, *i.e.*, $\overline{\mathcal{G}}$, as $\mathcal{G}$ has learned to mimic the non-novel data. Thus, the tuning can be summarised as

$$T_1 - max(S_{GAN}(\overline{\mathcal{G}})) > 0. \tag{5.3}$$

Similar to the detection phase, $\mathcal{G}$ generates data instances in a vector of length V_L, and T_1 must be greater than the largest S_{GAN} of generated data instances, or conservatively equal to it. $\mathcal{G}$ samples from a probability density function; thus, sampling techniques are utilizable to ascertain $max(S_{GAN}(\overline{\mathcal{G}}))$.

$\overline{\mathcal{G}}$ might contain misleading generations; this matter is shown in several SHM-related papers that employed GAN [92, 116]. The reason that the probability density function is a continuous function with infinite points, and the $\mathcal{G}$ is not uniform across that distribution. Consequently, while tuning T_1, the informative and non-informative generations must be separated. The two classes are distinguishable by fitting a 2-class Gaussian mixture model

(GMM) to the generated scores outlined in Algorithm 2. The two classes represent the informative (G_1, lower scores) and possible non-informative generations (G_2, higher scores). Suppose the mean of G_2 (candidate outlier) is three or more standard deviations of G_1 away from the mean of G_1 (*i.e.*, a simple outlier detection paradigm in Gaussian distributions). In that case, G_2 is labeled as non-informative and is discarded when calculating T_1. During the Detection phase, and to enable resurgent novelty detection, the GAN model is saved for each newly encountered novelty (Fig. 5.1). Then, in the presence of a novelty (*i.e.*, exceeding thresholds), it replaces the current $\mathcal{D}$, and the ones that do not fail are separated (*i.e.*, resurgent-novelty candidates). If several past classes pass this criterion, the one with the greatest difference between the detection threshold and the detection score is selected. Evaluation of the proposed recurrent novelty detection is presented in the results section.

Algorithm 2: Tuning-phase

Require: n, $\mathcal{G}$ ▷ n is the number of samples
  $\overline{\mathcal{G}} \in \mathbb{R}^{\frac{D_L}{2} \times V_L \times n} \leftarrow$ generate V_l data instances, n times
  $\mathcal{D}(\overline{\mathcal{G}}) \in \mathbb{R}^{V_L \times n}$ ▷ get the $\mathcal{D}$ scores
  $X \in \mathbb{R}^{1 \times n} \leftarrow$ get the maximum of $\mathcal{D}(\overline{\mathcal{G}})$'s second dimension
  $G_1, G_2 \leftarrow$ train a 2-class GMM on X
  If mean(G_2)>mean(G_1)+3std(G_1)
    discard G_2, $G_2 \leftarrow \emptyset$
  end
  $T_1 = max(G_1, G_2)$

5.2.2 GAN Architecture

In the context of novelty detection, novel neural network architectures, such as CNN and RNN, were practiced in the literature. For instance, CNN were operated on both 1D signals [105], and 2D images [41, 106]. 2D CNN can also receive sensors' data in the format of a 2D grid [81, 82], preserving the spatial distribution of sensors. On the other hand, RNN were employed to construct time-series structural responses [72] and sensor outputs [40, 72].

Accordingly, this study employs fully connected (Dense), CNN, and RNN architectures for GAN models' $\mathcal{D}$. $\mathcal{G}$ retains the same Dense architecture as shown in Fig. 5.2. $\mathcal{D}$ must have one output with a sigmoid activation function, as it represents a binary, probabilistic classifier (Equation 5.1). Other activation functions, such as ReLU, are employable by defining a new score that can capture the difference in those functions' output for novel and non-novel data instances. These are the basis for the Dense-based $\mathcal{D}$ architecture, shown in Fig. 5.3.

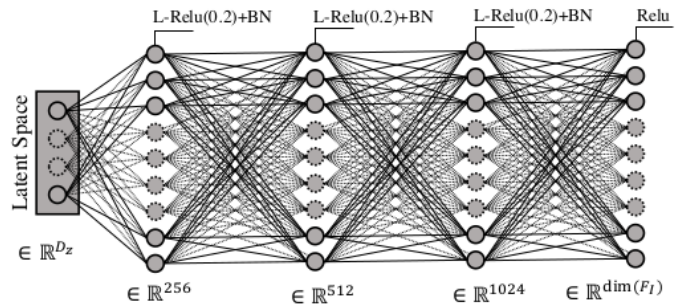


Figure 5.2: Dense $\mathcal{G}$ architecture, L-ReLU(α): Leaky ReLU activation with negative α slope, BN: batch normalization.

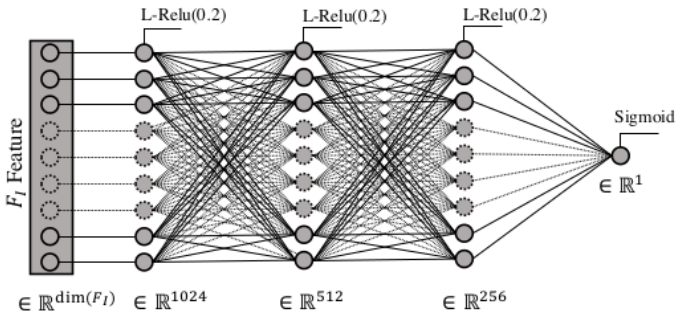


Figure 5.3: Dense $\mathcal{D}$ architecture, L-ReLU(α): Leaky ReLU activation with negative α slope, BN: batch normalization.

The CNN-based $\mathcal{D}$ architecture is shown in Fig. 5.4. The F_I feature is a matrix with N columns (*i.e.*, number of channels) that is first cut into vectors for each channel for the application of 1D convolutional layers. For clarity, strides and paddings are not shown in Fig. 5.5; their selection per the F_I feature must result in a single output.

The Long-Short-Term-Memory (LSTM)-based $\mathcal{D}$ is shown in Fig. 5.5. Each channels' FFT is passed into single-sized different multi-layer (*i.e.*, stacked) LSTMs. LSTM architectures are often used for sequential data (*e.g.*, time signals) in the field of SHM. FFT of signals is also a sequential feature, assuming the structure response is composed of several harmonics (*i.e.*, modes) with different attributes. In that case, the FFT of various structural conditions is unique, forming sequential data: distinct spectral dependencies define unique structural states (*i.e.*, modes). Accordingly, LSTM reads the spectral dependencies from the sequential feature to detect novelties.

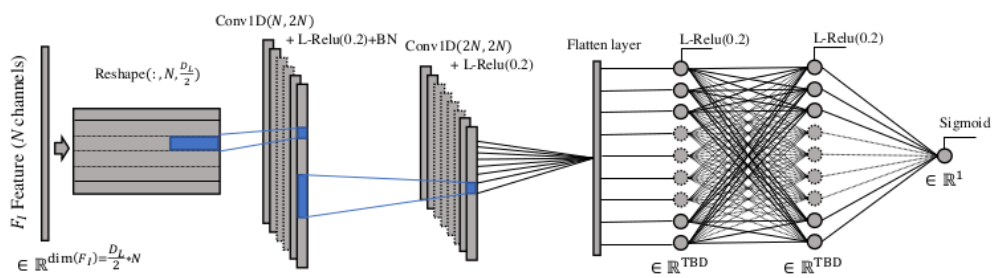


Figure 5.4: CNN-based $\mathcal{D}$ architecture, L-ReLU(α): Leaky ReLU activation with α negative slope, Conv1D(i,o): 1D convolutional layer with i input and o output channels, respectively.

5.2.3 GAN training and loss patterns

As discussed, GANs consist of two neural networks, namely, a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The training data instances contain no labels (*i.e.*, unsupervised), and arrive from an unknown data distribution P_{data}. These are assumed to be "real" or "normal" data. $\mathcal{G}$ imitates real data by manipulating a sample noise (*i.e.*, z) drawn from the latent space (*i.e.*, P_{lat}) to have the same dimension of the real data (*i.e.*, $N.D_L/2 \times 1$). $\mathcal{G}$'s outputs are labeled as "fake" data. During training, and at each iteration, $\mathcal{D}$ distinguishes between "real" and

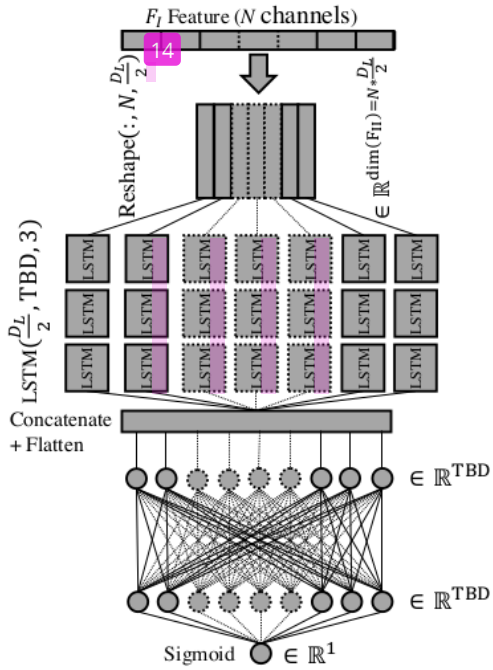


Figure 5.5: LSTM-based $\mathcal{D}$ architecture, LSTM(i,h,n) with input size of i, hidden size of h, and n stacked layers.

"fake" data. The training loss for $\mathcal{D}$ is defined as

$$\ell_{\mathcal{D}} = -\mathbb{E}_{x_{F_I}\sim P_{\text{data}}}\left[\log \mathcal{D}(x_{F_I})\right] - \mathbb{E}_{z\sim P_{lat}}\left[\log\left(1-\mathcal{D}(\mathcal{G}(z))\right)\right], \tag{5.4}$$

where x_{F_I} is the feature representation for "real" data. This loss function penalizes $\mathcal{D}$ when "fake" data are classified as "real", and vice versa. The generator $\mathcal{G}$ loss function is defined as

$$\ell_{\mathcal{G}} = -\mathbb{E}_{z\sim P_{lat}}\left[\log \mathcal{D}(\mathcal{G}(z))\right]. \tag{5.5}$$

The generator learns to generate real-resembling data points since the loss penalizes $\mathcal{G}$ when the generated "fake" data are not good enough to fool $\mathcal{D}$, which tries to distinguish between normal ("real") and novel data ("fake") in the detection phase.

Excluding the problem of mode-collapse during training [38], a typical GAN training loss can have several patterns observed, which we term as (1) learning initiation, (2) intertwined-learning, (3) $\mathcal{D}$-conquered, and (4) loss divergence/failure, illustrated in Fig. 5.11. While all the patterns might not be observed during training, based on the difference in the $\mathcal{G}$ and $\mathcal{D}$ architectures, after the learning initiation step, any of the other three patterns are likely to appear. However, in cases where $\mathcal{G}$ cannot ultimately compete with $\mathcal{D}$, the $\mathcal{D}$-conquered loss pattern would be observable with the loss of $\mathcal{D}$ going towards zero. With limited training sets, regardless of $\mathcal{D}$ and $\mathcal{G}$ architectures and their relative learning capacities, deep learning models in general [99], including GAN's discriminator [48] tend to overfit the training data. In such cases, deep multi-class, or binary (*i.e.*, $\mathcal{D}$) models can reach a stable decision boundary due to manageable distributions of limited data and the learning capacities of such models. The gap in Intertwined learning may vary, depending on $\mathcal{D}$'s superiority over $\mathcal{G}$. In this study, and for the intertwined learning and $\mathcal{D}$-conquered patterns, several detection simulations are carried out on different $\mathcal{D}$ architectures to pinpoint the best pattern for SHM tasks.

5.2.4 Adaptive threshold

Thresholds are tuned so that no novelties are identified from the GAN-generated data instances through the Tuning phase. Two sources of bias are not accounted for in the tuning process. Based on Fig. 5.11, the two biases are (1) unseen data from the same class after the Training phase and (2) the level of $\mathcal{D}$'s overfitting. With the same $\mathcal{D}$-$\mathcal{G}$ pair (*i.e.*, the same loss-interactions), the sudden increment in the detection scores at two points of Training-to-Detection and normal-to-novel data differs with characteristics of unseen data instances and the overfitting of $\mathcal{D}$ are the biases.

The incoming detection scores of unseen data instances, containing the impact of overfitting and unseen data characteristics, are mingled into the Simple threshold to remove the bias. A hypothesis is proposed to advance the mixing design: in the most extreme overfitting case, S_{GAN} for the training set is zero, and the variation in $\overline{\mathcal{G}}$ scores (*i.e.*, tuned thresholds) represents the possible deviation range of a test data over that zero-$\mathcal{D}$ score to be not categorized as a novelty. Thus, the zero is replaced with a representative value from the incoming data stream during the detection. That representative value is chosen to be the running average of 50th percentiles of detection scores' batches (*i.e.*, each batch is a vector of scores with the length V_L), from time-step 0 (*i.e.*, transition from Training to Detection phase, Fig. 5.11) to the current time-step t. The new threshold is designated as the Adaptive threshold and is defined as:

$$T_{1A}(t) = T_1 + \text{mean}(P_{50} S_{GAN}(0 - t)). \tag{5.6}$$

The chosen running average thresholds add a plateau to the detection scores, preventing false alarms. The Adaptive and Simple thresholds are studied for various architectures to assess the Adaptive thresholds effectiveness.

5.3 Experimental details

5.3.1 Datasets

We utilize two benchmark SHM datasets, Qatar University Grandstand Simulator (QUGS) and Yellow Frame.

QUGS aims to have complete stadia models with seats, risers, treads, seats, and handrails,

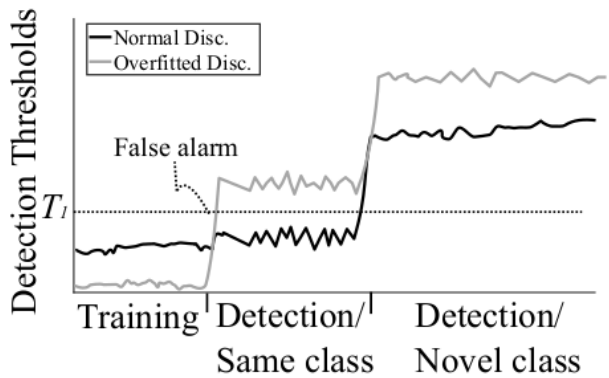


Figure 5.6: Sharper rise of detection scores for unseen data instances, due to an overfitted $\mathcal{D}$.

to develop damage defection methodologies based on the model's acquired experimental data. Currently, the structure in QUGS contains a test frame with a capacity of 30 spectators. The details for sections and geometries are presented in Avci et al. [12]. The structure and sensor placements are shown in Fig. 5.7. This dataset contains 31 data classes (one intact and thirty damaged states). Sensors are installed near all bolts; consequently, all possible damages have sensors installed at their locations (Fig. 5.8). All damage scenarios were captured via accelerometers with a sampling frequency of 1024 Hz over 256 seconds, resulting in 262,144 data points. This dataset's damage scenarios are as small as loosened bolts, rendering it useful to test the proposed framework's detection abilities for small damages. The contributions of the proposed technique in this study regarding the most recent studies on this dataset [2, 12] are explained in the results section.

