

POLITECNICO DI TORINO

MASTER's Degree in Mechanical Engineering

MASTER's Degree Thesis



**Politecnico
di Torino**

TITLE:

**Analysis of noise effects with Deep Learning and Structural
Health Monitoring applications**

Supervisors:

Prof. Giuseppe Carlo Marano

Ph.D. Marco Martino Rosso

Ph.D. Angelo Aloisio

Presented by:

Fei HUO

ACADEMIC YEAR 2021/2022

Table of Contents

1. Introduction	2
1.1 Structural Health Monitoring (SHM).....	2
1.2 Big Data and Artificial Intelligence in SHM	4
1.3 Application of Sensors in SHM	6
2. Main Tasks and Research Parameters.....	15
2.1 Numerical Beam Model.....	17
2.2 Selection of Sensor Parameters	17
2.3 Gaussian White Noise	20
2.4 Damage Indicator Methods	23
3. CNN and ANN models.....	27
3.1 Datasets	27
3.2 CNN Model.....	34
3.3 MLP Model.....	37
3.4 Selection of Common Parameters	38
4. Outcomes and Conclusion.....	39
4.1 MLP and CNN Multiclass Classification Results.....	39
4.2 Discussion	44
4.3 Conclusions and Future Remarks.....	50
Reference.....	53

Abstract

Structural Health Monitoring (SHM) has always been a hot topic in the field of Mechanical and Civil Engineering, its' first-level task is damage detection. Traditional Stochastic Subspace Identification (SSI) performs damage detection through operational modal analysis (OMA), which is a data-driven method and has high identification accuracy, but huge amount of data and calculation. Advances in artificial intelligence tools represent the frontier of SHM, enabling non-destructive assessments directly based on the output-only vibration signals. In this thesis, three different features are proposed to deal with multi-class damage classification tasks, the first method relies only on statistical features, in contrast, the second method considers the most informative subspace-based damage indicator. Finally, the third method considers the entire set of damage indicators obtained through a reasonably variable range of input parameters as features. Two Deep Learning methods – Multi-layer Perceptron (MLP) and 1D-convolutional network (CNN) are designed to compare and verify the results. To verify the noise effect on different features and Deep Learning models, three groups of different reasonable Signal-to-Noise Ratio (SNR) levels were set up to simulate MEMS accelerometer sensors in real working condition. The proposed experiments are tested on numerical benchmark problems. The results show that, even in the data collected by real sensors with different noise effects, Deep Learning model and Stochastic Subspace Identification method have good noise immunity performance and good on multiclassification behavior. Subspace-based damage-sensitive features is still the most effective indicator in the field of Structure health monitoring.

Keywords: structural health monitoring, stochastic subspace identification operational modal analysis, subspace-based damage indicators, deep learning, multi-layer perceptron, 1D-Convolutional network, Signal-to-Noise Ratio

1. Introduction

1.1 Structural Health Monitoring (SHM)

Structural health monitoring (SHM) is the process of implementing a damage identification strategy for aerospace, civil, and mechanical engineering infrastructure[1]. It is an interdisciplinary subject that emerged in the 20th century, has achieved rapid development in recent years. According to SAE Standard ARP6461 [2], SHM is defined as ‘The process of acquiring and analyzing data from on-board sensors to evaluate the health of a structure’.

1.1.1 SHM Technology

Traditional Nondestructive Testing (NDT) procedures, such as ultrasonics, X-rays, or thermography, among others, have the main task of detecting cracks and discontinuities within the material or on its surface for quality control of newly manufactured parts and quality assurance during service. Their reliability is well proven. The difference between NDT and SHM technology are obvious. NDT are more mature technologies but require human intervention and therefore have higher labor costs; they require the use of external probes or equipment to access structural components; they are not suitable for condition-based maintenance (CBM) concepts.[3]

The process of predicting the remaining useful life of a structure is closely related to structural health monitoring, which is the main goal of SHM. These two concepts are also defined as Prognosis and Diagnosis. But they are different: prognosis is mainly focused on statistical analysis, while diagnosis is more related to sensors, signal processing, and damage identification algorithms. The SHM problem presents at least five levels of analysis as illustrated in [4]: Level 1, associated with the initial damage detection phase; Level 2, damage localization phase; Level 3, identification of the type of damage; Level 4, quantification of damage severity; Level 5, related to the prediction of the actual remaining service life concerning the nominal one. The first four levels are associated with the well-acknowledged diagnosis phase[5], whereas the last one is related to the prognosis phase[6].

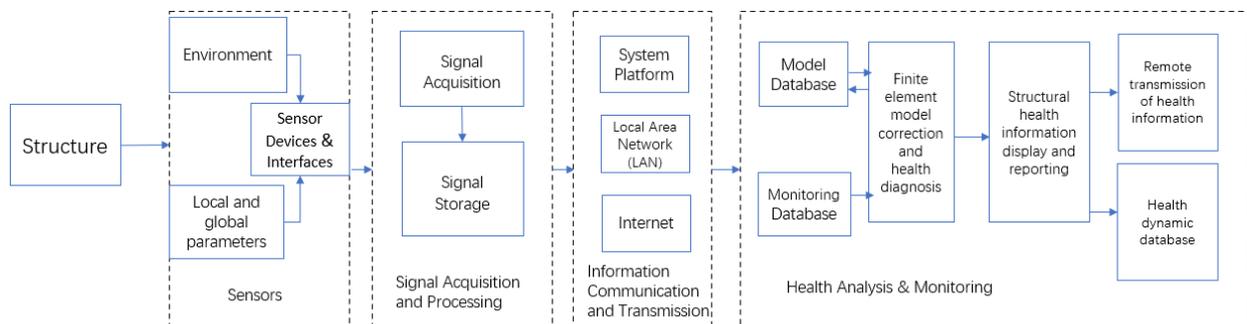


Fig 1.1 Principle of structural health monitoring system

Fig. 1.1 shows that a traditional structural health monitoring system which capable of assessing the structural state and ensuring the safe operation of the structure. The system achieved by few steps. First, install various sensors on the structure to be monitored to collect the characteristic information of the structure, and then use the relevant algorithm to process the collected data, analyze and predict the various responses of the structure, and display it in real-time status. In response to the normal operation of the structure, alarm and limit measures will be taken.

Therefore, SHM systems include three key elements [3]:

- A network of sensors, permanently attached to the structure. This aspect establishes the main difference from conventional Non-Destructive Testing (NDT) procedures and is essential for performing automated inspections.
- On-board data handling and computing facilities. The huge number of sensors continuously produces a large amount of data to be processed in real-time. SHM was feasible when large-capacity PCs were available (since the mid-1980s).
- Algorithms that compare stored data from the pristine structure with recently acquired data, after correcting for environmental factors, to calculate a damage index and to inform about damage existence, localization, and type.

In conclusion, Real-time monitoring of structural performance can ensure the safety of the structure and the early detection and repair of problems, thereby greatly reducing the cost of life. Therefore, the use of health monitoring systems in critical structures (aircraft, rotorcraft, vehicles, bridges, and ships) can help detect damage and inform stakeholders about the location, nature, and severity of the damage.

1.1.2 Structural Condition Assessment and Damage Identification

In this research, we employ simulations to approximate the application of SHM techniques to a bridge-like structure health monitoring system. Health monitoring refers to the strategy and process of damage identification and its characterization for engineering structures. Structural damage refers to changes in the material parameters of a structure and its geometric characteristics. The process of structural health monitoring involves the acquisition of structural response using a sensor array with periodic sampling, the extraction of damage sensitive indicators, and the statistical analysis of damage sensitive indicators to determine the current structural health condition.[7] Structural health monitoring techniques are widely used in the maintenance and management of large bridges, and these monitoring systems have accumulated a large amount of data[8], based on which effective interpretation of the state of the structure and identification of possible damage has become a key issue in current SHM research.

In traditional bridge maintenance management, structural condition assessment based on manual inspection plays an important role, however, manual inspection is heavy and subjective, and it is difficult to achieve long-term quantitative tracking of structural performance. However, the massive monitoring data accumulated by the bridge structural health monitoring system has not been effectively utilized due to limitations of data analysis methods and algorithms[8].

There are two types of methods: ‘model-based’ and ‘data-driven’ for structural condition assessment and damage identification using SHM data. The model-based approach is essentially a process of finite element modeling of bridge structures[9], model modification, and system parameter inversion, which requires high accuracy of the theoretical model and quality of monitoring data and is not ideal for application in practical engineering. The data-driven approach focuses on the change pattern of the correlation between the input and output data obtained from monitoring to identify the mode corresponding to the structural state, and with the help of the mature statistical theory, the data-driven approach is widely used in SHM[10].

1.2 Big Data and Artificial Intelligence in SHM

Traditional statistical methods can only analyze a small number of low-dimensional data samples due to the limitation of computing power and analysis methods[8], and cannot present the analysis results efficiently, so they are not yet sufficient to solve the analysis problem of massive and high-dimensional SHM data. The emergence of big data and artificial intelligence tools have promoted the development of SHM.

The Concept of Big data

The concept of ‘big data’ was first introduced in 1996, the column of big data was launched in Nature magazine in 2008, and the research report of McKinsey & Company in 2011 summarized a comprehensive analysis of key technologies and application areas of big data. A National Institute of Standards and Technology report defined big data as consisting of ‘extensive datasets—primarily in the characteristics of volume, velocity, and/or variability—that require a scalable architecture for efficient storage, manipulation, and analysis.’ The connotation and extension of ‘big data’ are constantly enriched, in different literature,

Data	Logic	Technologies			
Large volume	Use all data for analysis	acquisition	storage	Processing	Analysis
Large variety	Accept data confusion, the large amount of data to compensate for poor quality Pursue correlation rather than causation	Internet	HDFS	Hadoop	Knowledge Discovery
Fast growth rate		Internet of Things	NoSQL	Storm	Machine Learning
Low value density		Internet of Vehicles	NewSQL	Spark	Data Mining
				Cloud Computing	Pattern Recognition
				GPU	Statistics

Fig 1.2 Components of Big Data

Big data is described as data sets, achievable functions, data realization process, architecture, and technology. Big data should be understood in terms of data set characteristics, Logic, and technologies. Big data analysis should have three basic logics, namely, ‘use all data for analysis’, ‘accept data confusion, the large amount of data to compensate for poor quality’, and ‘pursue correlation rather than causation’. Thus, it has the following characteristics: compared with traditional statistical analysis using only a small random sample of data, using the whole data can find more details and valuable information; accepting data confusion and increasing the amount of data can simplify the analysis model and avoid overfitting, to obtain

more accurate analysis results; cutting from correlation can provide a new perspective for data analysis. The current big data processing technology has made it possible to analyze all data, and many successful applications have been achieved by mining correlations through big data analysis; however, when the quantity of data is large to compensate for the poor quality, if the noise in all data is more than the signal, the signal is easily masked, therefore, we cannot blindly include all data, but still need to find data that are strongly correlated with the analysis purpose.[11]

Fig 1.2 shows the components of Big Data. The technology of big data is mainly reflected in several aspects such as data acquisition, storage, computation processing, and analysis methods. The collection, storage, and computation of big data are mostly presented in the form of software tools, such as Internet of Things and Internet for data acquisition, Hadoop distributed file system and NoSQL database for data storage, and Hadoop ecosystem, MapReduce, Spark, Storm, and cloud computing for data computation. Big data analysis methods cover data analysis methods expressed in various terms, including knowledge discovery in databases (KDD), data mining, machine learning, pattern recognition, statistics, etc.[12]

The concept of Artificial Intelligence

An important concept related to big data analytics is artificial intelligence, which is a grand concept that predates big data by a long shot[13]. The main difference between artificial intelligence and big data analytics is the difference in objectives, with the former generating something with intelligent behavior and the latter used to discover the knowledge hidden in the data. However, both rely on the support of big data entities to achieve and can share the same analysis methods. The current representative technology of artificial intelligence is deep learning, which belongs to a subset of machine learning and is often highlighted separately from machine learning due to its excellent performance in recognition problems such as pictures and speech[14]. The function of data visualization in the big data analysis process is to present the results of data analysis, which is also called big data visualization analysis when used directly to explore the data and mine the patterns in the data. A characteristic of big data is the variety of data, which is expressed in the data set as high dimensionality of data. High-dimensional data is difficult to visualize effectively and can cause dimensional catastrophe in data analysis, i.e., the data set is sparsely distributed in the high-dimensional space and lacks sufficient data to build models. Traditional data analysis often reduces the number of variables in the dataset by dimensionality reduction, which also brings about a reduction in the amount of information in the original dataset[15].

Big data and AI technology in SHM

Big data technology is improving the problems of insufficient computing power and inefficient data analysis methods and showing broad application prospects in data processing of SHM. Big data visual analytics provides new ideas for effective presentation and analysis of high-dimensional data and shows potential for application in pattern recognition problems in SHM. Based on SHM data, structural damage or abnormality identification can first be carried out, but this is still difficult to be ideally achieved in actual bridges structures, firstly, because most of the bridges currently installed with SHM systems have been in service for a relatively short period of time and have not yet shown obvious damage and degradation;

secondly, due to the limitations of sensing technology and data analysis methods, it is difficult to identify minor damage in the initial stage of the structure. One of the main purposes of SHM is to perform structural condition assessment. For new structures that have not yet been significantly damaged in the early stages of service, SHM data can be used to analyze the structural response patterns under normal environment and operational loads, to define the normal state of the structure and analyze the reasons for deviations from the structural state. Studies related to structural condition assessment also include load effect analysis, reliability analysis, safety warning, and sensor failure identification.[16]

Under the data-driven approach, bridge structure damage or abnormality identification and condition assessment can mostly be categorized as a pattern recognition problem. In traditional data-driven analysis, the pattern recognition is generally preceded by dimensionality reduction of multi-channel and multi-species sensor time series data to improve the efficiency and accuracy of machine learning methods. However, dimensionality reduction has the problem of fixed order, i.e., the final number of variables needs to be determined to retain as much information as possible in the original data.

1.3 Application of Sensors in SHM

The monitoring content of bridge SHM mainly includes two categories of environmental operational loads and structural response. The monitoring items of environmental and operational loads include temperature, precipitation, air humidity, wind load, traffic load, seismic input, etc.; the monitoring items of structural response include geometric deformation and displacement, acceleration, strain, angle of rotation, cable force, etc.[17]

1.3.1 Sensor Types

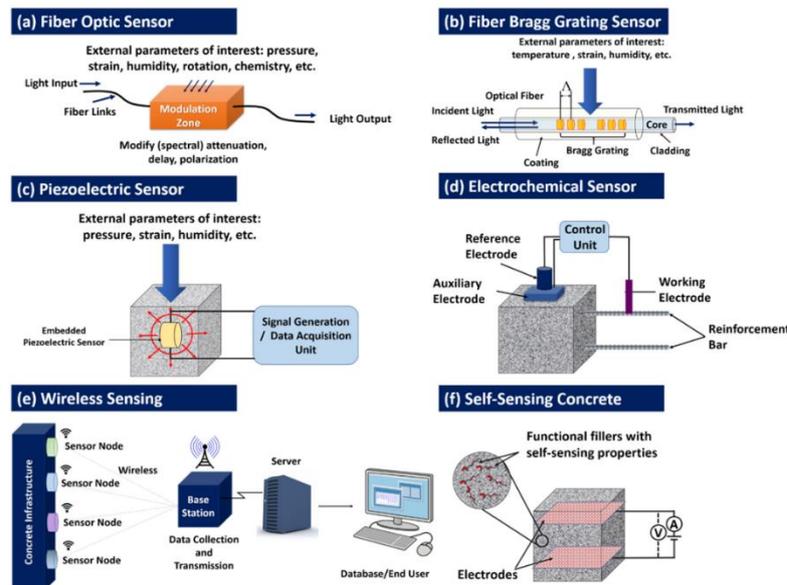


Fig.1.3 from [17] illustration of a sample(a)Optical Fiber Sensor(b) Fiber Bragg Grating Sensor(c)Piezoelectric sensor(d)Electrochemical sensor(e)Wireless sensor system (f)Self-Sensing Concrete

SHM technique purely depends on the sensors to monitor the structure, so different sensors can be used. There are many types of sensors used by the industry for structural health monitoring [18]. Three mostly used types are discussed in the following:

- (1) Fiber optic sensors (FOS)
- (2) Piezoelectric sensors (PZT)
- (3) Micro-Electro-Mechanical systems (MEMS) Wireless Sensors

Fiber optic and Bragg grating sensors

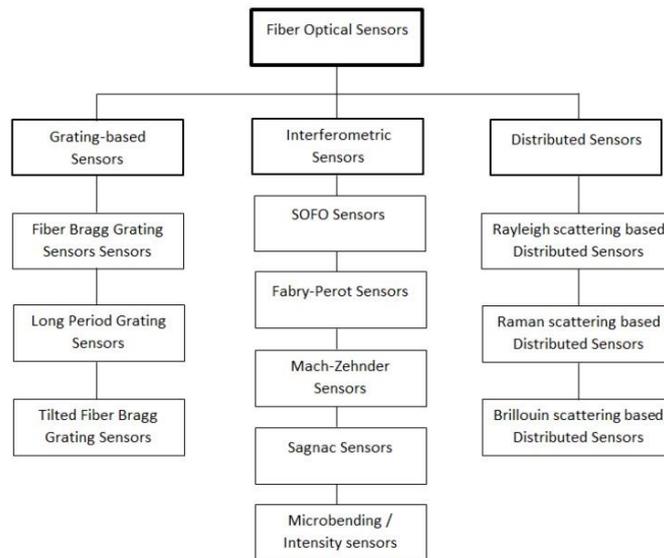


Fig1.4. Overview of fiber optic sensor technologies

One of the fastest growing areas in SHM of civil structures is the development and the use of fiber optic sensors (FOS). The fiber optic sensing function is based on changes in the characteristics of the light signal transmitted along the fiber. A typical optical fiber is made of a glass fiber. Although polymer optical fibers (POF) were introduced at the same time as glass fibers and they are much easier to handle, the FOS market is dominated by glass optical fibers (GOF). Polymer fibers are also able to perform much larger strain rates (up to 100%) while the limit of glass fiber is around 2% [19]. An optical fiber is capable of transmitting light and signal over very long distances without the need for amplification. Exposure to external perturbations or deformation (e.g., stretch, bend or strain) of optical fiber impacts the transmitted optical signal and leads to a change in the characteristic of an output signal or a shift in the reflected wavelength.