Yellow Frame is a one-third modular four-story steel frame erected at the University of British Columbia [63], acting as a benchmark dataset for evaluating the most recent improvements in SHM methodologies. The modular frame is shown in Fig. 5.9. This frame's modular components (*i.e.*, braces, beams, and masses) are manipulable for simulating diverse damage scenarios. More information on the structure can be found in Mendler et al.

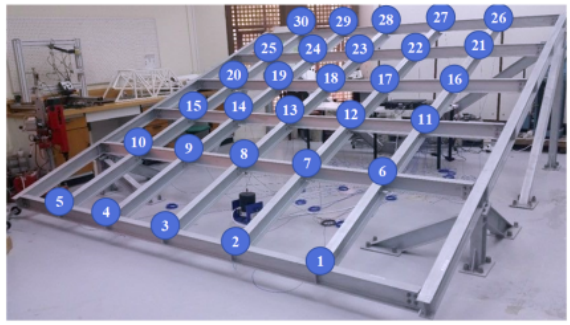


Figure 5.7: QUGS structure

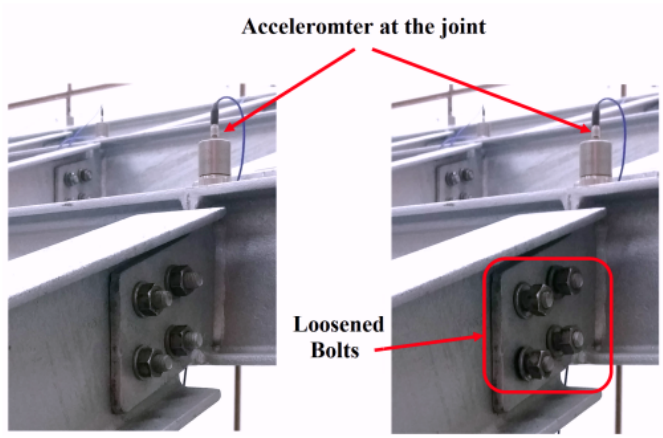


Figure 5.8: Induced damages in QUGS

[63]. A benchmark dataset from the available Yellow Frame datasets with twenty-one classes (one intact and twenty damaged states) is selected in this study. In this dataset, the frame is instrumented with fifteen accelerometers with a sampling frequency of 1000 Hz (Fig. 5.10), with detailed instrumentation explained in Allahdadian [4].

5.3.2 Research questions

In this study, four main research questions are addressed: (1) the framework's performance in different loss patterns that a GAN model encounters during the training, (2) the best

Figure 5.9: Yellow Frame structure

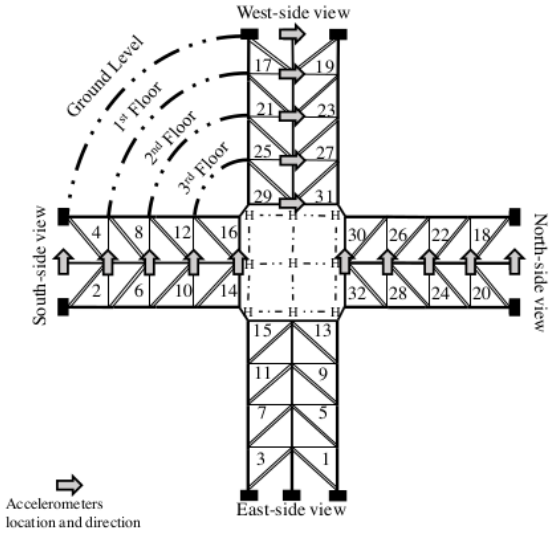


Figure 5.10: Sensor placements in Yellow Frame

neural network architecture for $\mathcal{D}$, (3) the possibility of recurrent novelty detection, and (4) the detection robustness to structures' sensor configurations. The first analysis is for the selection of the best GAN training loss pattern to select $\mathcal{D}$ and $\mathcal{G}$ from, and for all $\mathcal{D}$ architectures studied in the Loss Pattern analysis. In the Architecture Selection analysis, a permutation of different classes of damages is monitored (for the selected loss pattern) to test the robustness of the detection results. Then, the best $\mathcal{D}$ architecture is decided. In the

Recurrent Novelty Detection analysis, the recurrent damage detection algorithm is analyzed, evaluating its effectiveness. Finally, in the Sensor Configuration analysis, the framework's robustness concerning different sensor plannings is evaluated.

Loss Pattern analysis

Loss Pattern analysis consists of three stages. The first step is to compare Dense, CNN, and LSTM architectures with the assessment of novelty detection results. The metrics for the assessment are detection's true and false alarm ratio statistics. The true alarm ratio is the number of correctly detected novelties divided by the total number of novel classes, with the perfect detection having the true alarm ratio of 1. The false alarm ratio is the number of wrongly detected novelties divided by the total number of detection windows in the entire dataset. The false alarm ratio of 0 indicates a perfect result. The training is performed for each class of damages until all discussed trends (Fig. 5.11) are observable on GANs' training loss diagrams.

In the Loss Pattern analysis, the training loss diagram is split up manually into three sections, namely, initial learning, intertwined learning, and $\mathcal{D}$-conquered portions (Fig. 5.12). The initial learning section is put aside, and for the rest two portions, each part is divided into two sub-parts. The GAN model continues to learn during each loss pattern, and the division of each section into two subparts is for discerning the effect of GAN modes' updates. Three epochs are randomly selected from those, resulting in 12 trained sets of $\mathcal{D}$ and $\mathcal{G}$ for each class of damage, as shown in Fig. 5.12. The wide selection of epochs over each range enables a robust statistical inference for different training patterns.

In the GANs' training losses, and especially for the $\mathcal{D}$-conquered loss pattern, there are points in which $\mathcal{D}$ and $\mathcal{G}$ losses are intertwined (I-points), having huge chaotic behavior

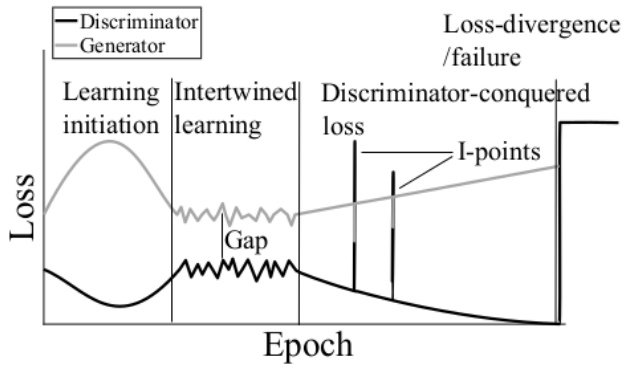


Figure 5.11: Potentially observable GAN training loss patterns

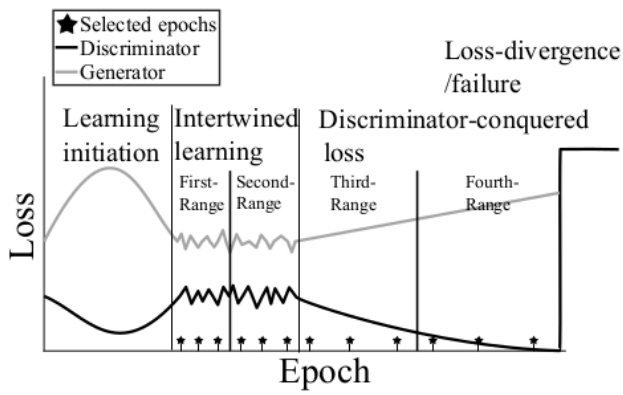


Figure 5.12: Loss Pattern Analysis

(Fig. 5.11). In such points, $\mathcal{D}$ is weak at identifying a feature that does not resemble the class understudy as fake, and the loss function heavily changes $\mathcal{D}$ by teaching that specific pattern to it. Each of the I-points represents an improvement in the detection potential of $\mathcal{D}$. However, these points must be avoided while selecting epochs from those ranges, as they are not an indicator of "$\mathcal{D}$-conquered" loss pattern. So for Loss Pattern analysis in $\mathcal{D}$-conquered regime, the epochs are randomly selected from the $\mathcal{D}$-conquered section while I-points are avoided.

Architecture Selection analysis

The detection results from the Loss Pattern analysis can be biased towards the arrangement of the data classes. For instance, the second damage may be detectable after the first damaged class and not after the fifth class. Hence, with the loss pattern selected, a permutation detection scenario is performed in Architecture Selection analysis to identify the most suitable architecture. However, the permutation detection will exponentially raise the total number of required analyses.

Recurrent Novelty Detection analysis

For the Recurrent Novelty Detection analysis, scenarios of recurrent damages are synthesized to evaluate the recurrent damage detection capability of the proposed framework. During the analysis, and at points of damage initiation (*i.e.*, a previously witnessed damage), three expected outcomes are possible: (1) no damage is detected, (2) the damage is detected; however, the previous damages are not identified, and (3) the framework alerted the presence of a previously witnessed damage. The recurrent damage accuracy is calculated as the number of correct recurrent damage identifications divided by the number of times the damage initiation is recognized.

Sensor Configuration analysis

Sensor Configuration analysis investigates the novelty detection's robustness per sensor configurations. For sensor reduction, new datasets with different input channels are constructed with fewer dimensions. Then Loss Pattern, Architecture Selection, and Recurrent Novelty Detection analyses are carried out, and the performance effect of sensor configuration is studied.

Table 5.1: QUGS dataset description and induced damages

Damage class	Loosened bolt ID	Damage class	Loosened bolt ID
SDC1	None	SDC2	1
SDC3	5	SDC4	7
SDC5	9	FDC6	13
SDC7	18	SDC8	22
SDC9	24	SDC10	26
SDC11	30		

Table 5.2: Yellow Frame description, and induced damage

Damage class	Removed brace ID	Damage class	Removed brace ID
FDC1	None	FDC2	2,4(II)
FDC3	FDC2+(18,20)(II)	FDC4	FDC3 +(1,3,17,19)(II)
FDC5	FDC2+(17,19)(II)	FDC6	FDC2+(18,20)(I)
FDC7	2(II)	FDC8	(2 , 4)(I)
FDC9	(25,27)(I)	FDC10	(29,31,8,6)(I)
FDC11	(21,23,29,31)(I)		

One brace is removed.
Both braces are removed.

5.3.3 Hyper-parameters & Experimental Setup

Eleven data classes (one normal and ten damaged classes) are selected for experimental analysis to maintain novelty detection settings across datasets. The selected classes and damage descriptions are reported in Table 5.1 and Table 5.2 for QUGS and Yellow Frame datasets. For QUGS, only ten sensors corresponding with the ten intended damages are selected; however, all 15 accelerometers' data are employed for Yellow Frame.

D_L is set to be 1000, and a fixed training length (T_L) of 100 is used for both datasets, resulting in approximately 100-second signals. Moreover, the detection length (V_L) of 10 and 20 are set for QUGS and Yellow Frame datasets, respectively. The n parameter is also

set to be 200000. We train with ADAM optimizer [49], with three hyper-parameters, namely, β_1, β_2, and learning rate. The learning rate of $1.00\text{E}-4$, β_2 of 0.999, and β_1 of 0.5 and 0.9 are employed for the Dense-, CNN-, and LSTM-based $\mathcal{D}$, respectively. For $\mathcal{G}$, D_z is set to be 100. A sample Google Colaboratory Jupyter notebook, including the neural network architectures for the QUGS dataset with $D_L = 1000$ and ten sensors is available at this link.

In the Loss Pattern analysis, and for each range in each architecture, 59,049 full (3^{10}) detections are performed. For Architecture Selection analysis, there are 11 classes; hence, the total class permutations is $11! = 39,916,800$, which is multiplied by 3^{11} for different epochs of the selected range. It is worth mentioning that since the framework's generalizability is under study, the best architecture/loss-trend selection is just performed for the QUGS dataset and tested on Yellow Frame to evaluate the method's robustness.

In Recurrent Novelty Detection analysis, first $\mathcal{D}$ and $\mathcal{G}$ are trained for all damage classes, and the trained models are saved. Then, the T_L portion of those classes is removed, and the truncated classes are permuted to evaluate the method's capability to detect the correct resurging damages. To have a robust estimation of sensor configuration's impact on novelty detection, and for each dataset, the complete permutation of sensors is the ideal option. For instance, regarding Yellow Frame with 15 sensors, the total sensor permutations are $2^{15} - 1 = 32767$, which must be multiplied by the Permutation Detection analysis instances of $3^{11} * 11!$, that sums up to 2.31×10^{17} detections. This level of analysis is prohibitive in terms of time and computational infrastructure limitations. Hence, for both datasets, the goal is set to maintain a non-biased plan of sensor reduction. Since on both datasets, damages are all across the structures; such unbiased approaches are accessible. For QUGS, the sensors are reduced by going towards the center of the stadia, reaching only the two sensors at its center (Fig. 5.13). For Yellow Frame, first, sensors are reduced to three by randomly selecting a side of the frame and keeping only the first three floors' sensors for the final reduction plan

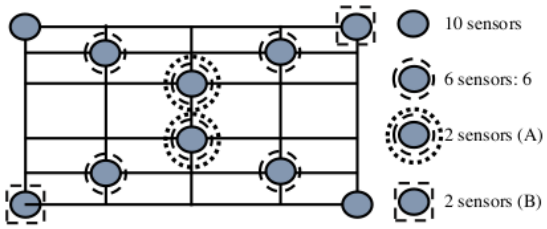


Figure 5.13: QUGS sensor selection paradigm

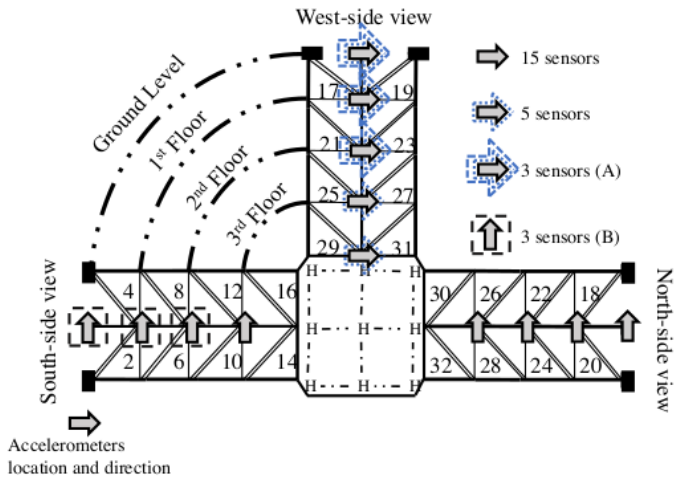


Figure 5.14: Yellow Frame sensor selection paradigm

(Fig. 5.14). On both datasets' reduction plans, two extra cases are considered for the lowest number of sensors to remove possible overlapping between configurations, designated with the "(B)" on those figures. .

5.3.4 Supplementary analyses

Supplementary examinations are performed to test the framework w.r.t. time complexity and performance sensitivity to user-defined parameters n and V_L.

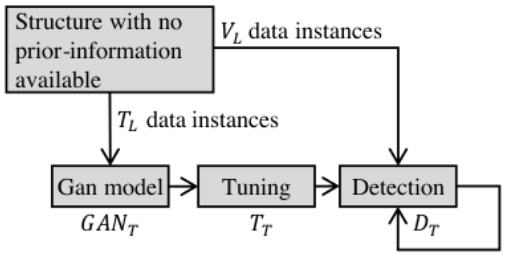


Figure 5.15: Time complexities of detection framework's components

Time Complexity analysis

One of the objectives of the proposed framework is to enable real-time novelty detection. Per framework (Fig. 5.1), the time complexities of different phases are summarized in Fig. 5.15. In this figure, for the first T_L data instances, the GAN model is trained with the time complexity of GAN_T followed by the Tuning phase with the complexity of T_T. With the thresholds tuned, the detection process takes place on blocks of data instances with the time complexity of D_T for each data block with the length of V_L. Since the Detection phase is block-wise, if the detection time for each block (D_T) is less than its duration, the method becomes real-time after several iterations. The detection phase is simply assembled from taking the FFT of a batch of data instances, getting the $\mathcal{D}$ output for them, and checking a single threshold; thus, saving time at each detection iteration is a probable outcome to compensate for GAN_T and T_T. Regarding different sensors and the two datasets, the number of parameters in the employed neural network architectures is reported in Table. 5.3 for all models.

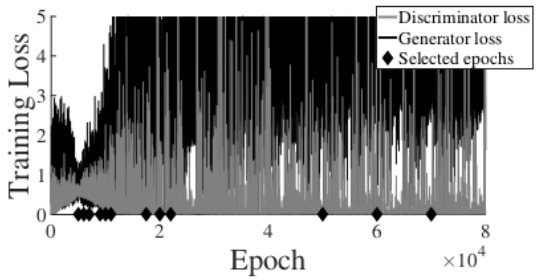
Detection Sensitivity analysis

The proposed framework has four parameters, namely, D_L, T_L, V_L, and n, from which V_L and n are detection parameters, directly controlling the detection results. Hence, sensitivity

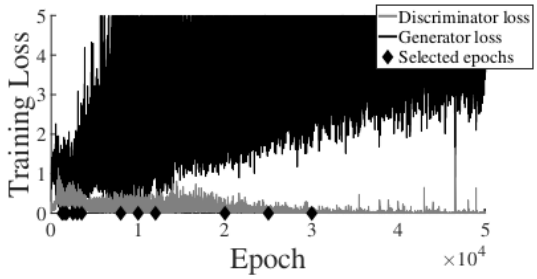
Table 5.3: Number of parameters in employed neural networks

	Dense		CNN		LSTM		LSTM†		LSTM‡	
15 sensors	$\mathcal{D}$	$\mathcal{G}$	$\mathcal{D}$	$\mathcal{G}$	$\mathcal{D}$	$\mathcal{G}$	$\mathcal{D}$	$\mathcal{G}$	$\mathcal{D}$	$\mathcal{G}$
Yellow Frame	1.2 e7	1.2 e7	6.6 e5	1.2 e7	3.6 e6	1.2 e7	1.2 e6	4.6 e6	7.5 e5	2.2 e6
QUGS	5.7 e6	5.8 e6	6.6 e5	5.8 e6	4.1 e6	5.8 e6	2.4 e6	3.7 e6	8.2 e5	1.1 e6

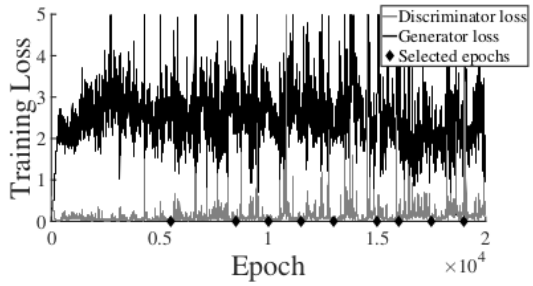
†5 sensors for Yellow Frame and 6 sensors for QUGS.
‡3 sensors for Yellow Frame and 2 sensors for QUGS.



(a) Dense architecture



(b) CNN architecture



(c) LSTM architecture

Figure 5.16: SDC4 GAN's training loss; QUGS

analyses are performed for V_L and n. Moreover, during the Detection Sensitivity analysis, a novel majority vote ensemble paradigm is practiced. As explained, GAN models, after training, have numerous pairs of $\mathcal{G}$ and $\mathcal{D}$. Neural network training is a stochastic process with each pair possessing distinct characteristics regarding false or true alarm detection. Employing the ensemble of simple machine learning tools is a well-known practice. In this study, aside from selecting three different epochs at each range (Fig. 5.12) and measuring the overall performance, the ensemble of those three epochs for each data class in a majority vote setting is likewise practiced during the Detection Sensitivity analysis.

5.4 QUGS Results and Discussion

The best loss pattern and architecture are selected in Loss Pattern and Architecture Selection analysis, respectively. Furthermore, the performance of the recurrent damage detection algorithm and the robustness of detection results regarding different sensor configurations is reported in Recurrent Novelty Detection and Sensor Configuration analysis, respectively.

5.4.1 Loss Pattern analysis results

GANs with varying Dense-, CNN-, and LSTM-based $\mathcal{D}$ are trained. A sample training pattern for class SDC4 is shown in Figs. 5.16a, 5.16b, and 5.16c, respectively, in which the selected epochs are marked as explained in Section 5.3.2. For the LSTM-based $\mathcal{D}$, the whole training loss diagram is in the Third-and Fourth-ranges (Fig. 5.16c). The reasons are the LSTM-based $\mathcal{D}$'s versatility at identifying patterns in sequential data, such as the FFT of signals [92]. Hence, three sets of different epochs are defined for the LSTM $\mathcal{G}$ based on experiencing I-points (Section 5.3.2).

Loss Pattern analysis results' statistics are reported in Table. 5.4. As expected, with the Simple threshold approach, the more the loss of $\mathcal{D}$ goes towards zero, the more the false alarm ratio increases. For the CNN models on the third range with Simple thresholds, the false alarm ratio is 0.86, indicating that each input vector of non-novel samples has around a 90% chance of being flagged as a false alarm. The Adaptive threshold approach alleviated that effect heavily and resulted in 100% damage detection with zero false alarms in the Fourth-range with CNN. The notable point in the LSTM architectures is that the false alarm ratio is zero regardless of the thresholds' type. This finding indicates that the LSTM models understand the sequential behavior of the FFT-based features, and when transiting between Training and Detection phases (*i.e.*, unseen data objects from the same class), no significant sudden changes occur in the detection scores while retaining high sensitivity to the true alarms. Furthermore, since the LSTM loss pattern is defined based on experiencing I-points, the results indicate that the detection accuracy increases with more I-points. The $\mathcal{D}$-conquered pattern and specifically the Fourth-range (*i.e.*, $\mathcal{D}$ loss go towards zero) has the most consistent high accuracy and low false alarm ratios with the Adaptive thresholds.

Fig. 5.17 depicts a sample detection from about the fifty-two thousand simulations of the Fourth-range Dense and CNN architectures and third group of selected epochs of LSTM-based $\mathcal{D}$. A noticeable trend is observable for the LSTM architecture: no discernible sudden jump or variation can be seen on the detection scores after training.

Table 5.4: Loss Pattern analysis results; QUGS (Adaptive and Simple thresholding)

		Dense		CNN		LSTM	
Range	**Accuracy metric**	**Adaptive**	**Simple**	**Adaptive**	**Simple**	**Adaptive**	**Simple**
First-range (First group of epochs for the LSTM-based $\mathcal{D}$)	True alarm ratio (mean ± std)	0.667 ± 0.094	0.820 ± 0.067	1.000 ± 0.000	0.691 ± 0.054	0.900 ± 0.000	0.900 ± 0.000
	False alarm ratio (mean ± std)	0.000 ± 0.000	0.000 ± 0.000	0.008 ± 0.006	0.605 ± 0.051	0.000 ± 0.000	0.000 ± 0.000
Second-range (Second group of epochs for the LSTM-based $\mathcal{D}$)	True alarm ratio (mean ± std)	0.522 ± 0.216	0.689 ± 0.172	0.900 ± 0.105	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	False alarm ratio (mean ± std)	0.000 ± 0.000	0.026 ± 0.021	0.000 ± 0.000	0.816 ± 0.074	0.000 ± 0.000	0.000 ± 0.000
Third-range (Third group of epochs for the LSTM-based $\mathcal{D}$)	True alarm ratio (mean ± std)	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	False alarm ratio (mean ± std)	0.000 ± 0.000	0.189 ± 0.035	0.000 ± 0.000	0.860 ± 0.036	0.000 ± 0.000	0.000 ± 0.000
Fourth-range	True alarm ratio (mean ± std)	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-	-
	False alarm ratio (mean ± std)	0.000 ± 0.000	0.151 ± 0.345	0.000 ± 0.000	0.496 ± 0.044	-	-

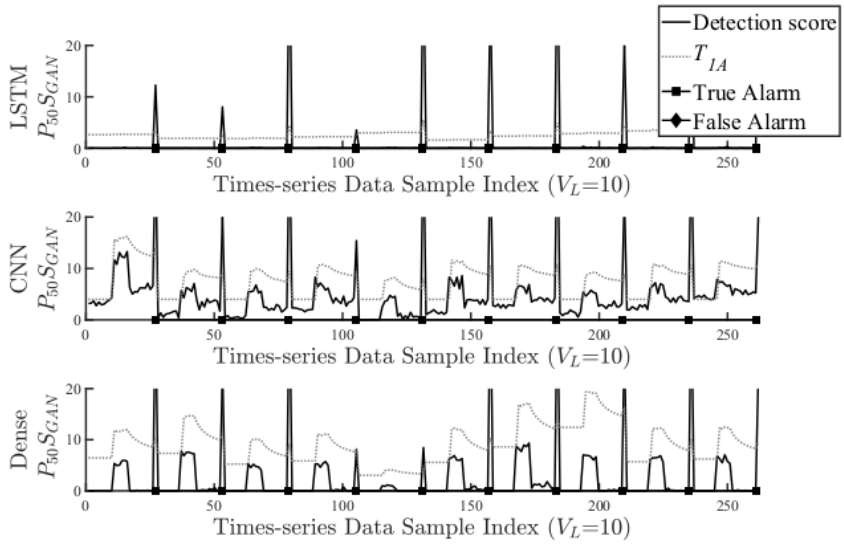


Figure 5.17: A sample real-time unsupervised novelty detection result for Dense, CNN, and LSTM-based architectures; Adaptive threshold approach

5.4.2 Architecture Selection and Recurrent Novelty Detection analyses results

In Architecture Selection and Recurrent Novelty Detection analyses, the permutation and recurrent damage scenarios are checked for the Fourth-range of the Dense and CNN-based $\mathcal{D}$ and the third group of selected epochs (*i.e.*, Fourth-range) of the LSTM-based $\mathcal{D}$. The Architecture Selection and Recurrent Novelty Detection analyses results are reported in Table. 5.5. According to Table. 5.5, regardless of the architecture of $\mathcal{D}$, if a novelty is defined, the framework can detect the presence of a previously seen novelty without even a single failure, and the conjoined recurrent damage detector section tends to work well. Based on the first three analyses, both LSTM and CNN architectures were superior to the Dense architecture. Among the CNN and LSTM model $\mathcal{D}$, CNNs shows a near-perfect detection result; however, the LSTM architecture has prominent superiority over the CNN model

Table 5.5: Architecture Selection and Recurrent Novelty Detection analyses results; QUGS

$\mathcal{D}$ architecture	True alarm ratio (mean ± (std))	Recurrent damage detection ratio (mean ± (std))
Dense	0.946 ± 0.004	1.000 ± 0.000
CNN	0.988 ± 0.004	1.000 ± 0.000
LSTM	0.975 ± 0.006	1.000 ± 0.000

for three reasons. The first reason is that the LSTM monotonous detection scores while transiting between the train and unseen data results in a low false alarm ratio on both types of thresholds. The second reason is that the LSTM networks are in the $\mathcal{D}$-conquered training loss phase from the beginning, and much fewer training epochs are required for them. The third reason is that the Adaptive threshold is designed to reduce the biases due to unseen data instances and overfitted $\mathcal{D}$ with promising improvements. However, incorporating the running average of detection scores can compromise the detection of progressive novelties, forming over time. Thus, the LSTM architecture that performs well on both thresholds is the preferable option. Overall, the results show the framework's robustness to the selection of $\mathcal{D}$ architecture with near-perfect detection results in their Fourth-range.

5.4.3 Sensor Configuration Analysis

Sensor Configuration analysis is carried out for the selected architecture (*i.e.*, LSTM), the selected epochs (*i.e.*, Fourth-range), and the Adaptive threshold, with the results reported in Table. 5.6. The results indicate the detection result's robustness to the sensor configuration, in which, with only two sensors, virtually the same detection result occurs.

5.4.4 Contributions w.r.t. prior works

Several studies applied to the QUGS dataset. Some studies utilize CNN networks for feature extraction, alleviating the structure-specific feature extraction and loss of information issues. In Avci et al. [12], the QUGS dataset had 11 classes with ten sensors, and in Abdeljaber et al. [2], the 31-class was employed.

Our proposed GAN-based method is unsupervised and can be applied without any former information about the structure compared to prior works. In Abdeljaber et al. [2], Avci et al. [12] the presence of novelties is identified based on the probability of damage, the "PoD" score. However, a probabilistic framework that can define thresholds for damage scores is of great importance as it can automate the detection process in a real-time setting. This study automatically extracts detection thresholds using GAN's generated data (*i.e.*, Tuning phase) with zero false alarms and 100% accuracy in several scenarios. Besides, with only two sensors, the same level of detection occurred. Fewer sensors indicate less cost, easier data acquisition, and no need for time-consuming sensor placement optimization, at least for the studied dataset herein.