Optical fiber sensors have several advantages over other sensing methodologies, including higher sensitivity, reduced size and weight, immunity to electromagnetic interference (EMI), as well as being relatively inexpensive [20], [21]. **Fig1.4.** Shows a variety of fiber optic sensor technologies. Fiber optic sensors come in varieties of types including interferometric, intensity, microband, polarimetric, wavelengths, distributed (using Brillouin, Rayleigh or Raman scattering techniques), fiber optic Bragg

grating (FBG) and hybrid sensors [22]. The shortcomings of FOS include the expensive nature of its interrogation system, fragility, the need for specialist expertise in the construction and deployment of fibers and the need for several repeaters to boost the signal. **Table 1.1** summarizes the most used fiber optic sensors and their advantages, disadvantages, and main applications.

Table 1.1. Characteristic summary of different fiber optic sensor type

Type of sensor	Advantages	Disadvantages	Applications
Interferometric Fabry-Perot sensor	High strain resolution	Hardly multiplexed, cross temperature-strain sensitivity, fragile	Strain, temperature, vibration, cure monitoring
Grating-based sensors (e.g. FBG)	Discrete measurements of strain and temperature over large areas and at selected needed locations, well know technology	Cross temperature-strain sensitivity, limited damage severity and location assessment	Temperature and strain measurements, low velocity impact damage detection, damage localization, cure monitoring
Distributed Optical Fiber	Measurements at any location along the fibre length, suitable for monitoring of large area composites	Expensive interrogation systems, spatial resolution in the cm to m range	Strain, temperature, vibration, delamination

Piezoelectric sensors (PZT)

Discovered by the Curie Brothers in the early 1880 s, piezoelectricity became the centerpiece of numerous applications. A piezoelectric sensor can measure variations in parameters such as acoustic emission, temperature, strain, force, pressure, or acceleration. It converts these parameters to a measurable electrical charge. Piezoelectric materials can transform electrical energy into mechanical energy, and conversely mechanical energy into electrical energy. Piezoelectric materials can play different roles; they can function as sensors, actuators, accelerators, and transducers. Among piezoelectric materials (ceramics, polymers, and composites), piezoelectric ceramic materials are the most dominant material type. They have very stable mechanical properties under a wide range of temperatures. According to the EN 50324 European Standard [23], PZT properties can be classified as either soft or hard. Soft PZT materials are ideal candidates for actuators and sensors, while hard PZT materials are suitable for high-powered applications and ultrasonic transducers. The use of piezoelectric patches and ‘smart patches ‘have been reported for structural health monitoring, rehabilitation and vibration damping [24].

There are still some faults in the design of piezoelectric transducers and sensors which can lead to false indicators on structural conditions. The design challenges include, but are not limited to the water solubility of some piezoelectric crystals (not suitable for humid environments), the influence of the temperature on the piezoelectric properties, limitations in the enhancement of the piezoelectric properties (e.g., the electromechanical coupling coefficient, the acoustic impedance), gradual degradation of the electromechanical properties of a PZT transducer, and the influence of the bonding layer between a PZT patch and a concrete structure.[17]

Micro-Electro-Mechanical systems (MEMS) WSS

Wireless sensors are nodes and platforms for autonomous data acquisition, so they are not true sensors. Traditional structural sensors such as piezoelectric pads can be attached to wireless platforms and benefit from their mobile computing and wireless communication elements. Wireless technology eliminates the need for extensive wiring between sensors and the data acquisition system, hence they have lower installation costs and allow for flexible system configurations. However, battery-powered wireless sensors are still intertwined with significant challenges related to their energy consumption, size, cost, communication range, hardware design and risk of data loss. Power-free wireless alternatives, known as radio-frequency identification (RFID) sensors, are being researched currently to address power consumption issues in wireless sensors [25]. **Table 1.2** shows the Characteristics comparison of different types of sensors for structural health monitoring to facilitate the selection of sensors in further research in this thesis.

Table 1.2. Characteristics comparison of different types of sensors for structural health monitoring

Type of sensor	Advantages	Disadvantages	Applications
Fiber Optic	Immune to electromagnetic interference, small size, light weight, cheap, durable, allow sensor multiplexing, sensing of large area structures, applicable to all type of composite materials, commercial solutions available solutions that can be tailored for each application	Expensive optoelectronic interrogation systems, temperature sensitive, fragility of the sensors makes embedding procedure challenging, strain is partially absorbed by the protective layer	Strain, low velocity impact damage, cure monitoring
Piezoelectric	High mechanical strength, operational for a wide frequency range, cheap, small sizes, work in both active and passive sensing methods, many commercial solutions available including higher sensitivity, reduced size and weight, immunity to electromagnetic	Lengthy cables, difficult signal interpretation	EMI: damage extent with statistic damage metrics; Lamb waves: detection of cracks, corrosion, delamination, holes, notches, degradation of lap joints, sensitive to small damages, can inspect large areas, can locate damage; Acoustic emission: passive sensing method, detection of matrix cracking, fibre-matrix debonding, delamination, fibre breakage, location of damage possible
Micro Electro-Mechanical System (MEMS) wireless smart sensors (WSS)	State-of-the-art WSSs offer the promise of wireless communication, onboard computation low cost, less invasive installation, small size, and performance equivalent to that of their macro-scale counterparts	Low resolution compared to wired accelerometers used in SHM application	Wireless sensors are nodes and platforms for autonomous data acquisition, Traditional structural sensors such as piezoelectric pads can be attached to wireless platforms

1.3.2 Common Sensor Errors

Sampling rate error

According to sampling theory, which is known as Nyquist sampling theorem[26]:

$$f_s \geq 2f_{max} \quad (1-1)$$

Where f_s is the sampling rate and should be choose, f_{max} is the maximum frequency of structure, when the sampling frequency adopted satisfies the formula, it is called over sampling. In this case, the sampled data has complete information and can restore the real situation. However, when the sampling frequency is not met the equation, it is called under-sampling. Under-sampling is the distortion of the data collected, and a huge sampling error is received, which cannot restore the real situation of the sample. At present, to ensure the accuracy of the experiment results, experts generally use oversampling, that is, to satisfy the Nyquist sampling theorem.

Time error

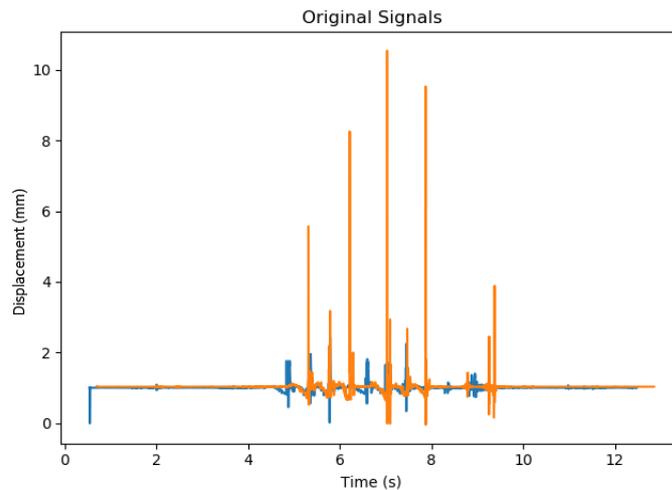


Fig 1.5. Time error caused by two different signals

In structural health monitoring, researchers use quite a lot of sensors, either of the same type or of different types, to measure different parts and characteristics of the same system. If the same signal acquisition system is not used, or if the sampling rate and zero time differ [27] between sensors, synchronizing these signals in post-processing can create significant difficulties.

As shown in **Figure 1.5**, the blue and orange signals come from two independent sensors that record displacements and have same sampling frequency. However, due to zero-point drift, in some cases two independent signals can be misinterpreted as separate signals with different sampling rates. In practice research will have data records of similar content from multiple independent sensors and would like to align the records to perform more complex analysis between them. However, even for devices configured to have the same sample rate, small differences from the true sample rate will appear on each device and can cause gradual deviations between samples. Devices may also have different internal timestamps at the start of recordings due to zero drift, resulting in an initial time offset between recordings. Therefore, researcher should perform synchronization on these records to account for both sample rate drift and initial clock offset. Synchronization needs to be both accurate and have a reasonable runtime. For high-precision accurate and expensive sensors and sensor acquisition systems, we can reduce the impact of time errors, and some devices can also be manually time-calibrated.

Sensor faults

A sensor is faulty when its measurements display unacceptable deviations from the true values of measured variable. Various sensor fault modes can occur due to different factors, such as a malfunction or failure of sensor components and the effects of electromagnetic interference. According to Kulla [28], there are seven typical sensor fault types: bias, drift, gain, precision degradation, complete failure 1 (constant), complete failure 2 (constant with noise) and complete failure 3 (bottom noise). Bias, drift, gain, and precision degradation are usually called soft sensor faults, in which the sensor has partially failed, whereas the three types of complete failure are usually called hard sensor faults.

Let $x^*(t)$ represent the true value of measured variable and $w(t)$ the normal sensor measurement noise. Then, the normal value displayed by a sensor, termed $x(t)$, can be expressed by:

$$x(t) = x^*(t) + w(t) \quad (1-2)$$

where t denotes the sampling instant. The mathematical expressions of the seven typical sensor fault types are listed in **Table 1.3**, where a , b and G are three parameters which control the corresponding sensor fault magnitudes, as well as $e(t)$ is an excessive random noise with zero-mean caused by sensor fault.

Table 1.3. Mathematical expressions of seven typical sensor fault types.[28]

Fault type	Mathematical expression
Bias	$x(t) = x^*(t) + w(t) + a$
Drift	$x(t) = x^*(t) + w(t) + a + bt$
Gain	$x(t) = G(x^*(t) + w(t))$
Precision degradation	$x(t) = x^*(t) + w(t) + e(t)$
Constant	$x(t) = a$
Constant with noise	$x(t) = a + e(t)$
Bottom noise	$x(t) = e(t)$

Fig. 1.6 is used to intuitively interpret the seven typical sensor fault types. The blue solid line represents the true values of measured variable, the grey solid line represents the normal sensor measurements, and the red dashed line represents the sensor outputs corrupted by typical fault types. A sensor is claimed to have bias if its outputs differ from the normal values by a constant. Sensor drift refers to the case where the differences between sensor outputs and normal values change linearly with time. Gain occurs when the normal values of a sensor are multiplied by a constant. The case in which sensor outputs are added to an excessive random noise is referred to as precision degradation. A fault is classified as complete failure 1 when sensor outputs remain constant with time and as complete failure 2 or 3 if sensor outputs are constant with noise or only noise regardless of the change of normal values.

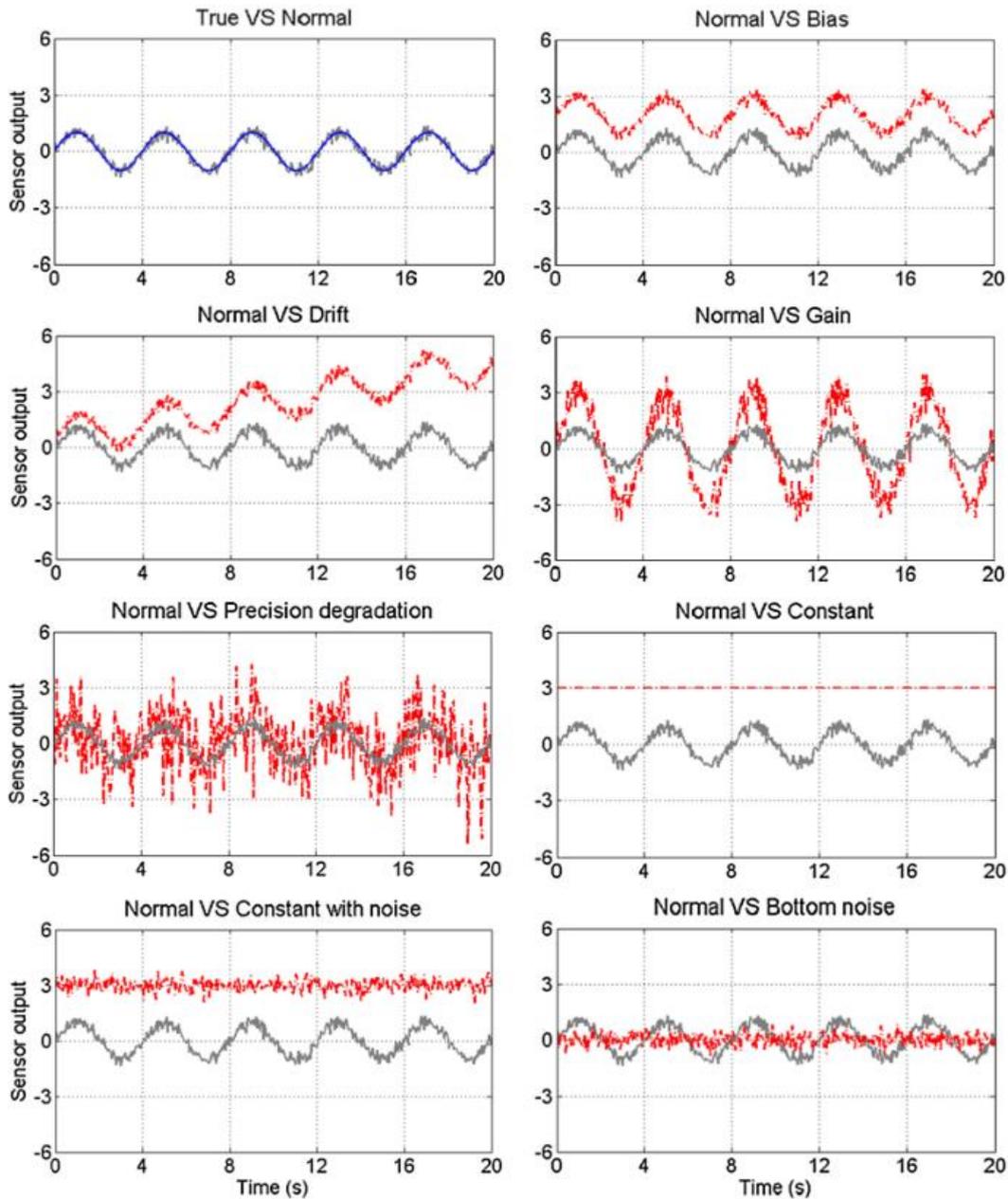


Fig. 1.6. Graphical representations of the outputs of normal sensor or faulty sensor corrupted by typical fault types[28]

1.3.3 Post Processing of signals

The sensor signals refer to the signal collected by the sensor after the signal of each channel is converted by the amplifier or converter, and then collected and concentrated by the data acquisition instrument. In the process of data conversion and transmission, due to the influence of external factors and sensor errors, the data collected by the collector must be mixed with many unwanted or problematic signal components. Therefore, before using the collected data, it is necessary to perform preliminary signal preprocessing on

the collected signal data to correct the distortion of the waveform, remove some noise and interference mixed in the signal, and weaken the redundant content in the signal.

Smooth Denoising

The signal obtained by the data collector will contain noise components. These noise signals mainly include irregular random interference signals and other periodic high-frequency interference signals. Compared with the normal signal, the random interference signal has a wider frequency band, so there will be many burrs on the collected discrete vibration signal data curve, and these burrs are the manifestation of the interference signal. The data is smoothed to reduce the influence of the interference signal on the real data. Smoothing can also be used to remove signal irregular trend terms. In the process of measuring the signal by the sensor, it will be interfered by a variety of influencing factors, which will cause the sampling signal of individual measuring points to have a very large deviation from the baseline[29].The smoothing technology removes irregular trend items from the collected Subtract from the original signal to obtain the true value of the data signal.

Filtering

In signal analysis, data filtering is the process of extracting part of the signal we are concerned about and interested in from the acquired signal. Its main functions are as follows: filter out noise or false components in test signals, improve signal-to-noise ratio, smooth analysis data, suppress interference signals, share frequency components, etc.