Table 5.6: Sensor Configuration analysis results; QUGS (Adaptive and Simple thresholding)

		10 sensors		6 sensors		2 sensors (A)		2 sensors-B	
Analysis	**Accuracy metric**	**Adaptive**	**Simple**	**Adaptive**	**Simple**	**Adaptive**	**Simple**	**Adaptive**	**Simple**
LP	True alarm ratio (mean ± std)	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	False alarm ratio (mean ± std)	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
AS	True alarm ratio (mean ± std)	0.975 ± 0.006	-	0.969 ± 0.008	-	0.982 ± 0.006	-	0.979 ± 0.007	-
RND	Mean recurrent detection (mean ± std)	1.000 ± 0.000	-	1.000 ± 0.000	-	1.000 ± 0.000	-	1.000 ± 0.000	-

†Loss Pattern analysis, ‡Architecture Selection analysis, * Recurrent Novelty Detection analysis.

5.5 Yellow Frame results and discussion

Here, the LSTM-based $\mathcal{D}$ concerning results from the previous dataset is used. Furthermore, for evaluating the impact of alteration in the LSTM's architecture, a weaker LSTM architecture with only one stacked layer in place of three stacked layers is employed, resulting in more challenging novelty detection.

5.5.1 Loss Pattern, Architecture Selection, and Recurrent Novelty Detection analyses results

Since the LSTM-based GAN's training loss is in $\mathcal{D}$-conquered regimen, which is the same as with QUGS, three ranges are defined. In the Yellow Frame case, detection is carried out in the Fourth-range (Fig. 5.12) with the LSTM-based $\mathcal{D}$. A sample GAN's training loss is shown in Fig. 5.17 for FDC5, with marked selected epochs. Three detection results out of 52049 detection results in the Loss Pattern analysis are also shown in Fig. 5.19. Loss Pattern, Architecture Selection, and Recurrent Novelty Detection analyses results are reported in Table. 5.7. Both thresholds detect all novelties with no false alarms in Loss Pattern analysis with the simpler LSTM architecture. Architecture Selection analysis results indicate a mean detection of 9.67 damages, and the Recurrent Novelty Detection analysis offers a full recurrent damage detection.

5.5.2 Sensor Configuration analysis results

Based on the sensor reduction approach explained for Yellow Frame (Fig. 5.14), the three main analyses are performed with Adaptive and Simple thresholds, reported in Table. 5.8.

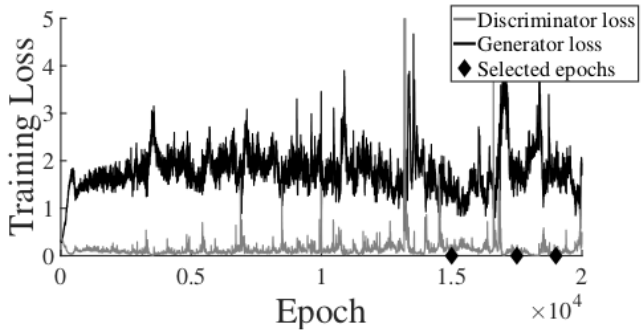


Figure 5.18: FDC5 GAN's training loss; LSTM architecture, Yellow Frame

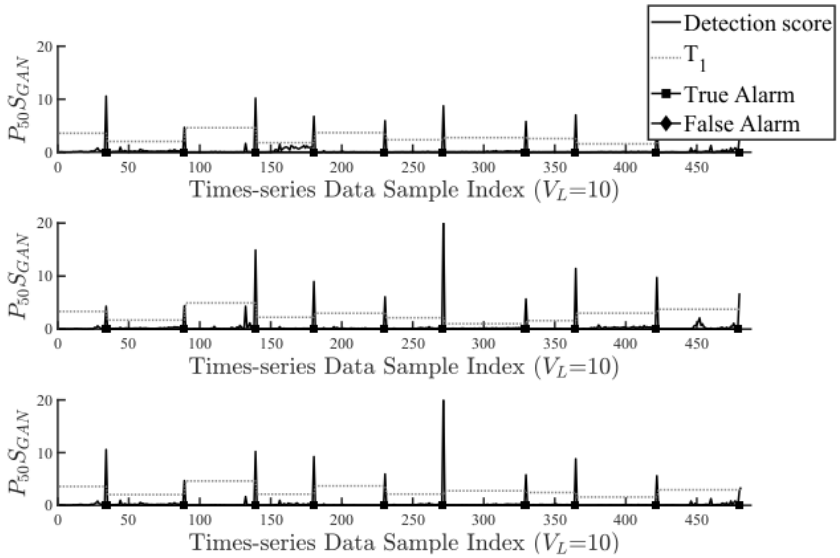


Figure 5.19: Samples of Yellow Frame real-time unsupervised novelty detection results for LSTM-based $\mathcal{D}$; Simple threshold approach

Table 5.7: Loss Pattern, Architecture Selection, and Recurrent Novelty Detection analyses results; Yellow Frame

		LSTM architecture	
Analysis	**Detection metric**	**Adaptive**	**Simple**
LP			
	True alarm ratio (mean ± std)	1.000 ± 0.000	1.000 ± 0.000
	False alarm ratio (mean ± std)	0.001 ± 0.000	0.001 ± 0.000
AS			
	Mean permutation detection (mean ± std)	0.963 ± 0.002	0.986 ± 0.021
RND			
	Mean recurrent detection (mean ± std)	1.000 ± 0.000	1.000 ± 0.000

†Loss Pattern analysis, ‡Architecture Selection analysis, * Recurrent Novelty Detection analysis.

Table 5.8: Sensor Configuration analysis results; Yellow Frame (Adaptive and Simple thresholding)

		15 sensors		**5 sensors**		**3 sensors**		**3 sensors-B**	
Analysis	**Accuracy metric**	**Adaptive**	**Simple**	**Adaptive**	**Simple**	**Adaptive**	**Simple**	**Adaptive**	**Simple**
LP									
	True alarm ratio (mean ± std)	1.000 ± 0.000	1.000 ± 0.000	0.922 ± 0.064	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	False alarm ratio (mean ± std)	0.001 ± 0.000	0.001 ± 0.000	0.005 ± 0.000	0.013 ± 0.010	0.003 ± 0.002	0.005 ± 0.002	0.006 ± 0.001	0.009 ± 0.004
AS									
	Mean permutation detection (mean ± std)	0.962 ± 0.002	-	0.954 ± 0.003	-	0.983 ± 0.002	-	0.987 ± 0.003	-
RND									

Loss Pattern analysis, ‡Architecture Selection analysis, * Recurrent Novelty Detection analysis

Based on Table. 5.8, even in the presence of three sensors, about the same detection is observable. Each data example is about 15 minutes in the Yellow Frame dataset, unlike the 4-minute-long data in QUGS. The longer duration shows that the framework is understanding novelties, not changes in the acquisition system behavior as time passes. Furthermore, from the eleven data classes, the first four are acquired consecutively from the structure on the same day, the next four on another day, and the rest on a third day, consecutively with less than 10 minutes stops between them. Hence, the framework is not detecting the loading/environmental changes, and it captures the presence of damages. The results show the detection results' robustness concerning sensor configurations.

The main reason behind acquiring such a high accuracy with two or three sensors is the feature's characteristics and selecting the most appropriate loss pattern and architecture. The high-dimensional feature is the whole half-spectrum of the signals; hence, even if the damage is far away from the sensors, or the number of novelties is much higher than the number of installed sensors, the effect of small or remote novelties can be seen on a portion of that half-spectrum FFT. If the detection is based on modes or any lowered-dimension feature, the novelties' effect might be deleted in the reduction process; however, the proposed framework can identify those novelties.

5.6 Supplementary analysis results

This section reports implementation details and results of the Time Complexity and Detection Sensitivity analyses.

5.6.1 Time Complexity analysis results

Per prior explanations in the Experimental details section (Fig. 5.15), the Time Complexity analysis is abstracted into the calculation of GAN_T, T_T, and D_T time complexities for the assessed architectures and sensor configurations. For conciseness, the analysis is performed for the Yellow Frame dataset with F_I of higher dimensionality (*i.e.*, higher complexity). For the different settings, G_T is reported in Table. 5.9. GAN training is implemented in upgraded Google Colaboratory (*i.e.*, PRO) with NVIDIA Tesla T4 GPU of 25.46 GB.

Results indicate the GAN training time to fluctuate around fifteen to twenty-five minutes regarding different architectures and the number of epochs. For instance, the LSTM-based $\mathcal{D}$ demands approximately 20 minutes of training to start the detection from an unseen structure with no prior information. Even though the training time entails accumulating 20 minutes of un-assessed data, the analysis time for detection blocks, D_T enables a real-time detection after several detection iterations. The fast execution time of the Detection phase is beholden to the simple arithmetic functions implemented in it, *e.g.*, taking the FFT and threshold comparison. For instance, for the LSTM model in Yellow Frame, GAN_T and D_T, resulting in the accumulation of 1236 seconds of un-assessed data. In the detection phase with $D_L = 1000$, $V_L = 20$, and $F_s = 1000$ Hz, each block of 20 seconds is assessed in less than 1 second. Therefore, at each detection occurrence, the analysis proceeds 19 seconds to un-assessed data. Eventually, after 65 detections (approximately 20 minutes), the analysis shifts in real-time. Since the method is robust per sensor configuration, with two sensors, the shift to the real-time state occurs in 4 minutes and 20 seconds. Based on the Time Complexity analysis, the framework can be regarded as real-time.

Table 5.9: Time Complexity analysis; Yellow Frame

Complexity (second)	**Dense $\mathcal{D}$ 80000 epochs**	**CNN $\mathcal{D}$ 50000 epochs**	**LSTM $\mathcal{D}$ 20000 epochs †**
GAN_T	1080	1550	1180 (222)
T_T	46	42	56 (23)
D_T	1	<1	<1 (<1)

†Values in parenthesis are for the 3-sensor scenario.

Table 5.10: Detection Sensitivity result; Yellow Frame

	Dense			
Threshold	$V_L = 10$	$V_L = 20$	$V_L = 30$	$V_L = 40$
Simple ($n = 200000$)	1.000 (0.018)	1.000 (0.005)	0.987 (0.005)	0.986 (0.002)
Simple ($n = 10000$)	1.000 (0.017)	1.000 (0.006)	0.991 (0.006)	0.988 (0.002)
Adaptive ($n = 200000$)	1.000 (0.008)	0.983 (0.003)	0.986 (0.004)	0.977 (0.001)
Adaptive ($n = 10000$)	1.000 (0.009)	0.983 (0.003)	0.983 (0.005)	0.977 (0.001)
Ensemble-Simple ($n = 200000$)	0.991 (0.005)	0.991 (0.000)	0.991 (0.000)	0.991 (0.000)
Ensemble-Adaptive ($n = 200000$)	0.991 (0.000)	0.991 (0.000)	0.991 (0.000)	0.991 (0.000)

5.6.2 Detection Sensitivity analysis results

Detection Sensitivity analysis applies to all the variations of studied $\mathcal{D}$ and sensor configurations. Only the Yellow Frame dataset with the lowest number of sensors is selected for conciseness, as higher sensitivity is expected with features of the least dimensionality. Four D_L and two n values are included in the analysis. V_L of 10, 20, 30, and 40 are chosen. Concerning n, *i.e.*, samples from $\mathcal{G}$ to tune thresholds, 10000 and 2000000 are practiced with a considerable difference to estimate the sensitivity thoroughly. Results are reported in Table. 5.10.

In anomaly detection, intrinsically, detection window size causes a trade-off between false and true alarms. With more stability in longer windows, false alarms are less prone to be detected, a positive outcome, while true alarms are compromised. According to Table. 5.10, increments of V_L has the same impact; fewer false and true alarms. Across both detests, enlarging the data size by the factor of 4, at most, the total accuracy is varied by around 2.3%. Regarding false alarms, the smallest V_L tends to have a sudden rise in the false alarms, receding with higher V_L. n is virtually neutral, conferring the insensitivity of framework to n; however, practicing a large value is advised. Overall, the sensitivity of the algorithm is insignificant; however, any improvement is of great advantage. One resolution is to use the ensemble technique with promising results. The ensemble heavily reduced the sensitivities since false alarms, unlike true damages, are often caused by different detectors' sensitivity to possible features in some data points, which are now removed in the majority vote design.

5.7 Conclusions and Future Work

Exercising Generative Adversarial Networks (GAN), consisting of a discriminator $\mathcal{D}$ and a generator $\mathcal{G}$, we proposed a GAN-based, prior-information-free, unsupervised, novelty detection framework in structural health monitoring. In this approach, $\mathcal{G}$ mitigates the lack of data through data generation, and $\mathcal{D}$ classifies novel and normal data instances. The main novelties of the approach are the demand for no prior information from structures, distinguishing between different novelties, and the robustness of novelty detection regarding structures' sensor configuration, key aspects of proposing general SHM approaches. More than 95% of novelties were identified in the studied datasets with only 100 seconds of training data. Such accuracy is attained with a comprehensive study on different GAN training loss patterns (*i.e.*, patterns of relative relations among $\mathcal{D}$ and $\mathcal{G}$ training losses) to select

the most informative $\mathcal{G}$, distinctive $\mathcal{D}$ pairs. As for model architectures, Long Short-Term Memory units (LSTM) were found to be the most suitable for several reasons, such as less sensitivity to overfitted $\mathcal{D}$, and stable novelty detection scores.

The proposed GAN-based model is a step towards unsupervised SHM in urban settings with various infrastructures. It functions on raw data, requiring no prior information with considerable robustness concerning sensor configuration. Future work should target collecting datasets with diverse and possibly occurring environmental and mechanical damages to shed light on the method's possible shortcomings and enable further improvements. The proposed method is shown to be robust to noise since it is applied in real-world structures; hence, future work can focus on long-time, multi-scenario, Finite-Element-based datasets.

Code release Source code will be released upon acceptance.

Acknowledgement The authors would like to thank Carlos Ventura, Alex Mendler, and Saeid Allahdadian for providing the Yellow Frame dataset and associated images.

Chapter 6:

Conclusion

Structural health monitoring (SHM) is interested in identifying novel structural states, aiding maintenance, and saving resources and human lives. Intrinsic complications in novelty detection, such as novelties to be unseen states, cause several challenges in data-driven SHM, supervised and unsupervised. In supervised SHM, novel data are scarce compared to normal data, driving data imbalance problems. Data generation with deep learning methods is a method of choice in different areas, such as computer vision, and is utilized herein to manage data imbalance in supervised SHM. Still, high-dimensional (raw) features and machine learning tools that handle such features are required to aid the data generation in SHM. Employing Long Short-Term Memory (LSTM) units for high-dimensional signal data interpretation is practiced for both SHM and sensor output validation (SOV). The high-dimensional feature enables training generative adversarial networks (GAN) to perform the data generation. Results indicate the effectiveness of LSTM networks in supervised SHM and SOV problems with over 98% mean accuracy across data classes in two benchmark datasets. Additionally, the generated data alleviates the data imbalance issues in both SHM and SOV, with more effectiveness in the SHM. The class classification accuracy in the minority class rose from 73.39% to 90.84%, and 44.77% to 64.58% for the SHM and SOV datasets, respectively.

Since novelties are unseen states in the future, unsupervised techniques are more suitable for

SHM. Identifying current challenges in unsupervised SHM is the key to developing strategies appropriate to real-world problems. Through the inspection of current approaches, four central challenges are identified: (1) loss of information due to the dimensionality reduction in features, (2) structure-dependent feature extraction, (3) lack of dynamic-class novelty detectors, (4) sensitivity to user-defined novelty detection parameters. The objective of this study is to address all four limitations simultaneously. Employing raw features is a possible solution to the first two challenges. The fast Fourier transform (FFT) of signals showed promising discrimination potential between different data classes coupled with the LSTM layers in the supervised study and hence utilized. Still, the main challenge is developing an approach to handle such a high-dimensional feature in an unsupervised SHM setting. We employed GAN to function with that high-dimensional feature. Regarding the third challenge, the main is the lack of data from detected novelties. If enough data are available, the same approach employed for the baseline condition applies to distinguish between novelties. GAN can generate enough samples to perform a dynamic detection and thus address the third challenge. Additionally, GAN is capable of data generation for the baseline class, developing a prior information-free technique. We used Monte Carlo histogram sampling to tune novelty detection thresholds for the fourth challenge. That procedure has two benefits: making the framework prior-information-free (*i.e.*, unlimited data generation) and removing possible biases in user-defined detection parameters.

The ultimate goal of SHM frameworks is to monitor different structures, solely or across an infrastructure network, for instance, in smart cities. Two key aspects of such a general approach are being prior-information-free and showing less sensitivity to sensor placement of infrastructure. This objective is attained by identifying the best loss pattern for GAN's training and selecting the best neural network architecture for the vibrational-based features in SHM. The Long Short-Term Memory units (LSTM), a type of recurrent neural network, is

found to be the most suitable architecture. FFT of signals is a sequence, and recurrent neural networks are the most suitable architecture for this feature. Such layers enable the detection system to understand the pattern across the FFT spectrum and help identify remote or small damages for different sensor plannings by looking at the whole spectrum. The results on two benchmark datasets showed this by acquiring the same or even slightly better accuracy in novelty detection with three and two sensors in place of the initially installed 15 and 10 sensors.

This work sets the foundations for a general unsupervised real-time SHM approach utilizing deep learning's recent and novel discoveries, such as GAN and LSTM layers. The method is shown to be capable of addressing several existing obstacles in SHM. The proposed technique is practiced on different structures, such as a bridge, a frame, and a stadium model. Those models have damage on different scales, from a lowered pier in a bridge to loosened bolts in the stadium model. Yet, there are limitations and possible research routes for further investigations. The essence of all developed methods was GAN, a deep generative model. As outlined and studied in Chapter 5, a significant complication with GAN models is their training stability, with some known issues such as mode-collapse. Consequently, GANs could be replaced with more recent developments in the AI field per generative models, such as deep Diffusion models. Diffusion models are more stable in training and outperformed GAN's generative power in recent investigations. Yet, it is to be mentioned that GAN not only does generate data but also identifies novelties with its discriminator. Yet, the diffusion model can only generate data, and developing simulation models might aid novelty detection with them.

The focus of this dissertation is on damage *detection*. Consequently, the method is unable to perform damage localization at this point. While driving a general technique, one aspect was to make the method insensitive to sensor location, which hinders damage localization.

Yet, a novel schema that separately monitors each channel's output is a possible way to add this capability to the method in future works. Additionally, publicly available SHM datasets have two main characteristics. First, acquisition times are very limited, and second, residual damages, both in the format of environmental changes or gradual defects, are missing. Long-term monitoring data with gradual damages and environmental variations are required further to explore the developed framework's capacities and shortcomings. Building such datasets is the other venue of research.

Bibliography

[1] Artemis modal, Aug 2020. URL https://svibs.com/.

[2] Osama Abdeljaber, Onur Avci, Serkan Kiranyaz, Moncef Gabbouj, and Daniel J Inman. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388:154–170, 2017. URL https://doi.org/10.1016/j.jsv.2016.10.043.

[3] Osama Abdeljaber, Onur Avci, Mustafa Serkan Kiranyaz, Boualem Boashash, Henry Sodano, and Daniel J Inman. 1-d cnns for structural damage detection: Verification on a structural health monitoring benchmark data. *Neurocomputing*, 275: 1308–1317, 2018. URL https://www.sciencedirect.com/science/article/pii/S0925231217315886.

[4] Saeid Allahdadian. *Robust statistical subspace-based damage assessment*. PhD thesis, University of British Columbia, 2017. URL http://hdl.handle.net/2429/62148.

[5] JP Amezquita-Sanchez, M Valtierra-Rodriguez, and H Adeli. Wireless smart sensors for monitoring the health condition of civil infrastructure. *Scientia Iranica*, 25(6): 2913–2925, 2018. URL https://dx.doi.org/10.24200/sci.2018.21136.

[6] Juan P Amezquita-Sanchez and Hojjat Adeli. Synchrosqueezed wavelet transform-

fractality model for locating, detecting, and quantifying damage in smart highrise building structures. *Smart Materials and Structures*, 24(6):065034, 2015.

[7] Juan P Amezquita-Sanchez, Hyo Seon Park, and Hojjat Adeli. A novel methodology for modal parameters identification of large smart structures using music, empirical wavelet transform, and hilbert transform. *Engineering Structures*, 147: 148–159, 2017. URL https://www.sciencedirect.com/science/article/pii/S0141029617300445.

[8] Juan Pablo Amezquita-Sanchez and H Adeli. Nonlinear measurements for feature extraction in structural health monitoring. *Scientia Iranica*, 26(6):3051–3059, 2019. URL https://doi.org/10.24200/SCI.2019.21669.

[9] Juan Pablo Amezquita-Sanchez and Hojjat Adeli. Signal processing techniques for vibration-based health monitoring of smart structures. *Archives of Computational Methods in Engineering*, 23(1):1–15, 2016. URL https://doi.org/10.1007/s11831-014-9135-7.

[10] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017. URL https://doi.org/10.48550/arXiv.1711.04340.

[11] Monica Arul, Ahsan Kareem, and Dae Kun Kwon. Identification of vortex-induced vibration of tall building pinnacle using cluster analysis for fatigue evaluation: Application to burj khalifa. *Journal of Structural Engineering*, 146(11):04020234, 2020. doi: 10.1061/(ASCE)ST.1943-541X.0002799. URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29ST.1943-541X.0002799.

[12] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, and Daniel J. Inman. Wireless and real-time structural damage detection: A novel decentralized

method for wireless sensor networks. *Journal of Sound and Vibration*, 424:158–172, 2018. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2018.03.008. URL https://www.sciencedirect.com/science/article/pii/S0022460X18301792.

[13] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, Moncef Gabbouj, and Daniel J. Inman. A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mechanical Systems and Signal Processing*, 147:107077, 2021. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2020.107077. URL https://www.sciencedirect.com/science/article/pii/S0888327020304635.

[14] Shaeela Ayesha, Muhammad Kashif Hanif, and Ramzan Talib. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, 59:44–58, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2020.01.005. URL https://www.sciencedirect.com/science/article/pii/S156625351930377X.

[15] Mohsen Azimi and Gokhan Pekcan. Structural health monitoring using extremely compressed data through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 35(6):597–614, 2020. doi: https://doi.org/10.1111/mice.12517. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12517.

[16] Yuequan Bao, Zhiyi Tang, Hui Li, and Yufeng Zhang. Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, 18(2):401–421, 2019. doi: 10.1177/1475921718757405. URL https://doi.org/10.1177/1475921718757405.

[17] M. Basseville, L. Mevel, and M. Goursat. Statistical model-based damage detection and localization: subspace-based residuals and damage-to-noise sensitivity ra-

tios. *Journal of Sound and Vibration*, 275(3):769–794, 2004. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2003.07.016. URL https://www.sciencedirect.com/science/article/pii/S0022460X03009556.

[18] G. Bernagozzi, C.E. Ventura, S. Allahdadian, Y. Kaya, L. Landi, and P.P. Diotallevi. Application of modal flexibility-based deflections for damage diagnosis of a steel frame structure. *Procedia Engineering*, 199:2026–2033, 2017. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2017.09.468. URL https://www.sciencedirect.com/science/article/pii/S1877705817339620.

[19] Giacomo Bernagozzi, Carlos E. Ventura, Saeid Allahdadian, Yavuz Kaya, Luca Landi, and Pier Paolo Diotallevi. Output-only damage diagnosis for plan-symmetric buildings with asymmetric damage using modal flexibility-based deflections. *Engineering Structures*, 207:110015, 2020. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2019.110015. URL https://www.sciencedirect.com/science/article/pii/S0141029619314476.

[20] Abd Ennour Bouzenad, Mahjoub El Mountassir, Slah Yaacoubi, Fethi Dahmene, Mahmoud Koabaz, Lilian Buchheit, Weina Ke, et al. A semi-supervised based k-means algorithm for optimal guided waves structural health monitoring: A case study. *Inventions*, 4(1):17, 2019. URL https://doi.org/10.3390/inventions4010017.

[21] Rune Brincker and Carlos Ventura. *Introduction to operational modal analysis*. John Wiley & Sons, 2015. URL https://doi.org/10.1002/9781118535141.

[22] Rune Brincker, Palle Andersen, and Lingmi Zhang. Modal identification and damage detection on a concrete highway bridge by frequency domain decomposition. In *Proceedings of Structural Engineering World Conference (SEWC), Yokohama, Japan, 2002*, 2002.

[23] L.A. Bull, T.J. Rogers, C. Wickramarachchi, E.J. Cross, K. Worden, and N. Dervilis. Probabilistic active learning: An online framework for structural health monitoring. *Mechanical Systems and Signal Processing*, 134:106294, 2019. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2019.106294. URL https://www.sciencedirect.com/science/article/pii/S0888327019305096.

[24] L.A. Bull, K. Worden, and N. Dervilis. Towards semi-supervised and probabilistic classification in structural health monitoring. *Mechanical Systems and Signal Processing*, 140:106653, 2020. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2020.106653. URL https://www.sciencedirect.com/science/article/pii/S088832702030039X.

[25] Francesco Cadini, Luca Lomazzi, Marc Ferrater Roca, Claudio Sbarufatti, and Marco Giglio. Neutralization of temperature effects in damage diagnosis of mdof systems by combinations of autoencoders and particle filters. *Mechanical Systems and Signal Processing*, 162:108048, 2022. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2021.108048. URL https://www.sciencedirect.com/science/article/pii/S0888327021004386.

[26] Minwoo Chang and Shamim N Pakzad. Optimal sensor placement for modal identification of bridge systems considering number of sensing nodes. *Journal of Bridge Engineering*, 19(6):04014019, 2014.

[27] Chih-Shen Cheng, Amir H Behzadan, and Arash Noshadravan. Deep learning for post-hurricane aerial damage assessment of buildings. *Computer-Aided Civil and Infrastructure Engineering*, 36(6):695–710, 2021.

[28] Pang-jo Chun, Shota Izumi, and Tatsuro Yamane. Automatic detection method of

cracks from concrete surface imagery using two-step light gradient boosting machine. *Computer-Aided Civil and Infrastructure Engineering*, 36(1):61–72, 2021.

[29] A Cruz, W Vélez, and Peter Thomson. Optimal sensor placement for modal identification of structures using genetic algorithms—a case study: the olympic stadium in cali, colombia. *Annals of Operations Research*, 181(1):769–781, 2010. URL https://doi.org/10.1007/s10479-009-0576-6.

[30] Rharã de Almeida Cardoso, Alexandre Cury, and Flavio Barbosa. Automated real-time damage detection strategy using raw dynamic measurements. *Engineering Structures*, 196:109364, 2019. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2019.109364. URL https://www.sciencedirect.com/science/article/pii/S0141029618340082.

[31] Guido De Roeck. Model–based methods of damage identification of structures under seismic excitation. In *Seismic Structural Health Monitoring*, pages 237–259. Springer, 2019. URL https://doi.org/10.1007/978-3-030-13976-6_10.

[32] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification (pt. 1)*, volume 2. Wiley-Interscience, 2000.

[33] Ricardo Dunia, S. Joe Qin, Thomas F. Edgar, and Thomas J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42 (10):2797–2812, 1996. doi: https://doi.org/10.1002/aic.690421011. URL https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690421011.

[34] Kareem A. Eltouny and Xiao Liang. Bayesian-optimized unsupervised learning approach for structural damage detection. *Computer-Aided Civil and Infrastructure Engineering*, 36(10):1249–1269, 2021. doi: https://doi.org/10.1111/mice.12680. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12680.

[35] Alireza Entezami and Hashem Shariatmadar. An unsupervised learning approach by novel damage indices in structural health monitoring for damage localization and quantification. *Structural Health Monitoring*, 17(2):325–345, 2018. URL https://doi.org/10.1177/147592171769357.

[36] Michael I Friswell and Daniel J Inman. Sensor validation for smart structures. *Journal of intelligent material systems and structures*, 10(12):973–982, 1999. URL https://doi.org/10.1106/GVD2-EGPN-C5B1-DPNX.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. volume 63, pages 139–144. ACM New York, NY, USA, 2020. URL https://doi.org/10.1145/3422622.

[38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. volume 63, pages 139–144. ACM New York, NY, USA, 2020. URL https://doi.org/10.1145/3422622.

[39] Meisam Gordan, Hashim Abdul Razak, Zubaidah Ismail, Khaled Ghaedi, Zhi Xin Tan, and Haider Hamad Ghayeb. A hybrid ann-based imperial competitive algorithm methodology for structural damage identification of slab-on-girder bridge using data mining. *Applied Soft Computing*, 88:106013, 2020. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2019.106013. URL https://www.sciencedirect.com/science/article/pii/S1568494619307951.

[40] Nur Sila Gulgec, Martin Takáč, and Shamim N. Pakzad. Structural sensing with deep learning: Strain estimation from acceleration data for fatigue assessment. *Computer-Aided Civil and Infrastructure Engineering*, 35(12):1349–1364, 2020. doi: https:

//doi.org/10.1111/mice.12565. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12565.

[41] Jingjing Guo, Qian Wang, and Yiting Li. Semi-supervised learning based on convolutional neural network and uncertainty filter for façade defects classification. *Computer-Aided Civil and Infrastructure Engineering*, 36(3):302–317, 2021.