SNR enhancement with POST-processing Algorithms

Sensors suffers from low Signal-to-Noise-Ratio (SNR) in practical applications. Hence, post processing of signals is required to improve system SNR. In [29] six SNR enhancement post-processing algorithms including temporal matched filtering (ATMF), wavelet denoising (WD), empirical mode decomposition (EMD),two-dimensional edge detection (2D-ED), two-dimensional adaptive bilateral filtering (2D-ABLF) and nonlocal mean algorithms (NLM) are discussed in detail. These algorithms can be divided into two categories: one- and two-dimensional denoising. According to the properties of these algorithms, ATMF, EMD, 2D-ED only improve the localization SNR but the other algorithms (WD, 2D-ABLF and NLM) enhance the signal SNR and localization SNR simultaneously.

Artificial Intelligent method in post-processing of signals

In current research on sensors and data analysis, two most common approaches are principal component analysis (PCA) and artificial neural networks (ANN). They are both used to model the normal sensor behavior and the newly observed readings will be compared to the model to determine if it is anomalous[30].Other techniques for fault detection include Ensemble Classifiers, Support Vector Machines, Clustering, and hybrid methods. For error correction, researcher proposed methods for missing data imputation and noise correction. The most common missing data imputation technique is Association Rule

Mining, with half of the respective papers proposing variations of that approach. Other approaches comprise of k-Nearest Neighbor, clustering, tensor-based singular value decomposition, and Probabilistic Matrix Factorization (PMF). On the other hand, error detection and correction problems, usually termed as Fault Detection, Isolation, and Recovery (FDIR). The PCA-based approach is the most common technique for FDIR, though there are other approaches such as ANN, Bayesian Network, and hybrid methods involving Kalman filter and Dempster–Shafer theory with Ontology.

1.3.4 Parameters for Specific Sensors

In **Table 1.4**, several representative vibration signal sensors and parameters for large structures and bridges are listed. In the next chapter, the comparison will be analyzed, and the sensor type chosen and targeted in this article will be determined.

Table 1.4 Summary of typical accelerometers used in SHM for bridges [31] [32] [33]

Type	Model	Interface	Noise-Density $\mu\text{g}/\sqrt{\text{Hz}}$	Sensitivity (V/g)	Range(g)	BW(Hz)
MEMS	LIS3L02DQ	Digital	110	1024LSB/g	± 2	0~1120
MEMS	M-A351	Analog	0.5	1	± 5	0~100
Piezoelectric	KB12VD	Analog	1 μg	1	± 0.6	0~300
Piezoelectric	KS48C	Analog	14 μg	1	± 6	0~4000
FOS	OS7510	Analog	50	5	± 10 below 10 Hz	0~350
FOS	OS7520	Analog	1	0.25	± 1 below 10 Hz	0~100
FBG	FS65ACC	Analog	0.5	59pm/g	± 10	0~50

1. Sensitivity for digital accelerometers is given in digital units (LSB) per g-force (g).

2. It is difficult to compare digital and analog sensors sensitivity in absolute terms, since the analog sensor digital sensitivity can be improved with better ADC characteristics.

3. Noise = noise density * sqrt (bandwidth * 1.6) Sensitivity should be, generally, the highest possible, since the typical level of vibration in SHM is small and a high sensitivity will likely produce a higher signal to noise ratio (SNR). For a sensor, the signal-to-noise ratio is not fixed. The noise in the parameters is the noise of the standard input at a given power, which is different from the actual application.

2. Main Tasks and Research Parameters

Structure health monitoring technology is to install sensors at key parts of the structure to measure the state and response. The sensors will transmit the monitored and collected data to the computer in real time, form a database. Combined with the information computer will automatically analyze the safety of the structure is displayed in the form of a visual interface. For the state that endangers the safety of the structure, an early warning will be taken to provide guidance and other related measures to avoid or reduce the possibility of accidents and achieve the purpose of ensuring the service safety of the bridge structure[34].

Yan et al. [35] have demonstrated the effectiveness of Subspace-based damage indicators in structural health detection. However, as we mentioned in Chapter 1, in the real world, sensors inevitably generate noise errors. Sources of error include the quality of the sensors itself, reading errors, and disturbances caused by the environment. Therefore, the main task of this thesis is to explore the effect of error on the experimental results, that is, the behavior of the deep learning model and different damage indicator under different noise effects, so to investigate the practical effect of deep learning model and different damage indicators in real working conditions. This demonstrates that the experimental results can be generalized to real-world sensors to verify whether subspace-based indicators are powerful indicators in the structural health monitoring of bridges (Beam) and the indicators' robustness for real sensors and real-world noises. As shown in **Figure 2.1**, we use a numerical beam model to get the raw and ideal data, which include reference initial state (undamaged) and current state (damaged or not). And we set 3 different reasonable Signal-to-Noise ratio levels to simulate the signals are collected by real sensors in different working conditions.

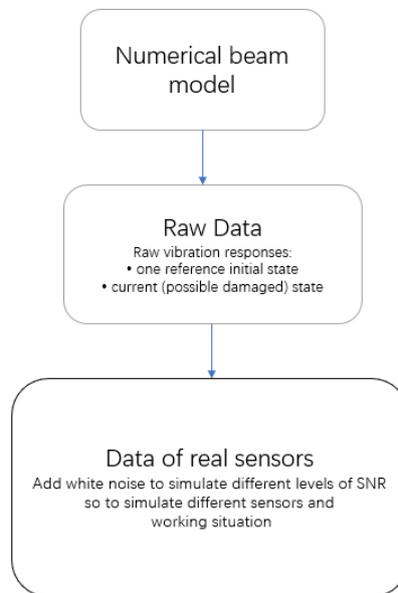


Figure 2.1. Datasets collection

After we collected these signals, we will extract three sets of damage indicators for each situation. As shown in **Figure 2.2**. In situation (A) we only use some statistical features estimated on the raw vibration data as damage indicators, in situation (B) we add an additional feature – a subspace-based indicator. And in situation (C) the third set of damage indicators considers as input a class of subspace-based damage indicators in the attempt of removing the arbitrary user’s choice of the parameters which significantly affected their calculations.

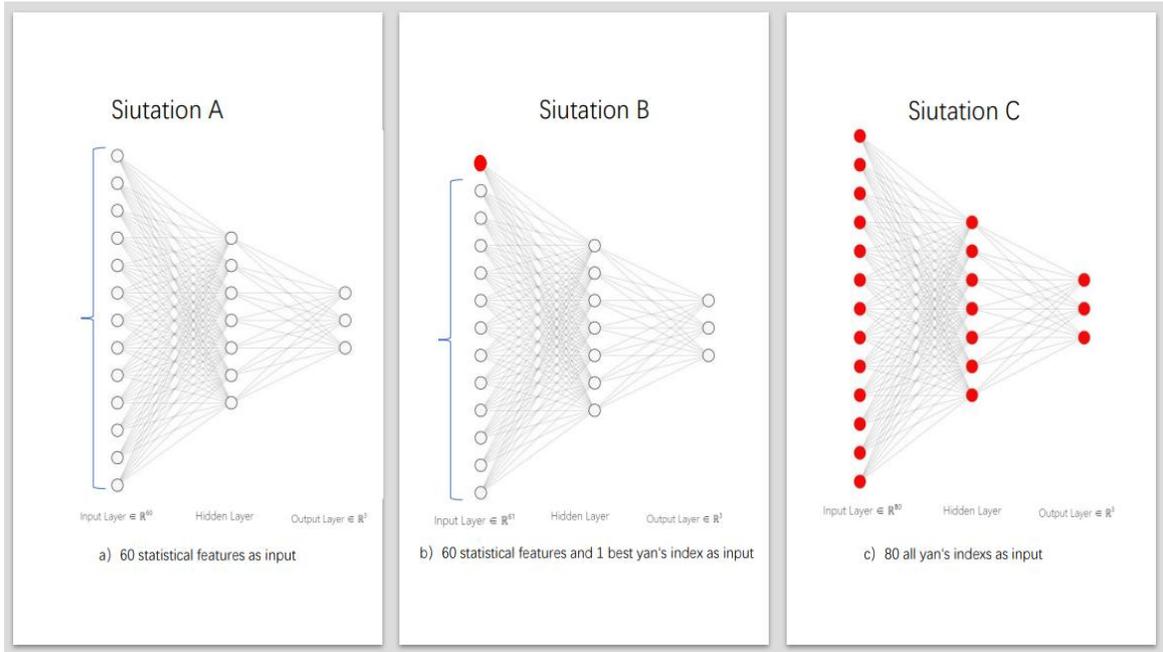


Figure 2.2. Model Schematic (not to scale) to solve SHM Level 1 by classification task performed by MLP or CNN architecture.

The current chapter is organized as follows. Section 2.1 presents the benchmark numerical beam model case studied and analyzed in this thesis, which is being used and proposed by ROSSO et al.[36].In section 2.2, author compared different sensor types, describe the type of sensors this research is aimed at, and the method selected to reduce unnecessary errors caused by the sampling process as introduced in section 1.3.2. Whereas Section 2.3 investigated the most used method to simulate real-world sensors and signal types from ideal sensors. Finally in Section 2.4 the theoretical basics of subspace-based damage indicators are briefly presented.

2.1 Numerical Beam Model

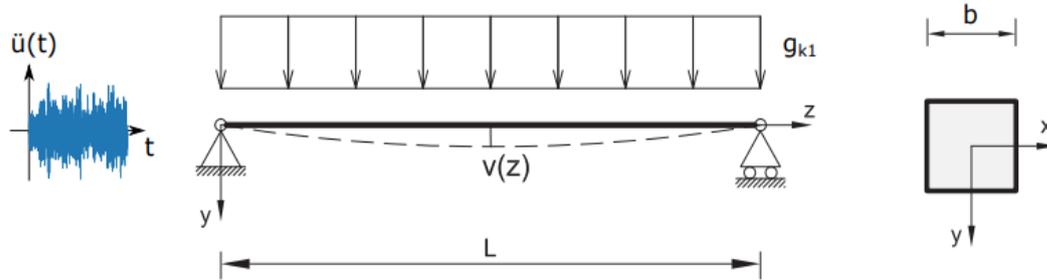


Figure 2.3.from [36] Simply supported beam geometry illustration. x, y, z denotes the Cartesian coordinates reference system, b is the square cross section side, L is the span length, g_{k1} is the self-weight load which produce the static deflection $v(z)$. A Gaussian white noise acceleration input ($\ddot{u}(t)$) is adopted as dynamic input to the simply supported beam.

In current study, we accepted an initial numerical beam model proposed by Rosso et al in [36]. As shown in **Figure 2.3**, this structure is implemented by the OpenSeesPy module[37]. OpenSeesPy module allows to exploit all the advantages and powerful capabilities of the open-source OpenSees finite element software controlled by a Python interpreter instead of the traditional Tool command language (Tcl) approach. The numerical model in this study analyzed is a steel beam which is simply supported. The beam has a square cross-section of side 0,10 m and a span length of 2,00 m. The steel material is characterized by Young's modulus $E = 210$ GPa and a mass density of $\rho = 7850$ kg/m³ like in the normal case. One beam support was dynamically excited by a Gaussian random white noise acceleration. The white noise process has been generated by a random sampling of a standard zero-mean normal $N(0, 1)$, normalized, and rescaled up to 0.01 g of peak ground acceleration (PGA). The input acceleration has been imposed similarly to an earthquake excitation limited to the vertical direction only and in one single support to be sure to excite all modes. Theoretically, it would be more accurate to excite every mass independently in the beam nodes. However, any perturbation propagates almost instantaneously in every element of the beam due to the high elastic wave propagation velocity compared to the modal dynamics[38].

2.2 Selection of Sensor Parameters

2.2.1 Selection of the Sensor Type

Measurement Parameters - Accelerometer

The data-driven approach focuses on the change pattern of the correlation between the input and output data obtained from monitoring to identify the mode corresponding to the structural state, and with the help of the mature statistical theory, the data-driven approach is widely used in SHM. To evaluate the state of health of a specific structure, methods widely used today in structural health monitoring systems rely on the use of output-only vibration data. As in section 2.1, the model used is excited by acceleration so we will use accelerometers.

Sensor Type - Micro Electro-Mechanical System Sensors

In recent years a shift of SHM research away from traditional wired methods toward the use of wireless smart sensors networks (WSSN) has been motivated by many attractive features of wireless smart sensors (WSS). The progress achieved in Micro Electro-Mechanical System (MEMS) technologies and wireless data transmission, has extended the effectiveness and range of applicability of WSSNs. State-of-the-art WSSs offer the promise of wireless communication, onboard computation, low cost, less invasive installation, small size, and performance equivalent to that of their macro-scale counterparts [39]. These features enable the deployment of a dense array of sensors on structures, which can provide real-time information about the performance of civil infrastructure. One of the most important structural responses employed by SHM systems is acceleration, and as a result, nearly all wireless smart sensors include accelerometers. WSSs traditionally rely on battery power, low-power MEMS-based accelerometers are typically employed. So, in this study, we will focus on MEMS-WSS accelerometers sensors. Some commonly used MEMS-WSS accelerometers in SHM are summarized in **Table 2.1**.

Table 2.1 Summary of typical MEMS- WSS accelerometers used in SHM for bridges[40]

Model	Interface	Noise-Density $\mu g/\sqrt{Hz}$	Voltage Source(V)	Sensitivity (V/g)	Range(g)	BW(Hz)
SD-1221	Analog	5	5	2.0	± 2	0~400
CXL01LF	Analog	70	6~30	2.0	± 1	0~50
CXL02LF	Analog	140	6~30	1.0	± 2	0~50
AC310-002	Analog	10	9~20	2.0	± 2	0~300
LIS2L02AL	Analog	30	3.3	0.66	± 2	0~100
LIS2L06AL	Analog	30	3.3	0.66/0.22	$\pm 6/\pm 2$	0~100
LIS3L02AS4	Analog	50	3.3	0.66	± 2	0~50
LIS344ALH	Analog	50	3.3	0.66/0.22	$\pm 6/\pm 2$	0~1800
LIS3L02DQ	Digital	110	3.3	1024LSB/g	± 2	0~1120
SF1500	Analog	0.3	$\pm 6 \sim \pm 15$	1.2	± 3	0~1500
SF1600	Analog	0.3	$\pm 6 \sim \pm 15$	1.2	± 3	0~1500

2.2.2 Effects of Sensor Errors

Sampling errors

According to the *Shannon sampling theory* (1-1), if we want to ensure the integrity of structure characteristic extraction, the sampling rate is more than two times of the highest frequency of structure itself. However, the natural frequency of structure can only be estimated based on experience, but it is not easy to estimate when the structure has recessive damage. For SHM, if we choose a higher sampling rate. SHM resources are limited, if monitoring method need a high rate to collect data, which would occupy lots of space and computing resources, of storage and transmission, in addition, there are problems such as storage and transmission, so it is not conducive to long-term monitoring. [41]The low-rate data collecting makes the monitoring sectors wider, more monitoring points and higher redundancy. Therefore, continuity and long-term property of monitoring, safety and integrity of data are well guaranteed. In theory, the higher the sampling frequency, the better, and the smaller the amplitude error caused by the sampling rate, but this

is not realistic. Because the sampling rate is limited by the highest sampling frequency of the acquisition devices, on the other hand, the higher the sampling rate, the larger the sampled data capacity and the larger the data file. In this thesis, author decide to choose a rather higher sampling rate for accuracy our model and algorithms. So, we will improve our sampling rate to avoid the sampling rate error. For further study, we can see the influence of lower sampling rate on our designed model and algorithm.

Time errors

The time error problem of the sensor has been introduced in Section 1.3.2. In actual process, the methods used to compensate for time errors mainly come from the field of autonomous driving[42].For the issue shown in **Figure 1.5**. First, we need resampling to ensure that the two signals have the same sampling frequency. Then, we must determine the time offset between the two recording starts. This problem can be defined as finding the optimal offset for the signal. When synchronizing signals with many samples, the search space for optimal offset is large, and computing how the two signals are synchronized is computationally expensive. Intuitively, the way we avoid this is to synchronize the ‘primary’ events in the recording, since these ‘primary’ events should be recorded in both signals, so their offset should align the signals. With this in mind, we generate a subset of the search space that likely contains the best solution. To determine this subset, we take the Point of Interest (POI) in each signal and use the offset that occurs when choosing a pair of points (one from each signal) to align the signals. Here, we define POI as any peak or valley found in the signal (**Figure 2.4**).

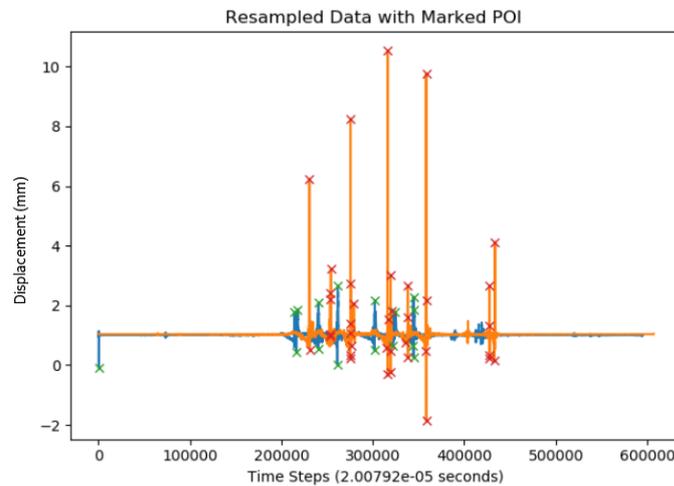


Figure 2.4. Resampled Data with Marked POI signed as 'X'. The points are designed to capture the main event that has been recorded in both signals

Figure 2.5. shows signals are aligned according to the best POI pair. It is the same signal shown at the beginning of this article but has been synchronized to deal with problematic time shifts and sampling frequency errors, and the signal's timestamps have been adjusted to better represent when data was recorded.

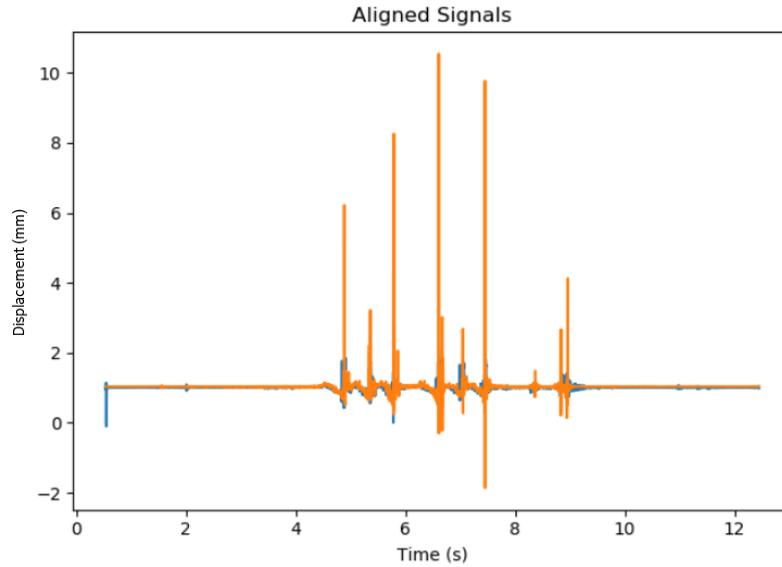


Figure 2.5. Aligned signals

Errors due to noise

As we already discussed there are many different sensor errors. However, in the SHM research process, we adopted a method that relies more on output-only vibration data, so we will add Gaussian white noise[43] as the sum effect of sensor errors, which is a well-acknowledged approach in the scientific literature to simulate the error and noise of our sensors, the unknown environmental dynamic excitation, and operational conditions in numerical cases or laboratory experiments. More details were specified in Section 2.3.

2.3 Gaussian White Noise

2.3.1 Theoretical Support for White Gaussian Noise

In the research of signal or image noise reduction, many researchers [44] use Gaussian white noise to add to clean samples to simulate noise-containing samples, and use this to verify the noise reduction effect of the proposed model (Such as noise reduction auto-encoder or Denoising Auto-encoder).

Synthetic noise rather real noise

Compared with real noise, the advantage of using synthetic noise is that it is easy to analyze the problem or the design algorithm, and it is easy to quantify and evaluate the effect of the algorithm. The essence of noise reduction is the reconstruction of the data itself to exclude corruption. Models and analyses of data and noise are required. The data models are those generally used, such as sparse coding, probabilistic, low-

rankness, collaborative filtering. These are based on certain mathematical assumptions.[45] As for the noise model, generally researcher define the pollution of noise as an additive or multiplicative random variable. Then if one knows the random distribution of this random variable, it's possible to design a corresponding reasonable algorithm. For real noise it's impossible to find its' distribution because real means unknown. Noise can be unstructured or structured. The noise in real data can be consistent or variable. The problem analysis in this case is extremely difficult, or the problem itself is untraceable and not well defined. Also, synthetic noise is convenient to quantify and evaluate the effect of the algorithm: To evaluate the effect of a noise reduction algorithm, a certain evaluation standard (metric) needs to be adopted. We generally divide evaluation criteria into objective and subjective. [46]

- Objective criteria: Common such metrics are Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE), Structured Similarity (SSIM), etc. The level of the absolute value of the metric directly reflects the quality of the method. There is also some work that tries to propose some objective quality metrics that do not require ground truth, but the most used classic metrics of this type require the noise-free ground truth of the signal as a reference without exception.
- Subjective criteria: When using simulated noise, it's naturally to have ground truth. But if it's real noise, it's impossible.

So, in general, for the noise reduction experiment of real noise, researcher must accept some subjective metrics: let the peoples to identify whether the effect of the noise reduction is good. Different people may have different preferences which often leads to individual differences in evaluation and disputes. Even the cost will be very high, which is not conducive to the efficiency of scientific research.

Gaussian noise

Compared to other synthetic noise distributions, Gaussian noise may be the best simulation of real noise when the noise source of real noise is particularly complex. In fact, it is not only the noise reduction algorithm of machine learning, but also traditional methods (Sparse coding, GMM, low-rank, collaborative filtering) and most of them like to use Gaussian White noise for simulation experiments.[46]

Gaussian noise is used to better simulate unknown real noise: in real environments, noise is often not caused by a single source, but a complex of noises from many different sources. The real noise can be regard as the sum of many random variables with different probability distributions, and each random variable is independent, then according to the ***Central Limit Theorem***, their normalized sum tends to increase with the number of noise sources close to a Gaussian distribution. In ***Shannon's information theory***, entropy is used to represent the degree of uncertainty of information. According to the maximum entropy theorem of limited power: If the average power is limited, then when the source conforms to the Gaussian distribution, the entropy of the source is the largest (the uncertainty of the noise is the largest, and the interference to the source is the largest). That is, white Gaussian noise is the noise that interferes the most with information if power is assumed to be limited. So not only deep learning, but many other communication systems use Gaussian white noise as noise.

Based on the assumptions above, the use of synthetic Gaussian noise is a simple and good approximate simulation when dealing with this complexity and without knowing what the real noise distribution is.

Gaussian white noise for Algorithm generalization

· Traditional case

Algorithms that perform well under Gaussian white noise experiments may not necessarily perform equally well under real noise. This depends on what the real noise looks like, and whether the design of the algorithm itself has a certain robustness to the noise distribution. Because Gaussian noise is only an approximation and simulation of real noise, there is no guarantee that the designed algorithm must perform equally well when dealing with real noise. But the Gaussian noise test has a certain rationality, so this kind of algorithm will have a certain noise reduction function in the case of real noise.

· Deep learning, the specificity of noise reduction

For deep learning, the model itself is highly data-driven, not rule-based. In other words, the design of deep learning algorithms, or the design of network structures, does not strongly depend on the probability distribution of noise[47]. This is good for generalization of noise reduction algorithms. Deep learning algorithms generally require supervised training. In this way, the choice of training data does often depend on the probability distribution of noise compared with traditional methods, the robustness of deep learning algorithms is higher in processing from a specific noise, generalize to unknown noise.

2.3.2 Add Gaussian White Noise to the Raw signals

In communication systems, noise not caused by the channel is typically modeled as additive noise. That is

$$y(t) = x(t) + n(t) \quad (2-1)$$

where $x(t)$ is the signal, $n(t)$ is the noise, and $y(t)$ is the final received signal due to the noise. The model of additive noise refers to the fact that noise and signal are relatively independent, and there will be noise whether there is a signal or not.

Author also added gaussian white noise to raw samples to simulate the different SNR situations for MEMS-accelerometer Sensors in real working conditions. This process is achieved by combine OpenSeesPy module and another python program written by author.

Signal-to-Noise ratio is the logarithm of the ratio of signal power to noise power

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (2-2)$$

Usually, the signal acquired is discrete, and the power of the signal can be directly calculated. Suppose we have a discrete signal $S = \{S_1^1, S_2^1, \dots, S_k^1\}$, then, the signal power P_{signal} can be calculated as:

$$P_{\text{signal}} = \frac{1}{n} \sum_{k=1}^n S_k^2 \quad (2-3)$$

With the signal power, if we want the signal-to-noise ratio of the sample after adding noise to be SNR (such as 20 dB, 10 dB ...), then we can calculate the power of the noise:

$$P_{\text{noise}} = \frac{P_{\text{signal}}}{10^{\frac{SNR}{10}}} \quad (2-4)$$

We can first generate a standard Gaussian distribution (mean 0, standard deviation 1) of the noise sequence (same length as the signal), and then convert to get the final Gaussian noise we want. In the Python program, through the Rand of NumPy. Random module[48], author achieved the program to generate a standard Gaussian distribution sequence. A python program to verify if the noisy samples are added the wanted SNR level.

2.4 Damage Indicator Methods

In the environmental vibration experiment, since only the output data of the vibration response is available and the real input situation is unknown, the frequency response function or impulse response function of the structure cannot be obtained according to the classical method. In this situation the system identifies the system identification method with only output data. The modal analysis is called Environmental Vibration Modal Analysis, Output Data-Only Modal Analysis, or Operational Modal Analysis. In recent years, it has been unified into operational modal analysis. In operational modal analysis (OMA) for dynamic identification of output only vibration data [49], one of the most widespread and well acknowledged technique is the stochastic subspace identification (SSI).

The stochastic subspace identification is a time domain method based on the dynamic identification of Operational Modal Analysis. The method combines theories of system identification, linear algebra and statistics to identify dynamic systems from state-space equations through matrix calculations. Stochastic subspace method takes the linear discrete state space equation as the basic model, combines the input term and the noise term, and assumes white noise, uses the statistical properties of white noise for calculation, and obtains the Kalman filter state sequence, and then applies the least squares calculation system matrix to complete the identification process. Stochastic subspace methods are divided into covariance-driven random subspace methods and data-driven random subspace methods. The covariance-driven random subspace identification method firstly calculates the block Toeplitz matrix composed of the output covariance sequence, performs singular value decomposition (SVD) on the Toeplitz matrix to obtain the observable matrix and the controllable matrix, and then uses the observable matrix and the controllable matrix to obtain system matrix to identify the modal parameters of the system.[50]

Stochastic subspace method can not only identify the frequency of the system accurately, but also the mode shape and damping of the system, which makes the identification result more meaningful and practical.

2.4.1 State- Space representation

The SSI method relies on the following state-space representation [51]:

$$x_{k+1} = Ax_k + v_k \quad (2-5)$$

$$y_k = Cx_k + w_k \quad (2-6)$$

In 2-5 $x_k \in \mathbb{R}^n$ are the states, $A \in \mathbb{R}^{n \times n}$ is called state transition matrix, unmeasured environmental excitation is contained in the term v_k which is a Gaussian white noise sequence with zero mean and constant covariance matrix $Q = \mathbf{E}(v_k v_k^T) \stackrel{\text{def}}{=} Q\delta(k - k')$ with $\mathbf{E}(\cdot)$ denoting the expectation operator. Then in 2-6 $y_k \in \mathbb{R}^r$ is the outputs, $C \in \mathbb{R}^{r \times n}$ is observation matrix, the w_k terms indicate the measurement noise. Notice that n is the system order and r is the number of sensors.

2.4.2. Subspace-based damage indicators: subspace residuals

The feature vector is approximately assumed as zero mean normal Gaussian distributed in the reference state (undamaged) which evolves in a non-zero mean when damaged. Residual vector monitoring the relative changes among the damaged state, and the initial reference state. The present study mainly focused on the subspace-based residuals approach. When influenced by noise and changes in input excitation patterns, Subspace residuals are conventional and robust [36]. The residual matrix relies on the orthonormal property between the subspaces related to a reference state in comparison with the subspace related to a noisy or damaged state. Subspace-based residuals approach is based on the covariance-driven output-only subspace identification algorithm, so that they do not strictly require the identification process of the structure.

Cross-covariance between the states and the outputs $G = \mathbf{E}(x_{k+1} y_k^T)$, the theoretical output covariances $R_i = \mathbf{E}(y_k y_{k-i}^T) = CA^{i-1}G$, and

$$H_{p+1,q} \stackrel{\text{def}}{=} \begin{bmatrix} R_1 & R_2 & \dots & R_q \\ R_2 & R_3 & \dots & R_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ R_{p+1} & R_{p+2} & \dots & R_{p+q} \end{bmatrix} \stackrel{\text{def}}{=} \text{Hank}(R_i) \quad (2-7)$$

the theoretic block Hankel matrix. Using measured data $(y_k)_{k=1,\dots,n}$, a consistent estimate $\hat{H}_{p+1,q}$ is obtained from the empirical output covariances

$$\hat{R}_i = \frac{1}{N} \sum_{k=1}^N y_k y_{k-i}^T \quad (2-8)$$

$$\hat{H}_{p+1,q} = \text{Hank}(\hat{R}_i) \quad (2-9)$$

In [52], originally proposed a residual function by a comparison of the system reference state (undamaged) with the current one (damaged or undamaged). The system parameters in terms of eigenvalues and eigenvectors in the reference and current states are denoted respectively θ_0 and θ . In OMA SSI-COV method, the modal intrinsic properties information is obtained by the factorization property of the Hankel matrix which allows decomposing it in an observability matrix and a reversed controllability matrix. The

observability matrix is thus calculated starting from the Singular Value Decomposition (SVD) of the Hankel matrix:

$$\hat{H}_{p+1,q} \approx [\mathbf{U}_1 \mathbf{U}_2] \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{V}_1 \mathbf{V}_2]^T \approx \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1 \quad (2-10)$$

\mathbf{U}_1 represents the left active subspace of the independent column vectors of the Hankel matrix, whereas \mathbf{U}_2 denotes the null subspace of the independent column vectors of the Hankel matrix. Similar definitions are provided for \mathbf{V}_1 and \mathbf{V}_2 for row vectors of Hankel matrix. Finally, \mathbf{S}_1 collects the non-neglectable singular values in a diagonal matrix sorted

$$\mathbf{U}_1^T \hat{H}_{p+1,q} \mathbf{V}_1 \approx \mathbf{S}_1 \quad (2-11)$$

$$\hat{H}_{p+1,q} \mathbf{V}_2 \approx \epsilon_V \quad (2-12)$$

$$\mathbf{U}_2 \hat{H}_{p+1,q} \approx \epsilon_U \quad (2-13)$$

Residues ϵ_V and ϵ_U can be different from zero vectors because of noise effect or neglected weakly excited high modes. They are not the ideal candidates for tracking relative changes for damage detection purposes, when at same state. The amplitude of the residues, resulting from the erroneous definition of the system order or the amplitude excitation, may mask the residues variation due to small structural damages. Instead, the above property can be exploited by comparing two different states. Therefore, instead of using the null space $S(\theta_0^T)$ on the parameterized observability matrix, an empirical (nonparametric) null space S is computed on an estimated block Hankel matrix from data in the reference state using e.g., the SVD. The residue is thus expressed in matrix form as

$$\hat{\epsilon}_c = S^T \hat{H}_{p+1,q} \quad (2-14)$$

where S^T is the left null space of the block Hankel matrix $\hat{H}_{p+1,q}$ in the reference state and $\hat{H}_{p+1,q}$ is the covariance block Hankel matrix in the current, possible damaged one.

From the practical point of view, the excitation covariance Q may vary between different measurement sessions of the system because of random environmental factors, whereas the excitation is assumed to be stationary during one single measurement session. A change in the excitation covariance Q leads to a change in the cross-covariance between states and outputs G and thus in the Hankel matrix. In [35], the authors presented a new residual definition, which appears to be robust to variations of excitation. Let $\hat{\mathbf{U}}_1$ be the matrix of the left singular vectors obtained from an SVD of $\hat{H}_{p+1,q}$. Since $\hat{\mathbf{U}}_1$ is a matrix with orthonormal columns, it is independent of the excitation Q . Therefore, the residual matrix can be written as

$$\hat{\epsilon}_r = S \hat{\mathbf{U}}_1^T \quad (2-15)$$

2.4.3. Yan et al. damage indicator

In [35], the authors have provided a geometrical interpretation to the residual matrix concept, such as expression of a loss of orthonormality between reference subspace and another current state. Therefore, they give some damage indicators related to the rotation angle which arises between the two subspaces

when structural damage occurs. Finally, they established that the best-found damage indicator was given by the norm of matrix $\hat{\epsilon}_r$

$$\hat{\sigma}_N^2 = \text{norm} (\hat{\epsilon}_r) \quad (2-16)$$

where $\text{norm} (\bullet): R^{m \times n} \rightarrow R$ denotes the matrix spectral norm operator which corresponds to the maximal singular value of a matrix, from the numerical point of view. Therefore, the authors adopt the following damage indicators:

$$I_{y, nr} = \text{norm} (\hat{\epsilon}_c) \quad (2-17)$$

$$I_{y, r} = \text{norm} (\hat{\epsilon}_r) \quad (2-18)$$

In the current study, the above-mentioned proposed modern methodologies have been selected as a subspace-based damage-sensitive feature to train the Classification models presented in next chapter and validated.