[42] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

[43] Oussama Hattab, Majdi Chaari, Matthew A. Franchek, and Taoufik Wassar. An adaptive modeling approach to structural health monitoring of multistory buildings. *Journal of Sound and Vibration*, 440:239–255, 2019. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2018.10.019. URL https://www.sciencedirect.com/science/article/pii/S0022460X18306928.

[44] Fangqiao Hu, Jin Zhao, Yong Huang, and Hui Li. Structure-aware 3d reconstruction for cable-stayed bridges: A learning-based method. *Computer-Aided Civil and Infrastructure Engineering*, 36(1):89–108, 2021. doi: https://doi.org/10.1111/mice.12568. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12568.

[45] XG Hua, YQ Ni, ZQ Chen, and JM Ko. Structural damage detection of cable-stayed bridges using changes in cable forces and model updating. *Jour-*

nal of structural engineering, 135(9):1093–1106, 2009. doi: https://doi.org/10.1061/(ASCE)0733-9445(2009)135:9(1093). URL https://doi.org/10.1061/(ASCE)0733-9445(2009)135:9(1093).

[46] ISO Central Secretary. Iso 2394: 2015. general principles on reliability for structures. Standard, International Organization for Standardization, Geneva, Switzerland, 20165.

[47] Kejie Jiang, Qiang Han, Xiuli Du, and Pinghe Ni. A decentralized unsupervised structural condition diagnosis approach using deep auto-encoders. *Computer-Aided Civil and Infrastructure Engineering*, 36(6):711–732, 2021. doi: https://doi.org/10.1111/mice.12641. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12641.

[48] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. 33: 12104–12114, 2020. URL https://proceedings.neurips.cc/paper/2020/file/8d30aa96e72440759f74bd2306c1fa3d-Paper.pdf.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. doi: https://doi.org/10.48550/arXiv.1412.6980. URL https://doi.org/10.48550/arXiv.1412.6980.

[50] Rocco Langone, Edwin Reynders, Siamak Mehrkanoon, and Johan A.K. Suykens. Automated structural health monitoring based on adaptive kernel spectral clustering. *Mechanical Systems and Signal Processing*, 90:64–78, 2017. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2016.12.002. URL https://www.sciencedirect.com/science/article/pii/S0888327016305131.

[51] Hyunseong Lee, Hyung Jin Lim, and Aditi Chattopadhyay. Data-driven system health monitoring technique using autoencoder for the safety management of

commercial aircraft. *Neural Computing and Applications*, 33(8):3235–3250, 2021. doi: https://doi.org/10.1007/s00521-020-05186-x. URL https://doi.org/10.1007/s00521-020-05186-x.

[52] Xiaoming Lei, Limin Sun, and Ye Xia. Lost data reconstruction for structural health monitoring using deep convolutional generative adversarial networks. *Structural Health Monitoring*, page 1475921720959226, 2020. doi: https://doi.org/10.1177/1475921720959226. URL https://doi.org/10.1177/1475921720959226.

[53] Yui Chun Leung. *Matlab-GAN*. https://github.com/zcemycl/Matlab-GAN, 2020.

[54] Hui Li and Jinping Ou. The state of the art in structural health monitoring of cable-stayed bridges. *Journal of Civil Structural Health Monitoring*, 6(1):43–67, 2016. doi: https://doi.org/10.1007/s13349-015-0115-x. URL https://doi.org/10.1007/s13349-015-0115-x.

[55] Zhijun Li, Hyo Seon Park, and Hojjat Adeli. New method for modal identification of super high-rise building structures using discretized synchrosqueezed wavelet and hilbert transforms. *The Structural Design of Tall and Special Buildings*, 26(3):e1312, 2017.

[56] Jason Lines, Luke M Davis, Jon Hills, and Anthony Bagnall. A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–297, 2012.

[57] Jingxiao Liu, Siheng Chen, Mario Bergés, Jacobo Bielak, James H Garrett, Jelena Kovačević, and Hae Young Noh. Diagnosis algorithms for indirect structural health monitoring of a bridge model via dimensionality reduction. *Mechanical Systems and Signal Processing*, 136:106454, 2020.

[58] Weizhou Liu, Long Cheng, and Deyuan Meng. Brain slices microscopic detection using simplified ssd with cycle-gan data augmentation. In *International Conference on Neural Information Processing*, pages 454–463. Springer, 2018.

[59] Xirui Ma, Yizhou Lin, Zhenhua Nie, and Hongwei Ma. Structural damage identification based on unsupervised feature-extraction via variational auto-encoder. *Measurement*, page 107811, 2020.

[60] Soroush Mahjoubi, Rojyar Barhemat, and Yi Bao. Optimal placement of triaxial accelerometers using hypotrochoid spiral optimization algorithm for automated monitoring of high-rise buildings. *Automation in Construction*, 118:103273, 2020. ISSN 0926-5805. doi: https://doi.org/10.1016/j.autcon.2020.103273. URL https://www.sciencedirect.com/science/article/pii/S0926580519314372.

[61] Jianxiao Mao, Hao Wang, and Billie F Spencer Jr. Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders. *Structural Health Monitoring*, 20(4):1609–1626, 2021. doi: https://doi.org/10.1177/14759217209246. URL https://doi.org/10.1177/14759217209246.

[62] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[63] A Mendler, CE Ventura, and S Allahdadian. The yellow frame: experimental studies and remote monitoring of the structural health monitoring benchmark structure. In *Special Topics in Structural Dynamics, Volume 5*, pages 233–244. Springer, 2019. doi: https://doi.org/10.1007/978-3-319-75390-4_21. URL https://doi.org/10.1007/978-3-319-75390-4_21.

[64] Alexander Mendler, Michael Döhler, and Carlos E. Ventura. A reliability-based approach to determine the minimum detectable damage for statistical damage detection. *Mechanical Systems and Signal Processing*, 154:107561, 2021. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2020.107561. URL https://www.sciencedirect.com/science/article/pii/S088832702030947X.

[65] Esequiel Mesquita, Paulo Antunes, Francisco Coelho, Paulo André, António Arêde, and Humberto Varum. Global overview on advances in structural health monitoring platforms. *Journal of Civil Structural Health Monitoring*, 6(3):461–475, 2016. doi: https://doi.org/10.1007/s13349-016-0184-5. URL https://doi.org/10.1007/s13349-016-0184-5.

[66] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950. doi: https://doi.org/10.1073/pnas.36.1.48. URL https://doi.org/10.1073/pnas.36.1.48.

[67] Viet Ha Nguyen and Jean-Claude Golinval. Fault detection based on kernel principal component analysis. *Engineering Structures*, 32(11):3683–3691, 2010. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2010.08.012. URL https://www.sciencedirect.com/science/article/pii/S0141029610003019.

[68] Byung Kwan Oh, Kyu Jin Kim, Yousok Kim, Hyo Seon Park, and Hojjat Adeli. Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings. *Applied Soft Computing*, 58:576–585, 2017. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2017.05.029. URL https://www.sciencedirect.com/science/article/pii/S1568494617302922.

[69] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary

DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[70] Chathurdara Sri Nadith Pathirage, Jun Li, Ling Li, Hong Hao, Wanquan Liu, and Pinghe Ni. Structural damage identification based on autoencoder neural networks and deep learning. *Engineering Structures*, 172:13–28, 2018. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2018.05.109. URL https://www.sciencedirect.com/science/article/pii/S0141029618302062.

[71] Bart Peeters and Guido De Roeck. One-year monitoring of the z24-bridge: environmental effects versus damage events. *Earthquake engineering & structural dynamics*, 30(2):149–171, 2001. doi: https://doi.org/10.1002/1096-9845(200102)30:2<149::AID-EQE1>3.0.CO;2-Z. URL https://doi.org/10.1002/1096-9845(200102)30:2<149::AID-EQE1>3.0.CO;2-Z.

[72] Carlos A. Perez-Ramirez, Juan P. Amezquita-Sanchez, Martin Valtierra-Rodriguez, Hojjat Adeli, Aurelio Dominguez-Gonzalez, and Rene J. Romero-Troncoso. Recurrent neural network model with bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, 178:603–615, 2019. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2018.10.065. URL https://www.sciencedirect.com/science/article/pii/S0141029618307235.

[73] Carlos Andres Perez-Ramirez, Juan Pablo Amezquita-Sanchez, Hojjat Adeli, Martin Valtierra-Rodriguez, Rene de Jesus Romero-Troncoso, Aurelio Dominguez-Gonzalez, and Roque Alfredo Osornio-Rios. Time-frequency techniques for modal parameters identification of civil structures from acquired dynamic signals. *Journal of Vibroengineering*, 18(5):3164–3185, 2016. doi: https://doi.org/10.21595/jve.2016.17220. URL https://doi.org/10.21595/jve.2016.17220.

[74] S. Joe Qin and Weihua Li. Detection, identification, and reconstruction of faulty sensors with maximized sensitivity. *AIChE Journal*, 45(9):1963–1976, 1999. doi: https://doi.org/10.1002/aic.690450913. URL https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690450913.

[75] Mohammad Hossein Rafiei and Hojjat Adeli. A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, 156:598–607, 2018. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2017.10.070. URL https://www.sciencedirect.com/science/article/pii/S0141029617323155.

[76] Mohammad Hossein Rafiei, Waleed H Khushefati, Ramazan Demirboga, and Hojjat Adeli. Supervised deep restricted boltzmann machine for estimation of concrete. *ACI Materials Journal*, 114(2), 2017. doi: https://doi.org/10.14359/51689560. URL https://doi.org/10.14359/51689560.

[77] Edwin Reynders and Guido De Roeck. Vibration-based damage identification: the z24 benchmark. 2014. URL https://lirias.kuleuven.be/1725994?limo=0.

[78] Edwin Reynders and Guido De Roeck. Continuous vibration monitoring and progressive damage testing on the z 24 bridge. *Encyclopedia of structural health monitoring*, 2009. doi: https://doi.org/10.1002/9780470061626.shm165. URL https://doi.org/10.1002/9780470061626.shm165.

[79] Davi VQ Rodrigues, Ziyan Tang, Jing Wang, Delong Zuo, and Changzhi Li. Structural health monitoring of a traffic signal support structure based on 5.8-ghz doppler radar with median filter and revised circle fitting. In *2020 IEEE Radio and Wireless Symposium (RWS)*, pages 187–190. IEEE, 2020. doi: 10.1109/RWS45077.2020.9050044. URL 10.1109/RWS45077.2020.9050044.

[80] TJ Rogers, K Worden, R Fuentes, N Dervilis, UT Tygesen, and EJ Cross. A bayesian non-parametric clustering approach for semi-supervised structural health monitoring. *Mechanical Systems and Signal Processing*, 119:100–119, 2019. URL https://doi.org/10.48550/arXiv.1701.00160.

[81] Seyed Omid Sajedi and Xiao Liang. Vibration-based semantic damage segmentation for large-scale structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 35(6):579–596, 2020. doi: https://doi.org/10.1111/mice.12523. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12523.

[82] Seyedomid Sajedi and Xiao Liang. Dual bayesian inference for risk-informed vibration-based damage diagnosis. *Computer-Aided Civil and Infrastructure Engineering*, 36(9): 1168–1184, 2021. doi: https://doi.org/10.1111/mice.12642. URL https://doi.org/10.1111/mice.12642.

[83] Addisson Salazar, Luis Vergara, and Gonzalo Safont. Generative adversarial networks and markov random fields for oversampling very small training sets. *Expert Systems with Applications*, 163:113819, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2020.113819. URL https://www.sciencedirect.com/science/article/pii/S095741742030631X.

[84] João P Santos, André D Orcesi, Christian Crémona, and Paulo Silveira. Baseline-free real-time assessment of structural changes. *Structure and Infrastructure Engineering*, 11(2):145–161, 2015. doi: https://doi.org/10.1080/15732479.2013.858169. URL https://doi.org/10.1080/15732479.2013.858169.

[85] Hassan Sarmadi and Alireza Entezami. Application of supervised learning to validation of damage detection. *Archive of Applied Mechanics*, 91(1):393–410, 2021.

doi: https://doi.org/10.1007/s00419-020-01779-z. URL https://doi.org/10.1007/s00419-020-01779-z.

[86] Hassan Sarmadi and Abbas Karamodin. A novel anomaly detection method based on adaptive mahalanobis-squared distance and one-class knn rule for structural health monitoring under environmental effects. *Mechanical Systems and Signal Processing*, 140:106495, 2020. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2019.106495. URL https://www.sciencedirect.com/science/article/pii/S0888327019307162.

[87] Hassan Sarmadi and Ka-Veng Yuen. Early damage detection by an innovative unsupervised learning method based on kernel null space and peak-over-threshold. *Computer-Aided Civil and Infrastructure Engineering*, 36(9):1150–1167, 2021. doi: https://doi.org/10.1111/mice.12635. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12635.

[88] Carmelo Scuro, Paolo Francesco Sciammarella, Francesco Lamonaca, Renato Sante Olivito, and Domenico Luca Carni. Iot for structural health monitoring. *IEEE Instrumentation & Measurement Magazine*, 21(6):4–14, 2018. doi: 10.1109/MIM.2018.8573586. URL 10.1109/MIM.2018.8573586.

[89] Zhiqiang Shang, Limin Sun, Ye Xia, and Wei Zhang. Vibration-based damage detection for bridges by deep convolutional denoising autoencoder. *Structural Health Monitoring*, page 1475921720942836, 2020. doi: https://doi.org/10.1177/1475921720942836. URL https://doi.org/10.1177/1475921720942836.

[90] Priyanka Singh, Umaid Faraz Ahmad, and Siddharth Yadav. Structural health monitoring and damage detection through machine learning approaches. In *E3S Web of Conferences*, volume 220, page 01096. EDP Sciences, 2020.

[91] Mohammad Hesam Soleimani-Babakamali, Roksana Soleimani-Babakamali, and Rodrigo Sarlo. A general framework for supervised structural health monitoring and sensor output validation mitigating data imbalance with generative adversarial networks-generated high-dimensional features. *Structural Health Monitoring*, page 14759217211025488, 2021. doi: https://doi.org/10.1177/14759217211025488. URL https://doi.org/10.1177/14759217211025488.

[92] Mohammad Hesam Soleimani-Babakamali, Reza Sepasdar, Kourosh Nasrollahzadeh, and Rodrigo Sarlo. A system reliability approach to real-time unsupervised structural health monitoring without prior information. *Mechanical Systems and Signal Processing*, 171:108913, 2022. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2022.108913. URL https://www.sciencedirect.com/science/article/pii/S0888327022000991.

[93] Mohammad Hesam Soleimani-Babakamali, Reza Sepasdar, Kourosh Nasrollahzadeh, and Rodrigo Sarlo. A system reliability approach to real-time unsupervised structural health monitoring without prior information. *Mechanical Systems and Signal Processing*, 171:108913, 2022. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2022.108913. URL https://www.sciencedirect.com/science/article/pii/S0888327022000991.