3. CNN and ANN models

Deep learning models are built to research on the effect of different damage indicators under actual MEMS-accelerometer sensors and real-world working conditions. Both deep learning models and damage indicators are data dependent. Therefore, the dataset becomes mission critical. In next sections, author introduces different datasets for the research and two deep learning models for experiments, also briefly introduced the selection of some common model parameters in the machine learning field.

3.1 Datasets

This section details the process from real data for the structure to data at different signal-to-noise ratio levels, enabling simulation of real-world sensor acquisition datasets. Three different cases are introduced—that is, the extraction methods of three different damage indicators.

3.1.1 Raw Data

Finite Element (FE) beam model

As introduced in section 2.1, in the thesis, author implement the Beam model through python programming.

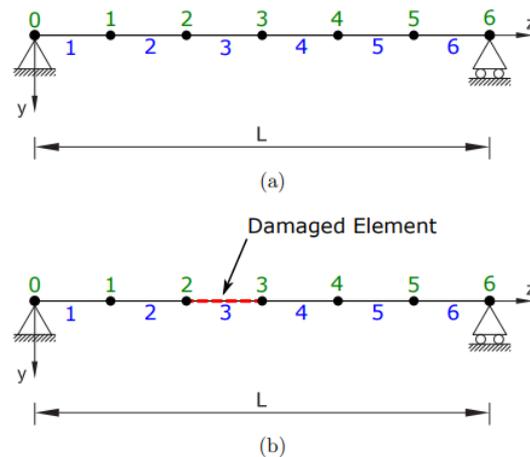


Figure 3.1. Finite Element (FE) beam model

As depicted in **Figure 3.1**, the beam element has been discretized with a uniform mesh, in which in each node the time history acceleration responses have been collected, simulating the presence of accelerometer sensors placed in correspondence of the nodes. Then, in the damaged model a severe damage (cross section reduction of 50%) or slight damage (cross section reduction of 25%) have been considered in specific fixed positions. The Beam has 7 nodes, marked as green dots 0 to 6. At the same time, point 0 and point 6 are fixed, so reasonable we assume there are five acceleration sensors on Beam, which are represented by green points 1 to 5. The acceleration responses have been collected in the finite element (FE) beam model nodes

to simulate a realistic monitoring system. Afterwards, the time histories analyses have been conducted through OpenSeesPy module considering a vertical acceleration history as input.

The input vertical acceleration history is a white noise sampled from a standard Gaussian distribution, with a sampling frequency of 50 Hz. The collected acceleration histories response data are simulating a monitoring setup composed of accelerometers with a sampling frequency of 500Hz. Five-minute acquisition duration sessions have been performed on damaged and undamaged cases. Finally, will have 150001 data for each sensor in each run. The data shape is shown in **Figure 3.2**.

Node 1	0.000000	0.000000	0.000000	0.000000	0.000000
Sensor 1	0.000000	-0.000001	0.000001
Sensor 2	0.000000	0.000004	-0.000003
Sensor 3	0.000000	-0.000018	-0.000000
Sensor 4	0.000000	0.000077	0.000039
Sensor 5	0.000000	-0.000113	-0.000071
Node 7	0.000000	-0.004278	-0.001656

7×15001

Fig 3.2 Shape of an acquired signals sample

Data Reproducibility

Since we used the Gaussian distribution in the collection of raw data, we will also use white Gaussian noise to get different SNR data. Because of the randomness of Gaussian distribution, in order to ensure that all data can be obtained repeatedly, we use the concept of seeds in random module in python[48]. That is, when the same seed is set, we can obtain the exact same data in any random case.

Dataset for reference state and current state

We need to set the reference state (undamaged) and the current state (undamaged or damaged) to compare the results.

◆ Reference state

We set the reference state as a completely undamaged situation.

◆ Current state

The current state is divided into three situations

1. Completely undamaged
2. Severe damaged, i.e., 50% reduction in cross-sectional area. As shown in **Figure 3.1(a)**, the damage position will be arbitrarily selected in the blue interval marked from 1 to 6, which can be damaged at a single position or multiple positions.
3. Slight damaged, that is, the cross-sectional area is reduced by 25%. The damage location and numbers of damage are the same as severe damaged

Figure 3.1(b) shows a damage situation where the damage occurs at the third part of the beam and the damage is severe.

Dataset visualization

Seeds are set up to ensure data reproducibility. Furthermore, to improve the accuracy of the result, we need to ensure that enough data is generated to train the Deep learning model. Therefore, we set seeds from 1 to 5000, that is, to generate 5000 runs of comparison data. Among them, if the current state is undamaged, to save storage space, the current state data is the same as the reference state data. When all reference state data are guaranteed to be undamaged, each seed only generates one current situation (0%, 25%, or 50% damaged).



Fig 3.3 Amount of data

Ultimately, we have an ideal sensor dataset described as the Beam model excited by a 50Hz white Gaussian noise and collected with a sampling frequency of 500Hz. The total amount of data is 8242 as in Figure 3.3 and a simple visualization of the data for one sensor is shown in Figure 3.4 .

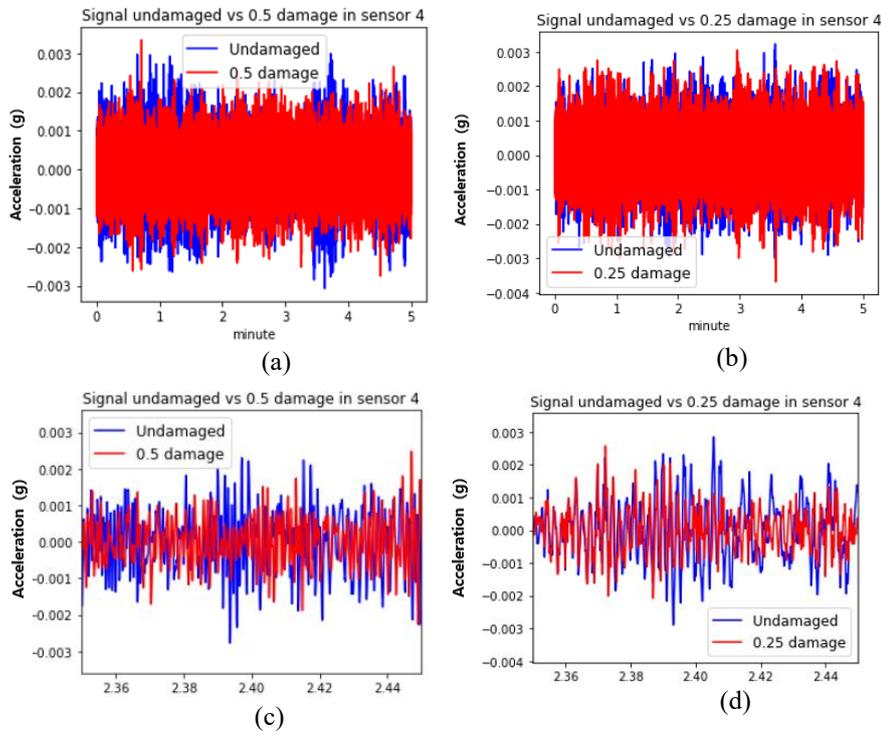


Fig 3.4 Data visualization for a certain sensor in different damage levels

3.1.2 Data from Real Sensors

In section 2.4, author selected MEMS acceleration sensors as target in this thesis. An important process is to simulate the raw signal to the real signal acquired by the MEMS accelerometer. In the field of structural health monitoring, SNR is not a fixed parameter of the sensors, since the actual working conditions of sensors are relatively harsh and the noise source is complex, it is difficult to determine the fixed SNR parameter in actual work. But according to [53] it is defined work perfectly when the SNR of the sensor is above 60dB. In addition, according to the [54], the current state-of-the-art MEMS sensors can guarantee between 27dB ~ 67 dB under ideal conditions.

Therefore, this experiment will set up three groups of different SNR levels for comparison. Set SNR levels to 20dB (low quality), 40dB (medium quality) and 60dB (good quality) for our MEMS sensors. Through three sets of comparisons, to verify whether the indicators and models we built can have better accuracy, robustness, noise reduction functions under different working conditions of real sensors. This aim is achieved through to the python program that has been introduced in Section 2.4.

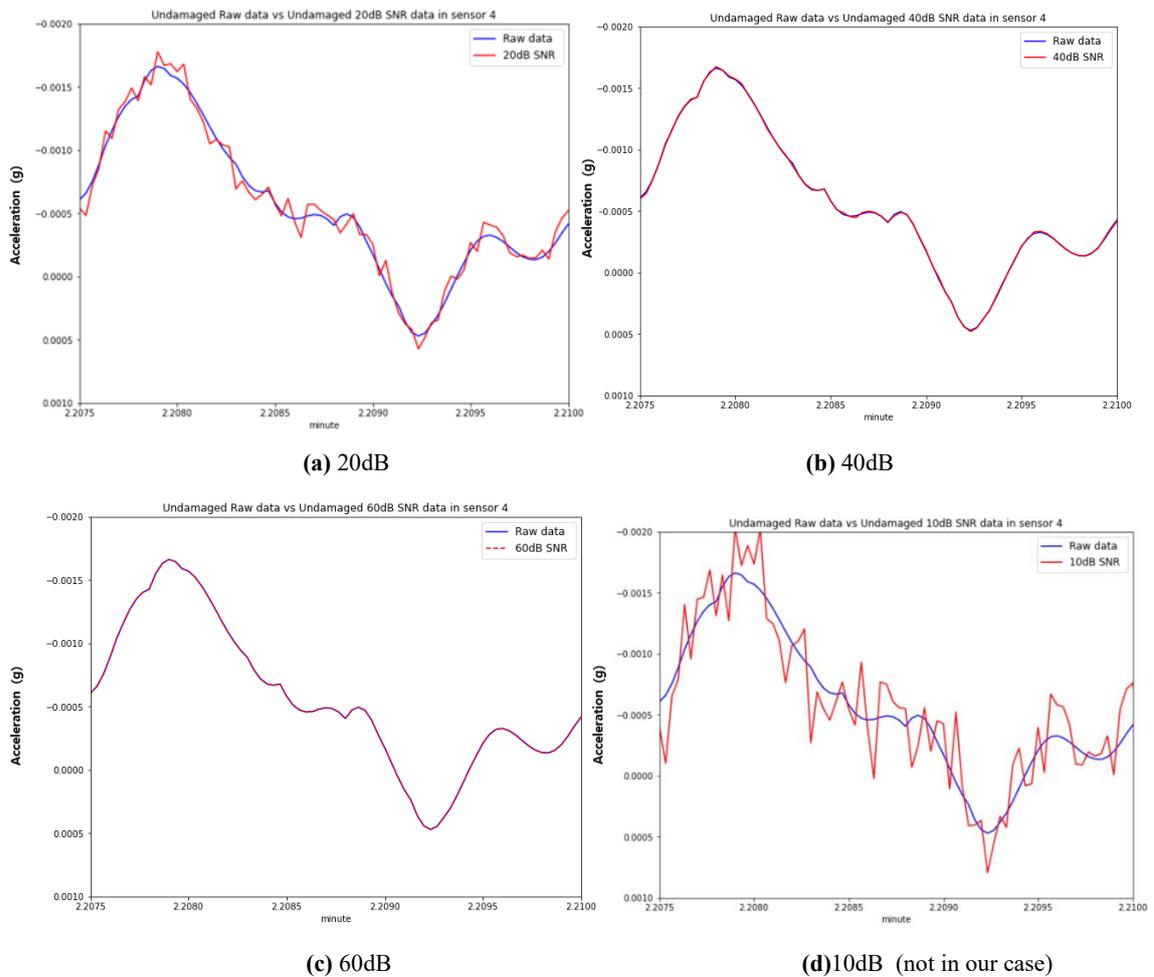


Fig 3.5 Data visualization for certain sensor in different SNR levels

3.1.3 Feature Extractions for Case A

After collecting these structural responses both for the reference and current states, for each acceleration record some statistical features was extracted. As demonstrated in [55], the following statistical features appear to be the most important ones which can be effectively considered by a machine learning model adopted to time series for SHM applications. From both acceleration time series recordings, denoted in general as x , the extracted features taken into consideration are the peak value $x_{P,(i,j)}$, the mean square value $x_{MS,(i,j)}$, the root mean square $x_{RMS,(i,j)}$, the variance $x_{VAR,(i,j)}$, the standard deviation $x_{STD,(i,j)}$, the skewness $x_{Skew,(i,j)}$, the kurtosis $x_{Kurt,(i,j)}$ and the K-factor $x_{K,(i,j)}$.

$$\begin{aligned}
 x_{P,(i,j)} &= \max\{|x_k|\}_{k=1}^n, & x_{MS,(i,j)} &= \frac{1}{n} \sum_{k=1}^n x_k^2, \\
 x_{RMS,(i,j)} &= \sqrt{\frac{1}{n} \sum_{i=1}^n x_k^2}, & x_{VAR,(i,j)} &= \frac{1}{n} \sum_{k=1}^n (x_k - \bar{x})^2, \\
 x_{STD,(i,j)} &= \sqrt{\frac{1}{n} \sum_{k=1}^n (x_k - \bar{x})^2}, & x_{Skew,(i,j)} &= \frac{1}{n} \sum_{k=1}^n \left(\frac{x_k - \bar{x}}{x_{STD,(i,j)}} \right)^3, \\
 x_{Kurt,(i,j)} &= \frac{1}{n} \sum_{k=1}^n \left(\frac{x_k - \bar{x}}{x_{STD,(i,j)}} \right)^4, & x_{K,(i,j)} &= x_{P,(i,j)} \cdot x_{RMS,(i,j)}.
 \end{aligned}$$

Subscript k denotes the k -th component of the time series vector x which has totally n elements. The subscript i denotes the i -th simulation out of the 5000 total runs, whereas j denotes the j -th accelerometer from which the time series has been recorded. Because of five minutes acquisitions have been recorded for each run with a sampling frequency of 500 Hz, n is thus equal to 150000. Since the number of accelerometers inside the beam domain is 5 (excluding the extremal support restraint points), as depicted in **Figure 3.1**. For each simulation acquisition, we accept the peak value $x_{P,(i,j)}$, the root mean square $x_{RMS,(i,j)}$, the variance $x_{VAR,(i,j)}$, the skewness $x_{Skew,(i,j)}$, the kurtosis $x_{Kurt,(i,j)}$ and the K-factor $x_{K,(i,j)}$. These 6 statistical features have been extracted from each accelerometer producing, in total, 30 extracted features. Remembering that for each simulation, two cases have been considered (current and reference), altogether, 60 features have been produced as input from each algorithm run.

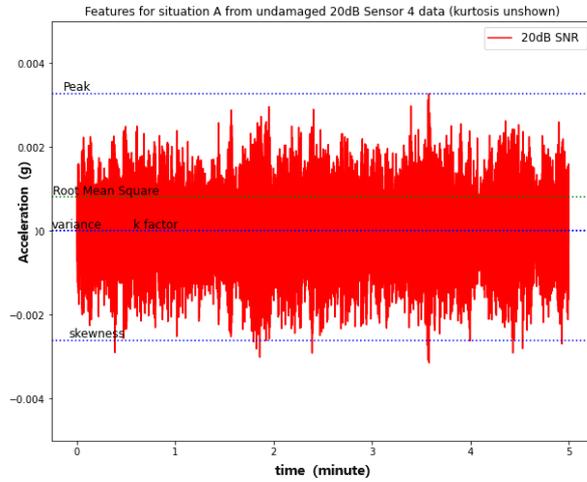


Fig 3.6 (a) Data visualization for Features for situation A

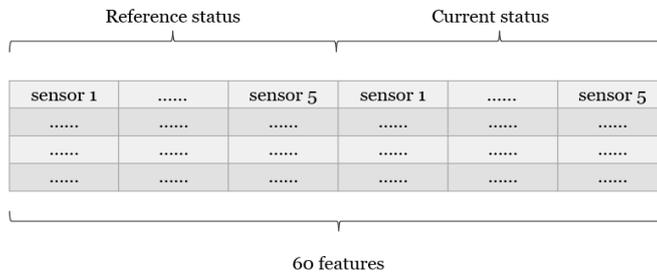


Fig 3.6 (b) Dataset structure for situation A

3.1.4 Feature Extractions for Case B

As showed in Section 2.2.1, Yan's et al. damage indicators do not require a prior OMA for dynamic structure identification. Rosso et al. [36] defined the user's choice parameters to get the most informative subspace-based damage indicator features. First is the choice of the active space dimension, consequently the null space dimension, and secondly the choice of time shift.

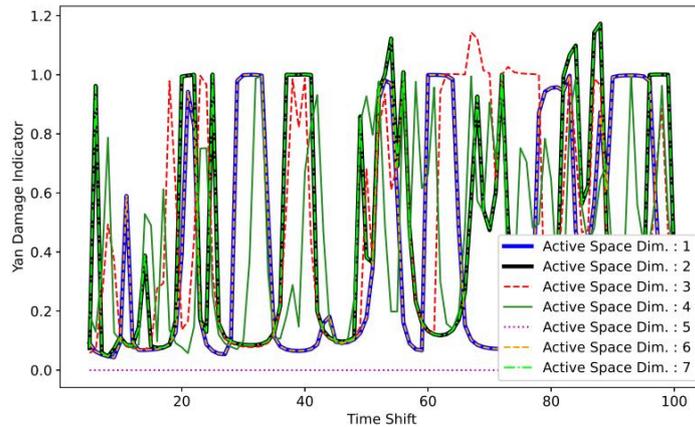


Fig 3.7 from [36] Sensitivity analysis on active space dimensions, compared with the time shift user's defined choice

Figure 3.7 is a great variation among the Yan et al. subspace-based damage indicator with different choices of both active subspace dimension and time shift. Firstly, the active space dimension is related to the space whose base is represented by the left singular column vectors associated with the singular values which are quite different from zero. It also shows the great variation among Yan et al. indicator with different choices of those 2 important parameters.

- **Active space dimension**

Active space dimension is related a space, whose base is represented by left singular column vectors with un-zero singular values. **Figure 3.7** demonstrated that when time shift is certain, Yan et al. indicators present different values with different active space dimensions.

- **Time shift**

Time shift related to the block rows considered in the block Toeplitz matrix assembling. According to [56], to identify the system order n , had to respect $li > n$, where l is number of sensors, i the number of block rows, i.e., the time shift. The literature proposed a good choice for the time shift, often based on power spectral density matrix

Figure 3.7 shows that the results of active space dimension 1 are almost same as dimension 6, whereas results of dimension 2 are same as dimension 7. Indeed, the results of this initial sensitivity analysis suggested adopting in this case an active space dimension equal to two which is consistent with the higher non-zero singular values. So, in this thesis, we the active space dimension 2 and time shift equal to 23(mitigate the computation effort).

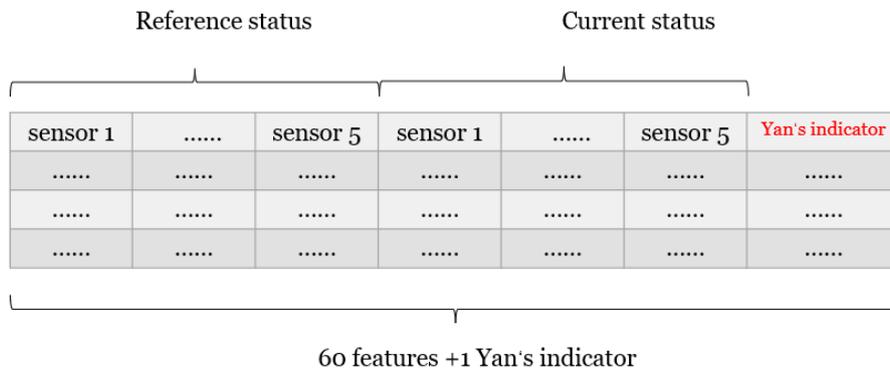


Fig 3.8 Dataset structure for situation B

3.1.5 Feature Extractions for Case C

1 Dimension 5 Time shift	4 Dimension 25 Time shift
.....
.....
.....

80 features × 5000 samples

Fig 3.9 Dataset structure for situation C

We already discussed the Yan et al. subspace-based damage indicator with different choices of both active subspace dimension and time shift in last session. In situation C we considering in input many Yan’s et al. damage indicators only. They have been calculated considering all the time shifts from 5 to 25 and truncation orders which define active space dimensions from 1 to 4, collecting in total 80 features for every one of the 5000 data. From **Figure 3.7** we can see these values to be reasonable intervals to detect all the main variations and hidden patterns in the subspace-based damage indicators, even containing the required computational effort.

3.2 CNN Model

3.2.1 1D-CNN

Many literatures focus on the use of 2-dimensional convolutional neural networks (2D CNNs), especially in image recognition applications. And 1-dimensional convolutional neural networks (1D CNNs) are only involved to a certain extent. CNNs are good at identifying simple features in data, and then use those simple

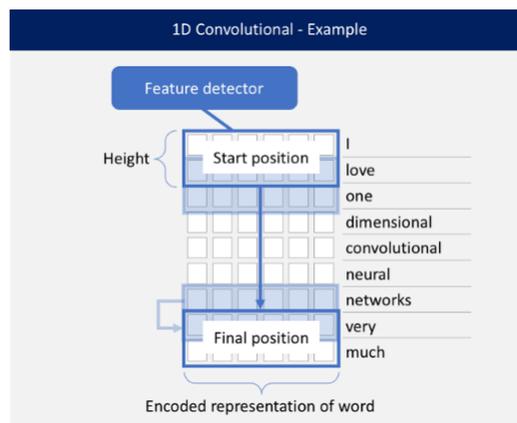


Fig 3.9 1-D CNN example[57]

features to generate more complex patterns in higher-level layers. 1D CNNs are very effective when we obtain a feature of interest from a short (fixed length) segment of the overall dataset, and the location of the feature in that data segment is not highly correlated. 1D CNNs work well for time series analysis of sensor

data (such as accelerometer data); they also work well for signal data with fixed length periods (such signals). In this example for natural language processing, a sentence is made up of 9 words. Each word is a vector that presents a word as a low dimensional representation. The feature detector will always cover the whole word. The height determines how many words are considered when training the feature detector. In our example. The height is based on our case (A), (B) or (C). In this example the feature detector will iterate through the data 8 times. Whether 1D, 2D, or 3D, Convolutional Neural Networks (CNNs) share the same characteristics and the same processing methods. The key difference is the dimensionality of the input data and how the feature detector (or filter) slides between the data. In this article, we will build on the features extracted from the accelerometer data, as we already discussed before, data come from numerical models, and the different damage indicators.

3.2.2 Build the CNN model

Our model is implemented in python through the TensorFlow and Keras modules. We will introduce the model for case C, as it is the most complex and representative one.

Table 3.1: Summary of the properties of the implemented CNN.

Layer	Output Shape	Activation Function	Parameters Number
Input and reshape layer	[(None,80,1)]	-	0
Conv1D_1	(None,77,15)	ReLU	75
Conv1D_2	(None,74,15)	Batch Normalization +ReLU	(915+60)
Max Pooling	(None,24,15)	-	0
Conv1D_3	(None,21,40)	Batch Normalization +ReLU	(2440+160)
Global Average Pooling	(None,40)	Drop out	0
Output Dense Layer	(None,3)	Softmax	123

Total Trainable parameters: 3773

Trainable parameters:3663

Epochs: 1000

Loss: Categorical Cross-entropy (Optimizer: Adam)

- **Input layer:** After the data is preprocessed, each data record contains 80 features. In order to prevent errors in the shape of the data, the first layer of the CNN network must guarantee that it is deformed into an 80 x 1 shape.

- **First 1D CNN layer:** The first layer defines filters (also called feature detectors) of height 4 (also called kernel size). Only by defining a filter, the neural network can learn a single feature in the first layer. This may not be enough, so we will define 15 filters. This gives us 15 different features trained in the first layer of the network. The output of the first neural network layer is a 77×15 matrix. Each column of the output matrix contains a filter weight. Each filter will contain 77 weight values, with the kernel size defined and the length of the input matrix considered.
- **Activation layer:** ReLU function. If the excitation function is not used (in fact, the excitation function is $f(x) = x$), in this case, the input of each layer of nodes is a linear function of the output of the upper layer, which is easy to verify, no matter how many layers your neural network has, the output is a linear combination of the input, which is equivalent to the effect of no hidden layer. In this case, it is the most primitive perceptron (Perceptron), then the approximation ability of the network is quite limited. For the above reasons, it is necessary to introduce a nonlinear function as an excitation function, so that the expressive ability of the deep neural network is more powerful (it is no longer a linear combination of inputs but can approximate almost any function).
- **Second 1D CNN layer:** The output of the first CNN will be fed into the second CNN layer. We will again define 15 different filters on this network layer for training. Following the same logic as the first layer, the size of the output matrix is 74×15 .
- **Batch Normalization layer:** BN not only standardizes the input layer, but also standardizes the input of each intermediate layer of the network (before the activation function), so that the output follows a normal distribution with a mean of 0 and a variance of 1, thereby avoiding variable distribution shifts problem. It is called batch normalization because during training, we normalize the input to each layer by only computing the mean and variance of a small batch of data in the current layer. It is equivalent to forcibly pulling the distribution of the input value of any neuron in each layer of neural network back to a standard normal distribution with a mean of 0 and a variance of 1.
- **Max Pooling Layer:** To reduce the complexity of the output and prevent overfitting of the data, a pooling layer is often used after the CNN layer. In our example, we chose a pooling layer of size 3. This means that the output matrix of this layer is only $1/3$ the size of the input matrix.
- **Third 1D CNN layer:** To learn higher-level features, another 1D CNN layer is used here. The output matrix after these two layers is a 21×40 matrix.
- **Average pooling layer:** Add an extra pooling layer to further avoid overfitting. Instead of taking the maximum value, the pooling this time takes the average of the two weights in the neural network. The size of the output matrix is 1×40 . Each feature detector has only one weight left in this layer of the neural network.
- **Dropout layer:** A dropout layer randomly assigns zero weights to neurons in the network. Since we chose a ratio of 0.5, 50% of the neurons will have zero weights. By doing this, the network is less responsive to

small changes in the data. Therefore, it can further improve the accuracy of unseen data processing. The output of this layer is still a 1 x 40 matrix.

- **Fully connected layer with SoftMax activation:** The last layer will reduce a vector of length 40 to a vector of length 3, since we have 3 classes to predict (i.e., ‘undamaged’, ‘slightly damaged’, ‘damaged severely’). The dimensionality reduction here is done by another matrix multiplication. SoftMax is used as activation function. It forces the sum of all output values of the neural network to be one. Therefore, the output value will represent the probability of occurrence of each of these three categories.

3.3 MLP Model

Our MLP model is implemented in python through the TensorFlow and Keras modules. We will introduce the model for case C, as it is the most complex and representative one.

Table 3.2: Summary of the properties of the implemented MLP model.

Layer	Output Shape	Activation Function	Parameters Number
Input and reshape layer	[(None,80)]	-	0
Hidden layer	(None,15)	ReLU	930
Output Dense Layer	(None,3)	Softmax	48

Total Trainable parameters: 978

Trainable parameters:978

Epochs: 1000

Loss: Categorical Cross-entropy (Optimizer: Adam)

- **Input layer:** After the data is preprocessed, each data record contains 80 features. In order to prevent errors in the shape of the data, the first layer of the ANN network must guarantee that it is deformed into an 80 x 1 shape.

- **Hidden fully connected layer with ReLU activation:** It is necessary to introduce a nonlinear function as an excitation function, so that the expressive ability of the Perceptron is more powerful (it is no longer a linear combination of inputs but can approximate almost any function). This is the difference between multilayer perceptron and perceptron.

- **Fully connected layer with SoftMax activation:** The last layer will reduce a vector of length 15 to a vector of length 3, since we have 3 classes to predict (i.e., ‘undamaged’, ‘slightly damaged’, ‘damaged severely’). SoftMax is used as activation function. It forces the sum of all output values of the neural network to be one. Therefore, the output value will represent the probability of occurrence of each of these three categories.

3.4 Selection of Common Parameters

Some commonly used parameters are from the Keras.[58]

Loss function

Categorical cross entropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.

Metrics

A metric is a function that is used to judge the performance of your model. Metric functions are similar to loss functions, except that the results from evaluating a metric are not used when training the model. Note that you may use any loss function as a metric. In multi-classification problems, the main metrics are validation accuracy, Recall and Precision. Specific metrics definitions are discussed in the next chapter

Optimizers

Optimizer is to guide the parameters of the loss function to update the appropriate size in the correct direction in the process of deep learning backpropagation, so that the updated parameters make the loss function value continuously approach the global minimum. Adam is an effective stochastic optimization method that only requires first-order gradients and requires only a small amount of memory. It is an excellent adaptive learning optimizer under default working conditions.

4. Outcomes and Conclusion

The (A), (B) and (C) above-mentioned cases have been implemented to research on the noise effect of the three-class classification problem. There have three different SNR levels (20dB,40dB,60dB), with three different damage situations, considering different datasets collected on a numerical simulation Beam performed both on a reference state and a current one. A popular assess of the performance on Deep Learning model is to divide the entire dataset into two parts: the training set which is employed to train the model permitting to learn the optimal weights, and a test set which allows testing the predictive capabilities of the trained model to evaluate generalization error [56].A further partition inside the training set determines the so-called validation set which allows evaluating accuracy performance during the training phase among the epochs. In the current study, a common way that 80% of the dataset split has been set for the training set and a 20% to the test set. The validation set size has been set to 10% of the training set.

4.1 MLP and CNN Multiclass Classification Results

The performance of the best trained MLP model on different SNR levels has been validated with the test set, whose classification results have been condensed in the confusion matrix illustrated in **Figure 4.1**. Meanwhile, **Figure 4.2**. shows the results from the best CNN model on same datasets. The accuracies are all above 90% for MLP models but decrease to 82.80% for CNN models in the case(A) and (B). The accuracy measures the portion of the test set which has been correctly classified (the sum of main diagonal terms) out of the entire 1000 samples. Two other metrics are presented in the confusion matrix: precision and recall. The precision measures the number of samples correctly classified in a certain class over the total number of samples which have been associated with that class, whereas the recall represents the number of samples correctly classified to a certain class over the number of samples which actually belongs to that class [56]. Focusing on a certain class (column), the precision evaluates how the model performs, concerning when it associates always that class even if the true class was another one (false positives). Instead, focusing on a certain true class (row), the recall evaluates the predictor performance in terms of correctness of classification with respect to the ground-truth, i.e. the actual number of elements which have supposed to belong to that class and even considering the so-called false negative. In this case, both the precision values and the recall values are quite high, above the 87.69% for all the classification possible outcomes in MLP models. But we can see a significant decrease on CNN, even to 65.58%.

Let's take CNN models in 60dB SNR levels for example. The performance of the trained model (A) has been validated with the test set, whose classification results have been condensed in the confusion matrix illustrated in **Figure 4.2 (g)**. The overall accuracy obtained is about 85.50%. The performance of the trained model (B) has been validated with the test set and the result in **Figure 4.2 (h)** shows the accuracy is about 87.20%, As expected the presence of the most informative Yan's et al. damage indicator with the statistical feature inputs provides better classification performance to the trained model. The higher classification of the method (B) with respect to method (A) proves that this most informative subspace- based feature

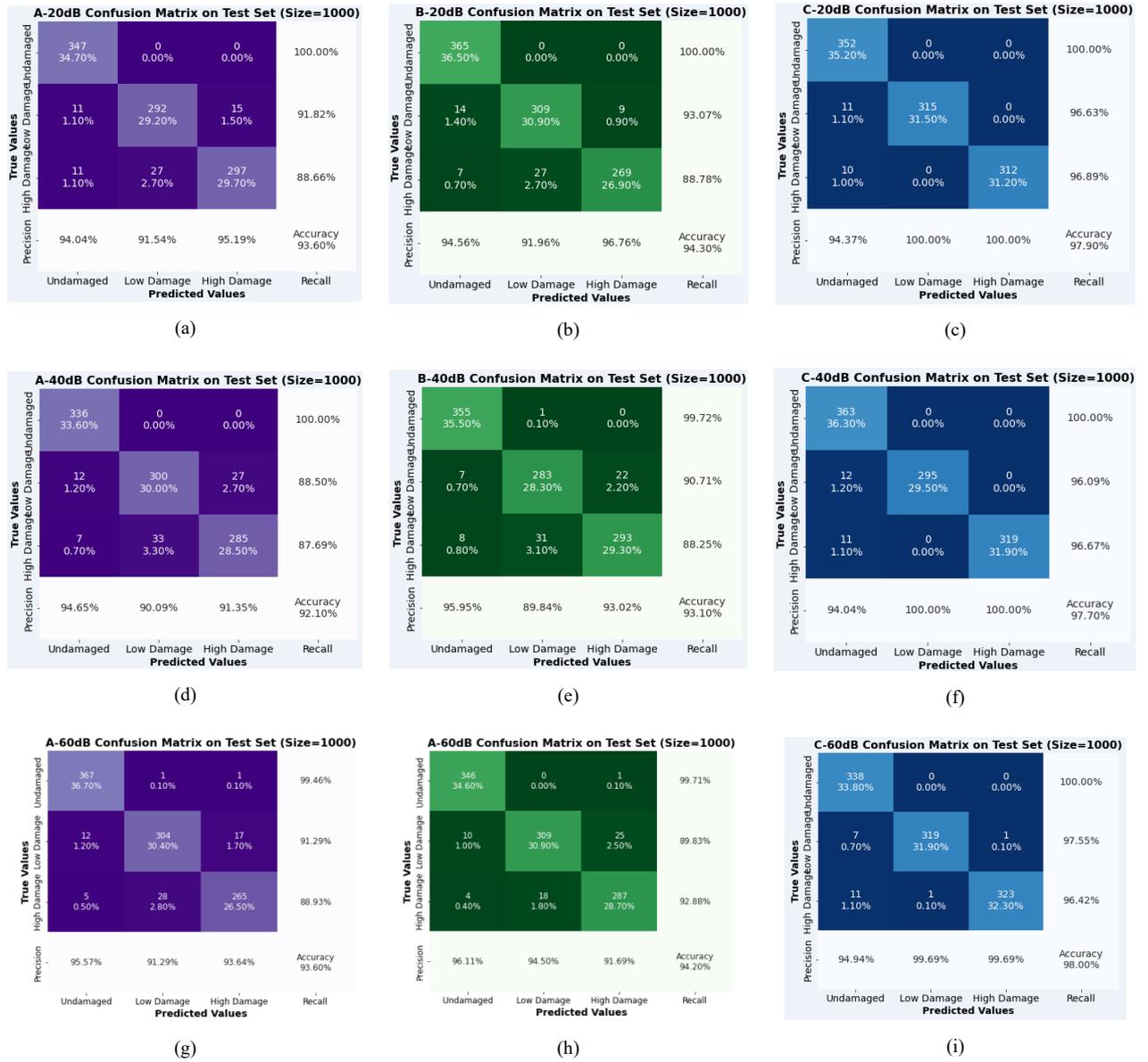


Figure 4.1. classification results of the MLP models

improves the classification performances of both CNN and ANN model in Level 1 of the SHM task in all noise levels. In Figure 4.2(i), for trained model in case(C), a final accuracy is 98.60%.