[94] William Soo Lon Wah, John S. Owen, Yung-Tsang Chen, Ahmed Elamin, and Gethin Wyn Roberts. Removal of masking effect for damage detection of structures. *Engineering Structures*, 183:646–661, 2019. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2019.01.005. URL https://www.sciencedirect.com/science/article/pii/S0141029618325070.

[95] William Soo Lon Wah, Yung-Tsang Chen, and John S Owen. A regression-based

damage detection method for structures subjected to changing environmental and operational conditions. *Engineering Structures*, 228:111462, 2021. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2020.111462. URL https://www.sciencedirect.com/science/article/pii/S0141029620340633.

[96] Christopher J Stull, Christopher J Earls, and Phaedon-Stelios Koutsourelakis. Model-based structural health monitoring of naval ship hulls. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1137–1149, 2011.

[97] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[98] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[99] Bin Tang, Ya Tu, Zhaoyue Zhang, and Yun Lin. Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks. *IEEE Access*, 6:15713–15722, 2018. doi: https://doi.org/10.1109/ACCESS.2018.2815741. URL https://doi.org/10.1109/ACCESS.2018.2815741.

[100] Zhiyi Tang, Zhicheng Chen, Yuequan Bao, and Hui Li. Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Structural Control and Health Monitoring*, 26(1):e2296, 2019. doi: https://doi.org/10.1002/stc.2296. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2296.

[101] **Mohammad Hesam Soleimani-Babakamali**, Ismini Lourentzou, and Rodrigo Sarlo. Does the curse of dimensionality apply to unsupervised shm? investigating

the trade-off between loss of information and generalizability to unseen structural conditions. In *Proceedings of the Thirteenth International Workshop on Structural Health Monitoring (IWSHM), March 15-17, 2022 (formerly December 7-9. 2021)*, pages 644–651. DEStech Publications Ltd., 15-17 Mar 2022, Stanford, CA, USA.

[102] **Mohammad Hesam Soleimani-Babakamali**, Roksana Soleimani-Babakamali, and Rodrigo Sarlo. A general framework for supervised structural health monitoring and sensor output validation mitigating data imbalance with generative adversarial networks-generated high-dimensional features. *Structural Health Monitoring*, page 14759217211025488, 2021. doi: https://doi.org/10.1177/14759217211025488. URL https://doi.org/10.1177/14759217211025488.

[103] **Mohammad Hesam Soleimani-Babakamali**, Reza Sepasdar, Kourosh Nasrollahzadeh, Ismini Lourentzou, and Rodrigo Sarlo. Toward a general unsupervised novelty detection framework in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 2022. doi: https://doi.org/10.1111/mice.12812. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12812.

[104] Sarehati Umar, Norhisham Bakhary, and A.R.Z. Abidin. Response surface methodology for damage detection using frequency and mode shape. *Measurement*, 115:258–268, 2018. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2017.10.047. URL https://www.sciencedirect.com/science/article/pii/S0263224117306796.

[105] Furui Wang, Gangbing Song, and Yi-Lung Mo. Shear loading detection of through bolts in bridge structures using a percussion-based one-dimensional memory-augmented convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, 36(3):289–301, 2021. doi: https://doi.org/10.1111/mice.12602. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12602.

[106] Niannian Wang, Xuefeng Zhao, Zheng Zou, Peng Zhao, and Fei Qi. Autonomous damage segmentation and measurement of glazed tiles in historic buildings via deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 35(3):277–291, 2020. doi: https://doi.org/10.1111/mice.12488. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12488.

[107] Zilong Wang and Young-Jin Cha. Unsupervised deep learning approach using a deep auto-encoder with an one-class support vector machine to detect structural damage. *Structural Health Monitoring*, page 1475921720934051, 2020. doi: https://doi.org/10.1177/1475921720934051. URL https://doi.org/10.1177/1475921720934051.

[108] Keith Worden, Charles R Farrar, Graeme Manson, and Gyuhae Park. The fundamental axioms of structural health monitoring. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(April):1639–1664, 2007. ISSN 1364-5021. doi: https://doi.org/10.1098/rspa.2007.1834. URL https://doi.org/10.1098/rspa.2007.1834.

[109] Ai-Min Yan and Jean-Claude Golinval. Null subspace-based damage detection of structures using vibration measurements. *Mechanical Systems and Signal Processing*, 20(3):611–626, 2006. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2005.04.010. URL https://www.sciencedirect.com/science/article/pii/S0888327005000798.

[110] Wang-Ji Yan, Dimitrios Chronopoulos, Ka-Veng Yuen, and Yi-Chen Zhu. Structural anomaly detection based on probabilistic distance measures of transmissibility function and statistical threshold selection scheme. *Mechanical Systems and Signal Processing*, 162:108009, 2022. ISSN 0888-3270. doi: https://doi.org/10.1016/

j.ymssp.2021.108009. URL https://www.sciencedirect.com/science/article/pii/S0888327021004040.

[111] Xiaoan Yan, Ying Liu, and Minping Jia. Health condition identification for rolling bearing using a multi-domain indicator-based optimized stacked denoising autoencoder. *Structural Health Monitoring*, 19(5):1602–1626, 2020. doi: https://doi.org/10.1177/1475921719893594. URL https://doi.org/10.1177/1475921719893594.

[112] Ting-Hua Yi, Hong-Nan Li, and Ming Gu. Optimal sensor placement for health monitoring of high-rise structure based on genetic algorithm. *Mathematical Problems in Engineering*, 2011, 2011. URL https://doi.org/10.1155/2011/395101.

[113] Ting-Hua Yi, Hai-Bin Huang, and Hong-Nan Li. Development of sensor validation methodologies for structural health monitoring: A comprehensive review. *Measurement*, 109:200–214, 2017. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2017.05.064. URL https://www.sciencedirect.com/science/article/pii/S0263224117303718.

[114] Ling Yu, Jun-hua Zhu, and Liu-jie Cheri. Parametric study on pca-based algorithm for structural health monitoring. In *2010 Prognostics and System Health Management Conference*, pages 1–6. IEEE, 2010. doi: 10.1109/PHM.2010.5413428. URL 10.1109/PHM.2010.5413428.

[115] Yang Yu, Chaoyue Wang, Xiaoyu Gu, and Jianchun Li. A novel deep learning-based method for damage identification of smart building structures. *Structural Health Monitoring*, 18(1):143–163, 2019. doi: https://doi.org/10.1177/1475921718804132. URL https://doi.org/10.1177/1475921718804132.

[116] Masoumeh Zareapoor, Pourya Shamsolmoali, and Jie Yang. Oversampling adversarial network for class-imbalanced fault diagnosis. *Mechanical Systems and Sig-*

nal Processing, 149:107175, 2021. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2020.107175. URL https://www.sciencedirect.com/science/article/pii/S0888327020305616.

[117] Shengyuan Zhang, Chun Min Li, and Wenjing Ye. Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network. *Mechanical Systems and Signal Processing*, 147:107107, 2021. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2020.107107. URL https://www.sciencedirect.com/science/article/pii/S0888327020304933.

[118] Xun Zhang, Ping Wang, Jian-Chun Xing, and Qi-Liang Yang. Optimal sensor placement of long-span cable-stayed bridges based on particle swarm optimization algorithm. In *Practical applications of intelligent systems*, pages 207–217. Springer, 2014. URL https://doi.org/10.1007/978-3-642-54927-4_20.

[119] Xueyan Zhao and Ziqiang Lang. Baseline model based structural health monitoring method under varying environment. *Renewable Energy*, 138:1166–1175, 2019. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2019.02.007. URL https://www.sciencedirect.com/science/article/pii/S096014811930151X.

[120] Yinan Zhao, Maosheng Gong, Zhanxuan Zuo, and Yanbin Gao. Bayesian estimation approach based on modified scam algorithm and its application in structural damage identification. *Structural Control and Health Monitoring*, 28(1):e2654, 2021. doi: https://doi.org/10.1002/stc.2654. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2654. e2654 STC-19-0451.R2.

Appendices

Chapter A:

First Appendix

A.1 Codes

The codes for the manuscript "Towards a general unsupervised novelty detection framework in structural health monitoring" are available in https://github.com/Hesam-92-19/CAVAIE_Paper_Codes. All codes are written in Google Colaboratory as Jupyter Notebooks.

A.2 Instructions

The GitHub directory contains two Jupyter notebooks. The first notebook regards Sections 5.2 and 5.3, named "Data_Networks_Training_Tuning.ipynb", while the other, "Accuracy_Evaluation.ipynb", is written to offer detection results with the same format as Section 5.4. Notebooks are accompanied with two ".mat" files including features, "F_I.mat", and data classes' size, "C_L.mat".

A.2.1 Data_Networks_Training_Tuning

This notebook, first, mounts the user's Google Drive, making the code's execution as simple as possible for users. Then, it creates a folder with the name the user defined through the variable "Folder_N". Simultaneously, several sub-folders are created to store analytical results obtained in the notebooks. Regarding the public dataset QUGS, available at http://onur-avci.com/benchmark/download/, two ".mat" files, namely "F_I.mat" and "C_L.mat" are provided in the GitHub directory to perform the novelty detection on QUGS. F_I.mat contains the F_I feature (Section 5.3) from QUGS dataset, that has eleven data classes (Section 5.2). For those eleven data classes, features are obtained with $T_L = 1000$. This parameter is fixed since we have no privilege to re-distribute the dataset for the users to opt for different T_L values. Yet, users can build their .mat files (must be stored as -V7.3 .mat file) for F_I and execute the code. Each QUGS data class has 262 data samples with $T_L = 1000$. In this notebook, data from the 2-Sensor scenario (Section 5.3) is implemented; consequently, data tensor in F_I.mat $\in \mathbb{R}^{262*11\times\frac{1000}{2}*2}$ (Fig. A.1). Notebook creates 11 sub-folders in the user's Google Drive, named DC1 to DC11, for the eleven data classes under the "LSTM" folder (Fig. A.2).

Figure A.1: F_I.mat contents.

The other .mat file, C_L.mat, is a one-dimensional array that stores the length of each data class. QUGS, as mentioned, has an equal length of 262 data (*i.e.*, with $T_L = 1000$) for each of the eleven data classes. Hence, 'C_L,mat" contains an array of digit 262 ($\in \mathbb{R}^{11\times1}$), as shown in Fig. A.3. The modular structure of the notebook makes its application to other

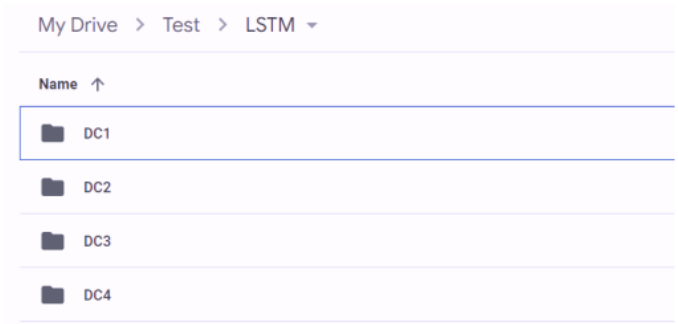


Figure A.2: Google Drive Subfolders made by Data_Networks_Training_Tuning.ipynb.

datasets straightforward, as users can input their data by conforming with the explained dimensionalities.

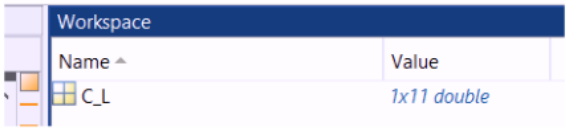


Figure A.3: C_L.mat contents.

Next, the custom LSTM-based $\mathcal{D}$ architecture with three stacked layers is coded under the class "LSTM_Net_2", and $\mathcal{G}$ is coded in that class as "generator". S_L in the code stands for the number of stacked LSTM layers, that is, three herein. For datasets with features of different dimensions, the LSTM network is to be updated. The LSTM network is trained in with the training function "trainLSTM" shown in Fig. A.5. This function has $\mathcal{D}$, $\mathcal{G}$, their solvers (*i.e.*, ADAM optimizers for $\mathcal{D}$ and $\mathcal{G}$), discriminator and generator losses (Section 5.2), epoch-interval to plot and save results, a Boolean with the False value to indicate training on all data classes and vice versa, batch size, P_z size, target epochs, the whole dataset (*i.e.*, F_I.mat), and computation device (*e.g.*, GPU and CPU, defined in the notebook) as inputs, respectively. As outlined in Section 5.3, to speed up the detection process with numerous detection instances (Section 5.3.3), GAN networks are trained on all data classes' training length beforehand. The trained network is called during the detection if a novel data class

is identified. Trained Networks and training losses are saved under data class directories, labeled with the epoch number they were saved in, with an example shown in Fig. A.4.

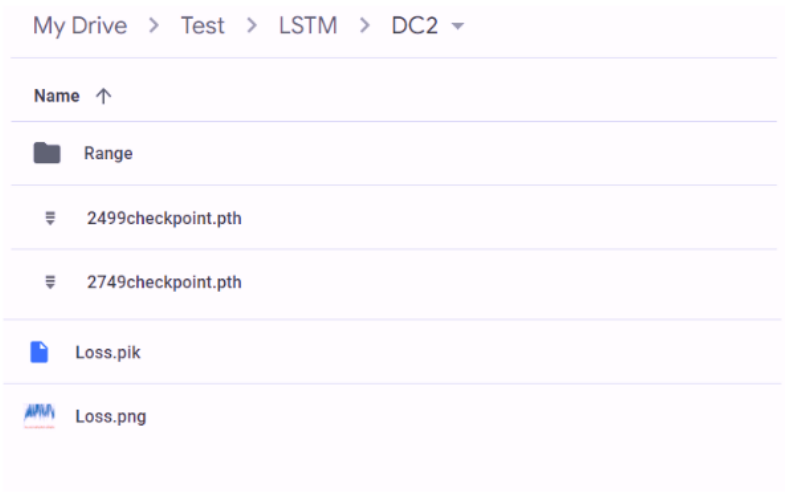


Figure A.4: Examples of GAN's training-associated results saved in DC directories.

```
def trainLSTM(D, G, D_solver, G_solver, discriminator_loss, generator_loss, show_every=10,
          cont_train=False,
          batch_size=100, noise_size=100, num_epochs=30, train_loader=None, device=None,D_C=None):
```

Figure A.5: Training loop function.