As expected, the presence of Yan's et al. damage indicators provide better classification performance to the trained model with respect to the previous cases. The outstanding higher classification performance of the method (C) with respect to the methods (A) and (B) proves that considering an entire set of informative subspace-based features remarkably improves the classification performances of the deep learning models for the damage detection task. Furthermore, the advantage of method (C) is that removes the arbitrary

choice of the user about governing parameters in the subspace-based damage indicator calculations.

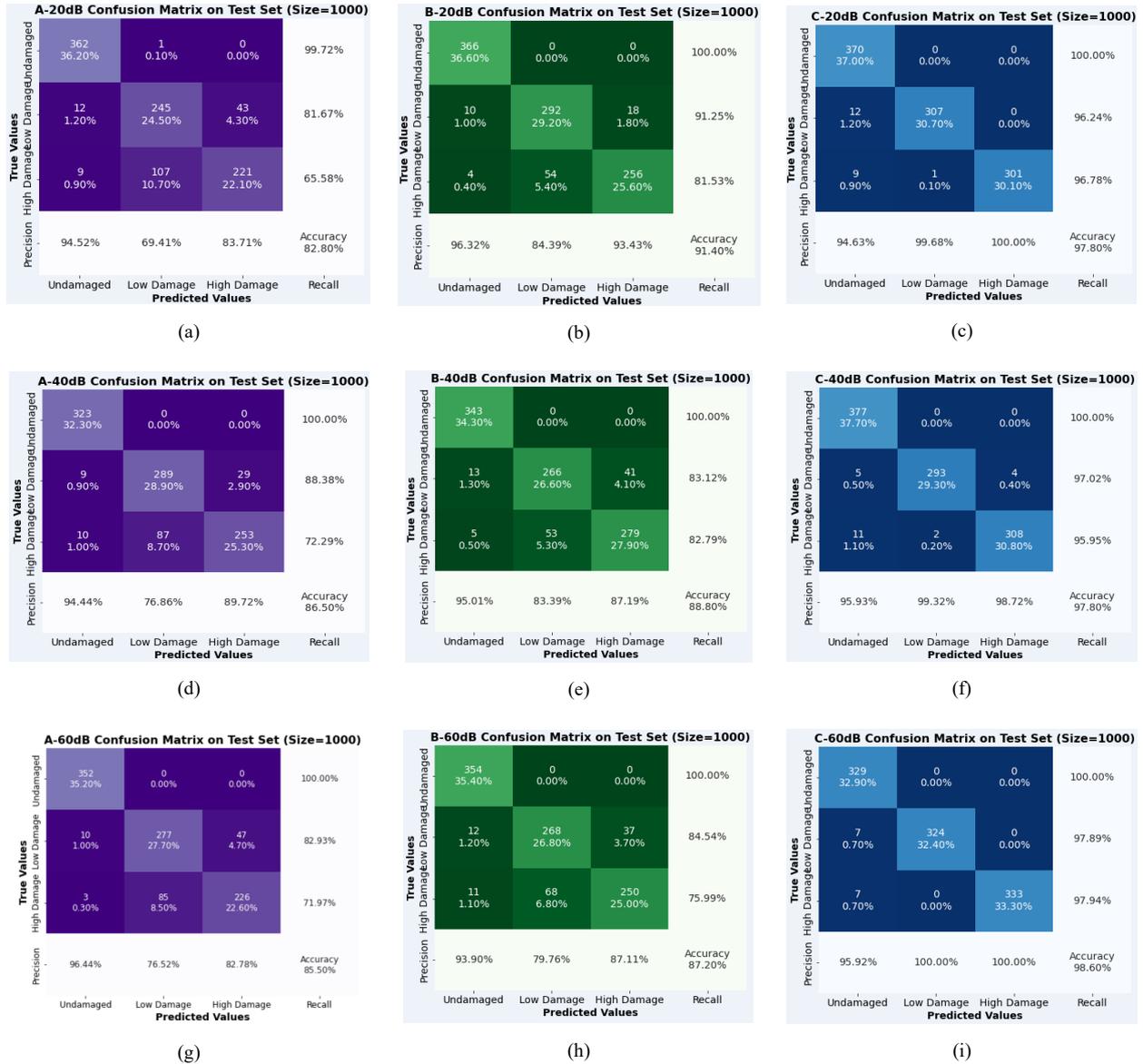


Figure 4.2. classification results of the CNN models

We had to save the best weights and models for prediction. The metrics to choose the best model is validation accuracy. The loss function both for the training and the validation sets is depicted over the epochs, for our CNN model in 60dB SNR levels is reporting on Figure 4.3. It is worth noting that when the validation loss starts to increase after a monotonic decreasing behavior, at that point the overfitting of the model is reached. In Figure 4.3(a), the validation loss is increase gradually, while the loss on training set is decreasing, this means we reached the overfitting point at around 400 epochs. The same situation

appears on **Figure 4.3(b)**, while the overfitting point is around 450 epochs. But it's worth noticing that the validation accuracy is a little bit increasing even when overfitting point reached. In **Figure 4.3(c)**, we don't evident the overfitting, actually the loss is rather low. This also proves the powerful of Yan's et al. damage indicators in our tasks. From a deeper insight into the loss and accuracy trends, it would virtually be possible to stop the training to epoch 150 to save computational cost and obtain almost the same performances.

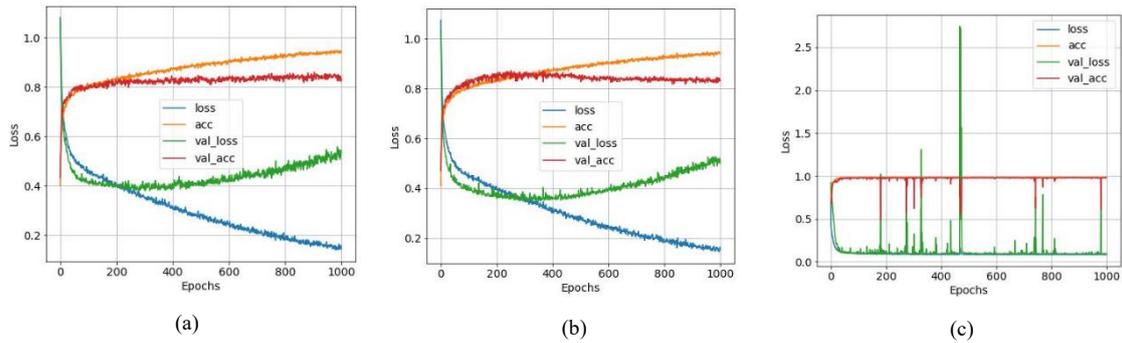
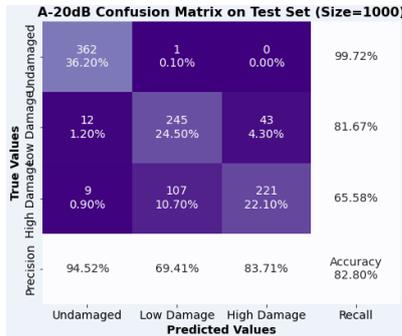


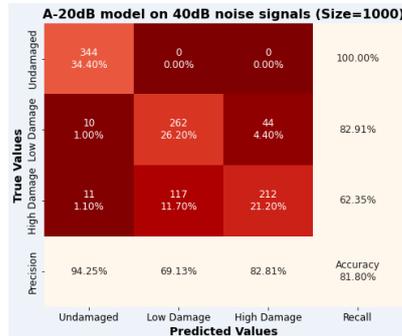
Figure 4.3 60-dB noise level CNN training performances history during the epochs.

Author sets up further experiments. The deep learning models trained at different signal-to-noise ratio levels will be cross-tested to further verify the robustness of the deep-learning models to noise effect. During the experiments, we adopted the best CNN models trained at 20dB SNR level and 60dB SNR level. **Figure 4.4.(a) (d) (g)** respectively show the performance of the best CNN models in method (A) (B) (C) as we already discussed before. Let's focus on the performance on method (A). The performances of the CNN model trained from 20dB SNR level has been validated with the test set from 40dB SNR level, whose classification results have been condensed in the confusion matrix illustrated in **Figure 4.4 (b)**. The overall accuracy obtained is about 81.80%. In **Figure 4.4(c)**, an accuracy is about 80.50% for the test set from 60dB SNR level. The same outcomes can be drawn even for method (B) and (C). It is obvious that even if the noise level has changed significantly, we can still make predictions with the previously trained model. Results did not change significantly

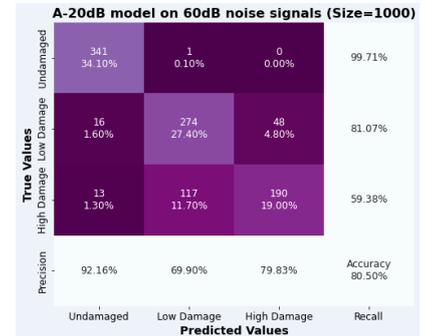
The current MLP and CNN models can provide quite interesting multiclass classification results considering the statistical time series features coupled with the Yan's et al. subspace damage indicator, extending the capabilities of the MLP model trained in [36]. Furthermore, a good generalization of the current deep learning models is related to the fact that the 5000 numerical simulations randomly considered both how many damaged elements to consider (none or all) and the level of damage to associate to those selected elements. This produced time-series signal which covers many different cases, which was anyway successfully traced back to three possible classification results: undamaged situation, low damage status (cross section reduction about 25%) and high damage condition (cross section reduction about 50%). The influence of noise is further reduced. Deep learning models are somewhat robust to noise.



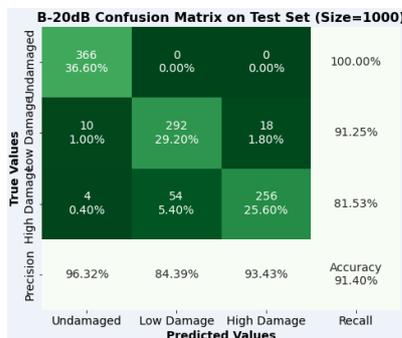
(a)



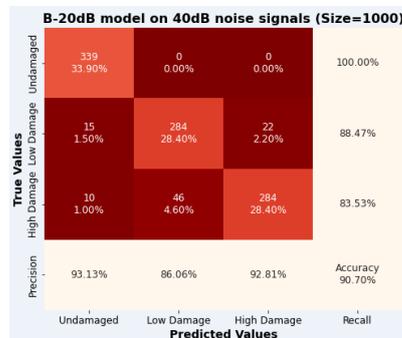
(b)



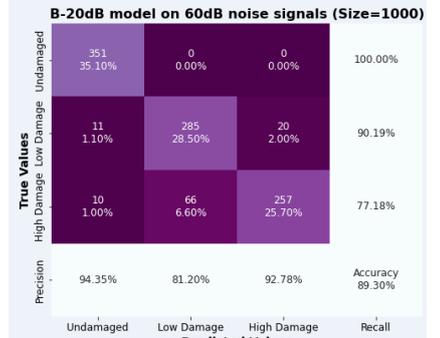
(c)



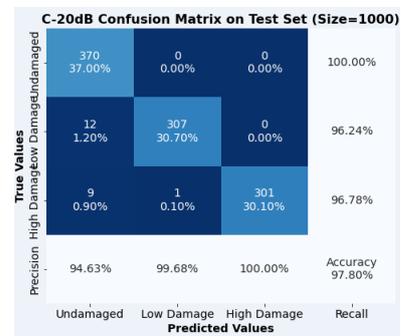
(d)



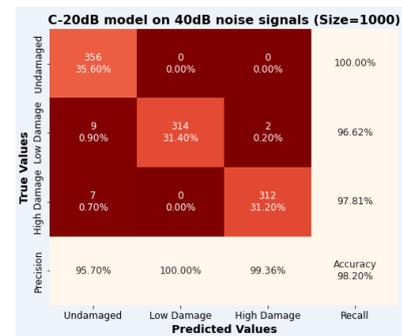
(e)



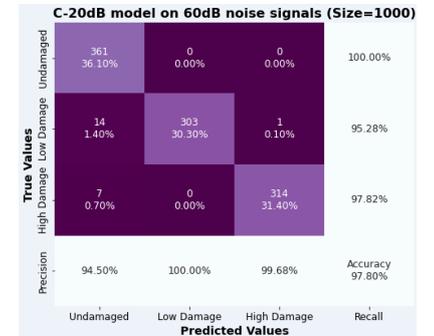
(f)



(g)



(h)



(i)

Figure 4.4. classification results of the 20dB CNN models on other SNR levels in all three cases

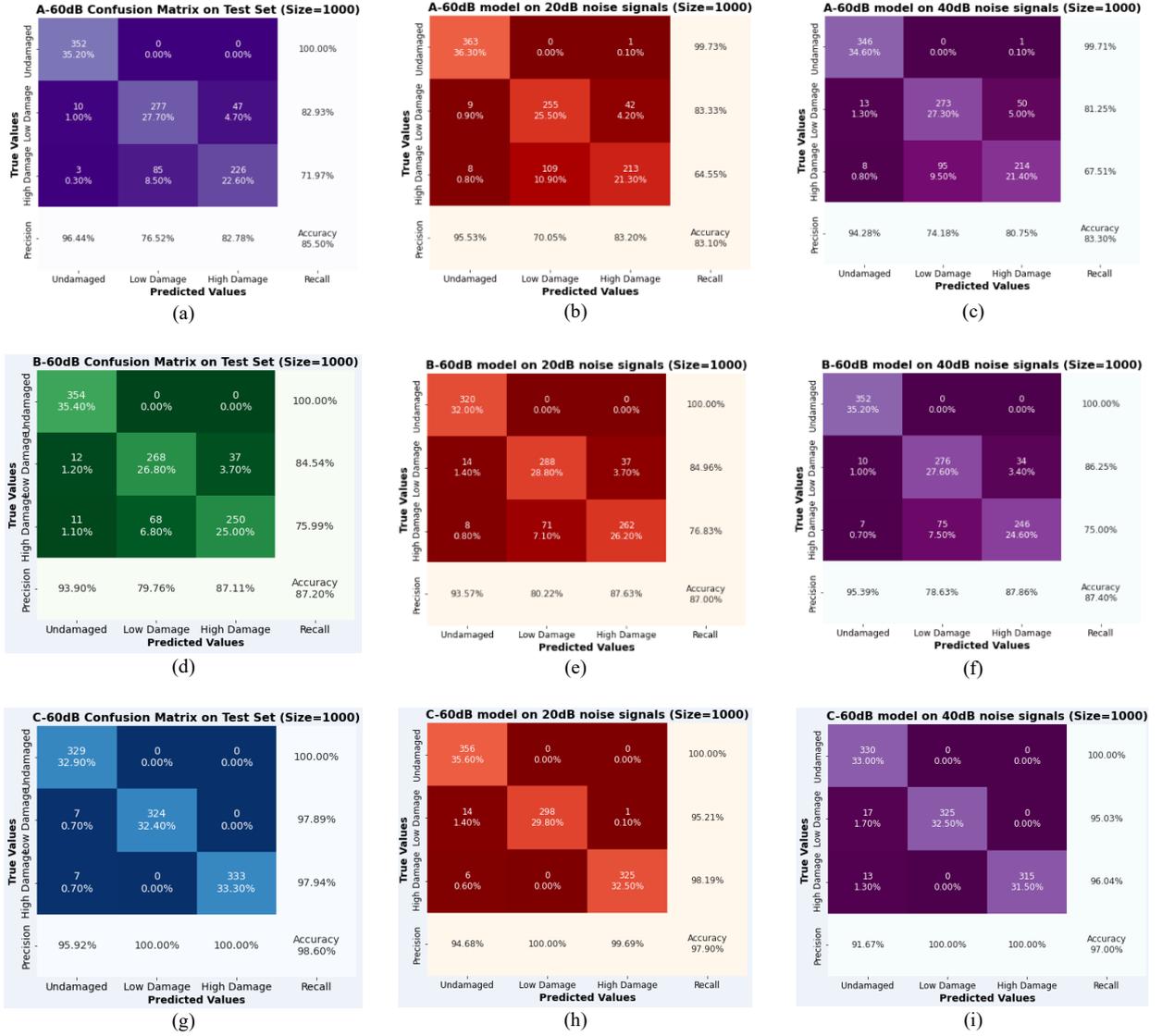


Figure 4.5. classification results of the 60dB CNN models on other SNR levels in all three cases

4.2 Discussion

The methods proposed in this thesis to fulfil the Level 1 of SHM approach related to damage detection task have been implemented by different Deep Learning models combined with powerful subspace-based damage indicators. Because of their major sensitivity to the presence of any structural damage, these indicators allow improving performance of multiclass damage classification of our architectures which initially considers statistical features only as input, such as in [55]. In this section, some comparisons among the proposed methods (A), (B), (C) on different SNR levels are discussed.