As discussed in 5.2, users might favor different loss patterns to select $\mathcal{G}$-$\mathcal{D}$ pairs to build their novelty detector. Training losses are saved as images and matrices at the end of each class's training (Fig. A.4). In Section 5.3, detailed investigations are made on which pattern outperforms others. Based on that, the epoch selection could be automatized by writing additional lines of codes, yet in the current notebook, users can input their epochs of choice as matrix "Range". Three epochs of 10000, 12500, and 15000 are pre-defined values in the notebook, and users can manipulate them by inspecting the loss-related saved data (Fig. A.4). Notebook stores the selected epochs' results under a new directory with the name of the variable "key". For instance, an inner directory is made within DC folders, with the key named "Range", as shown in Fig. A.4.

```
key='Range'
Folder_N='TEST'

Range=np.array([[10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000],
                [10000,12500,15000]])
```

Figure A.6: Pre-defined target epochs.

Threshold selection algorithm, Algorithm 2, is defined through function "Two_GMM". This function also accepts a reliability index, β, to be compatible with the research article explained in Chapter 4. Users can decide to turn that capability on or off. The resulting T values are stored in the "key" sub-folder of data classes' directories. Saved T values are named through "T_1" to "T_R", in which R is the total number of epochs, that is, three in the provided notebook (Fig. A.4).

As explained in Section 5.3, numerous detection instances exist. Thus, to speed up the process, all data instances detections scores, S_{GAN}, and all trained GAN's T (Section 5.2) are stored in separate matrices for each of the epochs, as outlined in Section 5.3. Such a matrix-based operation is enabled via fixing the number of each data class's data instances to be V_L's multiple. For instance, if QUGS has 262 data points and $V_L = 10$, the last two data instances are dropped. The resulting matrices are stored in the "key" sub-folder of each data class's directory as Python Pickle files with names "Data1" to "Data-R" for epochs one to R, respectively (Fig.A.7). With the stored arrays, the performance evaluation occurs within the next Notebook, Accuracy_Evaluation.ipynb.

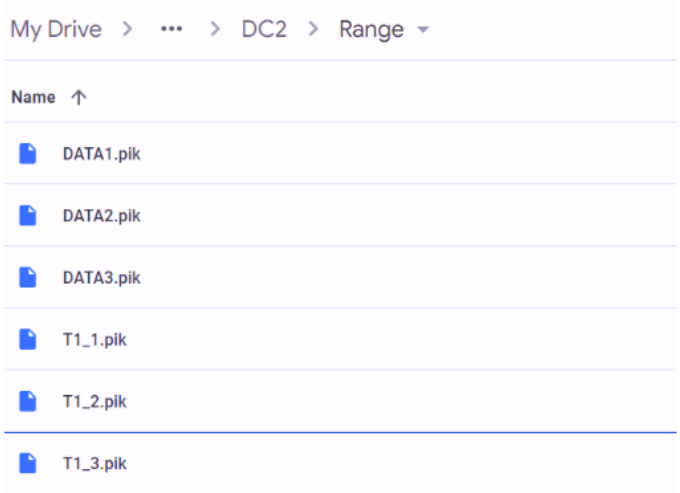


Figure A.7: Stored detection scores and thresholds in the Google Drive directory.

A.2.2 Accuracy_Evaluation

This notebook depicts novelty detection results in two sections. The first section named "Detection Displayer," an instance of novelty detection with the same order of data classes as in Table 5.8, is plotted. As the previous notebook explains, users can select the number of epochs from each data class, and detection scores/thresholds are computed and stored under the "key" folder. Users can also select the Adaptive or Simple thresholding method in this cell. Those epochs are coded in the matrix "E_pochs". That matrix is a one-by-eleven (*i.e.*, QUGS has 11 data classes), and the selected epoch numbers (*i.e.*, 0 to 2 for the current notebook with three selected epochs) are inputted there. If a novelty is identified, the selected epoch from that novelty is chosen to continue the detection; otherwise, the analysis keeps going forward. A sample detection output of this cell is shown in Fig. 5.19.

The second portion of the notebook is to perform the Permutation Detection (Section 5.3.3). Presuming the selected epochs as: A=[1,2,3,3,2,2,3,1,3,2,1], the 11! permutations of all different classes is simplified in a 11-by-10 matrix, in which each element can be either 0 or 1.

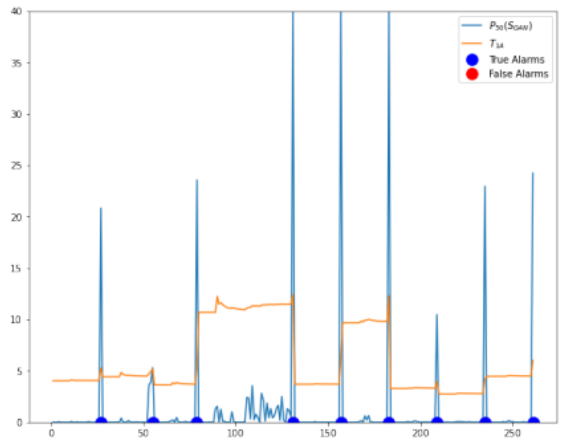


Figure A.8: Sample output of Detection Displayer cell.

0 stands for missed novelties, while 1 indicates the novelty detection took place (Fig. A.9). With this matrix, the whole permutation detection is reduced to detecting two different classes from each other for all possible binary combinations (*i.e.*, 10×11 matrix).

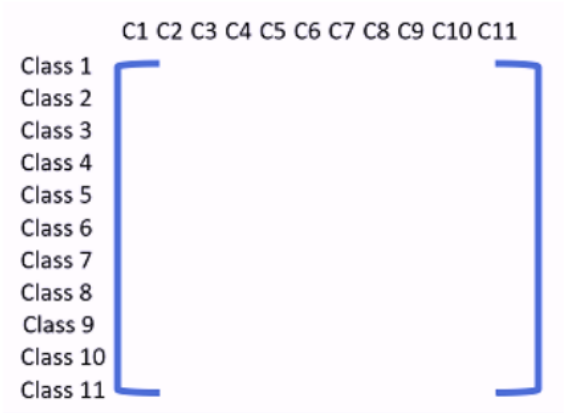


Figure A.9: Permutation detection matrix.

Supposing the completion of the 4th row of that matrix is targeted; first, the array of detection scores obtained from the previous notebook is permuted to have class 4 at the beginning. The GAN detector trained on class 4 is then used to determine whether each of the other classes has detection scores above the threshold. Suppose a class has higher

detection scores than class-4-threshold (e.g., Fig. A.10), the matrix element representing that pair of classes is set as 1. This matrix is built for each array of epochs. In the studied QUGS dataset, there are 11 data classes, and three epochs were selected from each in the previous notebook. Hence, there are $3^1 1$ permutation matrices. The reported accuracy is the mean summation of all matrix elements divided by 110 (*i.e.*, number of elements each matrix has).

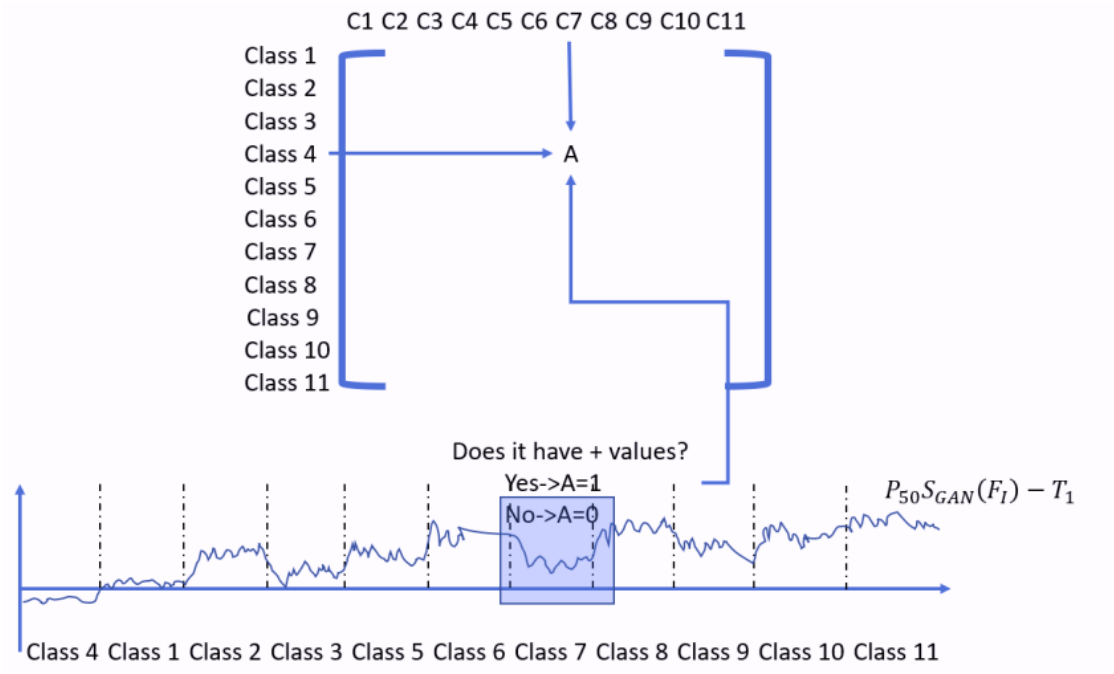


Figure A.10: Filling the permutation detection matrix.

SoleimaniBabakamali_MH_D_2022.pdf

ORIGINALITY REPORT

4%
SIMILARITY INDEX

PRIMARY SOURCES

1	job.x-mol.com Internet	336 words — 1%
2	www.x-mol.com Internet	210 words — < 1%
3	www.researchgate.net Internet	182 words — < 1%
4	vtechworks.lib.vt.edu Internet	112 words — < 1%
5	bwk.kuleuven.be Internet	78 words — < 1%
6	arxiv.org Internet	49 words — < 1%
7	Reza Sepasdar, Anuj Karpatne, Maryam Shakiba. "A data-driven approach to full-field nonlinear stress distribution and failure pattern prediction in composites using deep learning", Computer Methods in Applied Mechanics and Engineering, 2022 Crossref	44 words — < 1%
8	www.extrica.com Internet	44 words — < 1%

9	nvlpubs.nist.gov Internet	37 words — < 1%
10	tel.archives-ouvertes.fr Internet	33 words — < 1%
11	link.springer.com Internet	32 words — < 1%
12	jglobal.jst.go.jp Internet	31 words — < 1%
13	www.coursehero.com Internet	30 words — < 1%
14	www.dtic.mil Internet	22 words — < 1%
15	Yan, Chao. "Vision-Based Driver Behaviour Analysis.", The University of Liverpool (United Kingdom), 2021 ProQuest	21 words — < 1%
16	scholar.smu.edu Internet	20 words — < 1%
17	amsdottorato.unibo.it Internet	19 words — < 1%
18	research-information.bris.ac.uk Internet	19 words — < 1%
19	uis.brage.unit.no Internet	19 words — < 1%
20	ore.exeter.ac.uk Internet	18 words — < 1%

21	autonumerology.com Internet	17 words — < 1%
22	repository.kaust.edu.sa Internet	17 words — < 1%
23	pure.mpg.de Internet	16 words — < 1%
24	Matar, Mohammad G.. "A Bioinspired Approach to Mitigate Freeze-Thaw Damage in Concrete", University of Colorado at Boulder, 2022 ProQuest	15 words — < 1%
25	Wan, Zhiqiang. "Deep Learning for Data-Driven Decision-Making.", University of Rhode Island, 2020 ProQuest	14 words — < 1%
26	essd.copernicus.org Internet	14 words — < 1%
27	linchin.ndmctsgh.edu.tw Internet	12 words — < 1%
28	repositorium.uni-muenster.de Internet	12 words — < 1%
29	res.mdpi.com Internet	12 words — < 1%
30	scholar.dominican.edu Internet	12 words — < 1%
31	documents.mx Internet	11 words — < 1%

32	parasitesandvectors.biomedcentral.com Internet	11 words — < 1%
33	text-id.123dok.com Internet	11 words — < 1%
34	Kim, DongKyu. "Automatic Artifact Annotator for EEG Waves Using Recurrent and Convolutional Neural Networks.", The Cooper Union for the Advancement of Science and Art, 2020 ProQuest	10 words — < 1%
35	Luan, Lele. "Deep Vision for Sensing and Discovery of Nonlinear Dynamics", Northeastern University, 2022 ProQuest	10 words — < 1%
36	Mohamed A. El-Barawany, Hamdy K. Elminir, Hatem M. Elattar, Ebrahim Abdel Hamid, Dina Adel. "Data Driven Prognostics from Machine Learning to Deep Learning: A survey", Research Square Platform LLC, 2022 Crossref Posted Content	10 words — < 1%
37	Zhu, Yizhe. "Advanced Zero-Shot Learning: Object Recognition with Semantic Descriptions", Rutgers The State University of New Jersey, School of Graduate Studies, 2022 ProQuest	10 words — < 1%
38	archive.org Internet	10 words — < 1%
39	etheses.whiterose.ac.uk Internet	10 words — < 1%
40	mdpi-res.com Internet	10 words — < 1%

41	mspace.lib.umanitoba.ca Internet	10 words — < 1%
42	repositories.lib.utexas.edu Internet	10 words — < 1%
43	research-repository.griffith.edu.au Internet	10 words — < 1%
44	spiral.imperial.ac.uk Internet	10 words — < 1%
45	www.arxiv-vanity.com Internet	10 words — < 1%
46	www.ecocyb.ase.ro Internet	10 words — < 1%
47	www.kdhe.ks.gov Internet	10 words — < 1%
48	www.math.kth.se Internet	10 words — < 1%
49	Encyclopedia of Earthquake Engineering, 2015. Crossref	9 words — < 1%
50	Hassan Sarmadi. "Investigation of Machine Learning Methods for Structural Safety Assessment under Variability in Data: Comparative Studies and New Approaches", Journal of Performance of Constructed Facilities, 2021 Crossref	9 words — < 1%
51	Heaton, Howard Wayne. "Learning to Optimize with Guarantees", University of California, Los	9 words — < 1%

Angeles, 2021
ProQuest

52	Wang, Mengying. "Detecting, Identifying, and Reconstructing Fluid Flows.", University of Minnesota, 2021 ProQuest	9 words — < 1%
53	bmcgenomics.biomedcentral.com Internet	9 words — < 1%
54	catalog.uttyler.edu Internet	9 words — < 1%
55	cpansearch.perl.org Internet	9 words — < 1%
56	docshare.tips Internet	9 words — < 1%
57	ir.lib.uwo.ca Internet	9 words — < 1%
58	lnu.diva-portal.org Internet	9 words — < 1%
59	ntnuopen.ntnu.no Internet	9 words — < 1%
60	onlinelibrary.wiley.com Internet	9 words — < 1%
61	research.gold.ac.uk Internet	9 words — < 1%
62	strathprints.strath.ac.uk Internet	9 words — < 1%

63 www.hindawi.com
Internet

9 words — < 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 9 WORDS
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	< 9 WORDS