4.2.1 Comparisons between different features

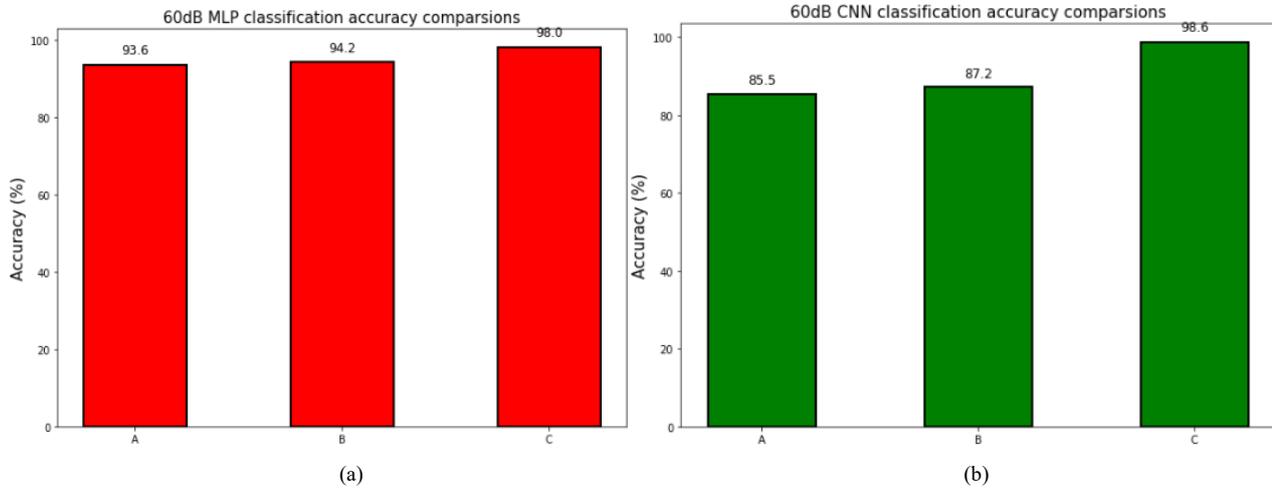


Figure 4.6. 60dB models classification comparisons on accuracy among the proposed methods (A), (B), (C)

As illustrated in **Figure 4.6**, the bar graph compares the overall accuracy on three different features obtained from the above-mentioned methods. These results are retrieved from the numerical Beam case which enables performing a novel and extensive empirical sensitivity analysis on parameters which affects the subspace-based damage indicators. The key point is that we also considering noise and error sources which may virtually affect a real-world case. We set different SNR levels to simulate real noise effects on sensors then calculate different datasets. For CNN models in **Figure 4-6 (b)**, The method (A), which considers the statistical features only, leads to a quite low accuracy level about 85.50%, but the accuracy classification performances increased a little bit to 87.20% when in the method (B), the most informative subspace-based damage indicator has been added to the inputs. This most informative indicator was obtained by setting the optimal parameters (time shift, active and null subspace dimensionalities) from the results of the empirical sensitivity analysis in **Figure 3.7**. Indeed, the association of statistical features with more damage sensitive ones have produced a little bit better results than considering statistical features alone. This improvement can be seen in all cases (**Figure 4.1** and **Figure 4.2**). Outstanding accuracy performance results have been obtained with the method (C), which reaches about 98.60%. The main advantage of the method (C) is related to the fact that it considers as an input dataset many subspace-based damage indicators only, removing the user arbitrary choice of the parameters which affect the damage indicator calculations. Therefore, this last proposed method (C) represents the most useful approach to effectively address the damage detection task.

Further comparisons have been performed in terms of recall and precision as illustrated in **Figures 4.7**. Recall is a very important metric in our case because we want all structural damaged to be detected. Through comparison, it is found that the improvement of the accuracy of case (C) is mainly reflected in the distinction of damage degree. In other words, in case (C) all damaged structures can be found, while in the

identification of undamaged structures, the precision is not much different from case (A) and case (B). Focusing on recall metric, for every of the considered methods

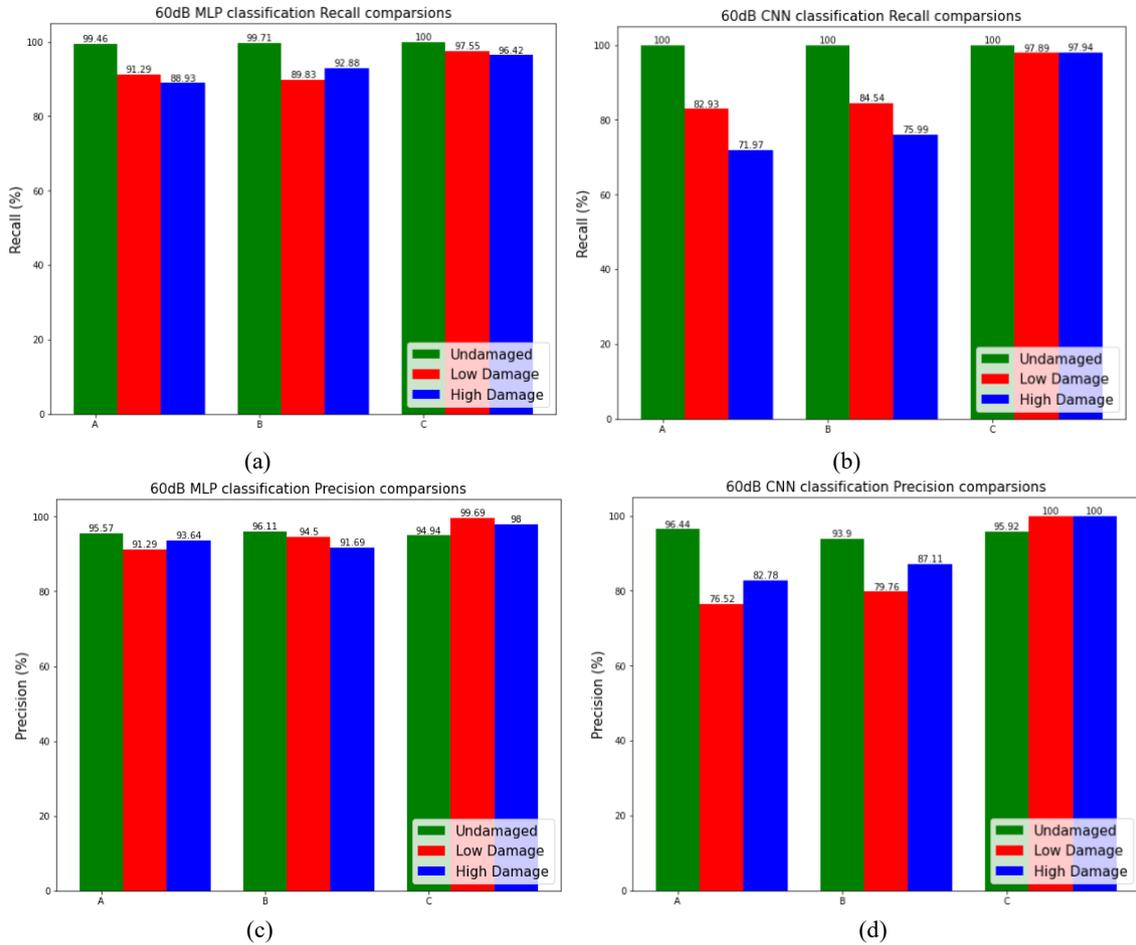


Figure 4.7. 60dB models classification comparisons on Recall and Precision among the proposed methods

the undamaged case has been always correctly classified with respect to the belonging class number ground truth (more than 90%). These results virtually reflect the fact that the subspace-based damage indicator value tends to be close to zero when an undamaged situation occurred both in reference state and current state. Therefore, the models have been probably able to learn this characteristic and exploit it to enhance its classification performances when dealing with undamaged cases. On the other hand, method (C) reaches the highest recall both in low damage and high damage classes which is higher than 97.94%. These shows advantages to consider the subspace-based damage indicator with respect to the statistical features alone.

On precision metric, it is clear from the graph that the remarkable precision values have been obtained in the method (C) for which 100% have been reached for low damage and high damage in CNN models. This fact evidence how the CNN has not misclassified the inputs which truly belongs to these two latter classes.

Indeed, the CNN models has probably learnt optimal weights to effectively reconduct low and high damage cases and only a few of the input has been misclassified as undamaged.

4.2.2 Comparison Between CNN and MLP models

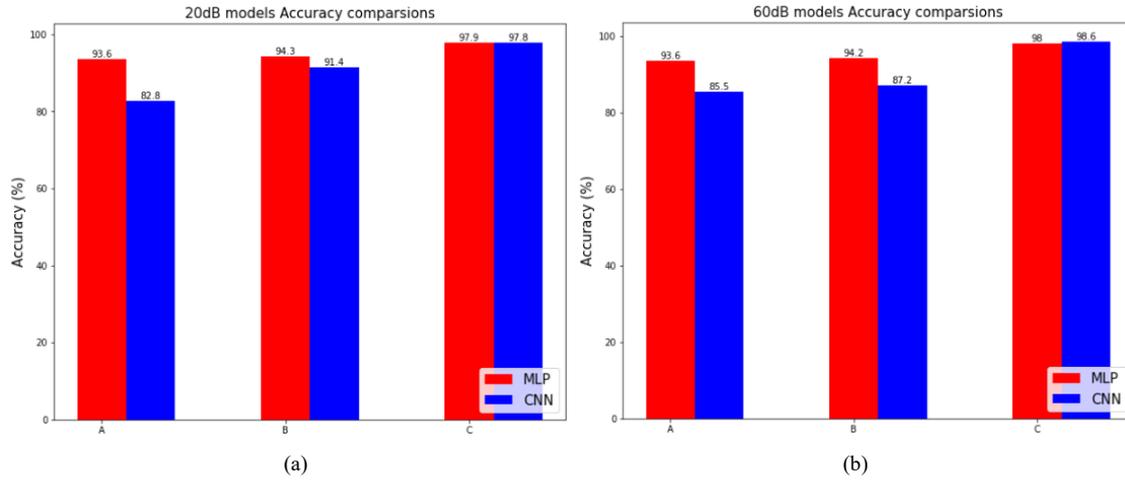


Figure 4.8. 20dB and 60dB models classification comparisons on accuracy among the proposed methods

Figure 4-8 shows the comparison of the overall accuracy between the two models - the CNN model and the ANN model on different noise levels. Regarding the accuracy performance of 1D-CNN in case (A). Let's focus on the 60dB models first, from Figure 4.8(b), the accuracy of CNN model in case(A) is 85.5%, which is clearly decreased from 93.6%, the accuracy from the MLP model in same case. The same happens in case (B), where the accuracy of the CNN model is 87.2%, a slight improvement over case (A) (discussed in the previous section), but still lower than the accuracy 94.2% of the MLP model. This situation occurs due to the reason that, CNN is a technology mostly used in Natural Language Processing and Image Processing. 1D-CNN is often used for raw signals from sensors, whose adjacent features are correlated in some extent (even from visualization). In case (A) and (B), the features we use are discrete statistical features, and there is no obvious connection between adjacent features. Therefore, the fully connected layer used by MLP may be able to see the overall features better than the convolution kernel of CNN, to obtain the final classification result. But there is still a very impressive accuracy in classification case (C). The accuracy of MLP reached 98%, while the accuracy of CNN reached 98.6%. This means from a very different method comparison proves the effectiveness of Yan's et al. damage indicators in SHM for real sensor datasets. Same results on 20dB-SNR(Figure 4.1-4.2 (a)(b)(c)) models. On 20dB SNR level, 1D-CNN model didn't perform as well as MLP on case (A) and (B). (82.80% vs 93.60%, 91.40% vs 94.30). Also, the main differences occur in distinguishing the degree of damage. Both MLP and CNN perform well in the case(C) with different noise levels (all around 98%), demonstrating the robustness of the SSI method, Yan's et al. damage indicators is a very powerful and effective indicator in the SHM field even in different noise levels and in different Deep Learning models.

4.2.3 Noise Effects

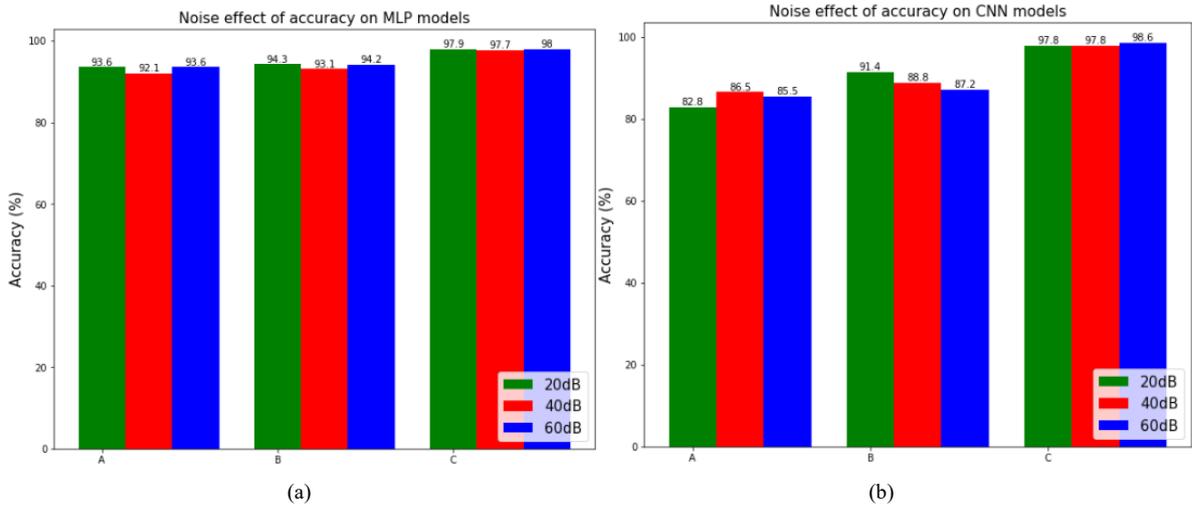


Figure 4.9. 20dB,40dB and 60dB models classification comparisons on accuracy among the proposed methods

Figure 4.9. illustrate the influence of different noise effect on different cases. For CNN models, in case (A), the final accuracy on 20dB SNR level (High noise level or worse signal quality) is 82.8%, while on 40dB (Medium quality) is 86.5% ,and 85.5% on 60dB (Good quality).The outcome is 91.4%,88.8% ,and 87.2% on case (B) and a rather better 97.8% ,97.8%, 98.6 % on case (C).There are indeed fluctuations in the results, but actually no significant difference between the outcomes from three different noise levels in any case, for both CNN and MLP (**Figure 4.9. (a)**) model.

These noise effects are reasonable to be consider the real and complex case when our structure is working on the real MEMS -accelerometers with real-world working conditions. It can be concluded that for real sensors, Yan’s et al. damage indicator is still an effective indicator in SHM. Also, the **Figure 4.9** shows both Yan’s et al. damage indicators and Deep Learning models have robustness to noise. First, deep learning is a data-driven technique, and many literatures in computer science have proven that deep learning itself has a certain ability to reduce noise effect (from real sensor errors and environmental influences). In our research, whether it is low noise level or very high noise level, using deep learning tools under different features conditions, we can get approximate experimental results. In addition, the Stochastic Subspace Identification method directly processes and analyzes the response data of the structure in the time domain and uses the theory of spatial projection to eliminate the noise signals that are not related to the response data and has a good ability to resist noise disturbance. In the three cases (A) (B) (C), Yan’s et al. damage indicator has the best performance on accuracy regardless of noise levels and deep learning models. As shown, in case (C), every classification accuracy is greater than 98%.

Best model’s robustness to noise

Figure 4.10. shows the performance of the best model trained in the higher noise condition (20dB SNR level) and the low noise condition (60dB SNR level) in the face of other noise levels respectively. We can see that, taking the result of the 20dB model as an example (**Figure 4.10(a)**). In case (A), the 20dB SNR

level best trained model has the highest classification accuracy 82.8% at 20dB SNR level, while at 40dB SNR level it is reduced by 1% to reach 81.8%. At 60dB SNR level the accuracy dropped again and reached 80.5%. Such results also appear in case (B) and case (C). In addition, the same experiment is performed on the training results under the 60dB model (**Figure 4.10(b)**), and we

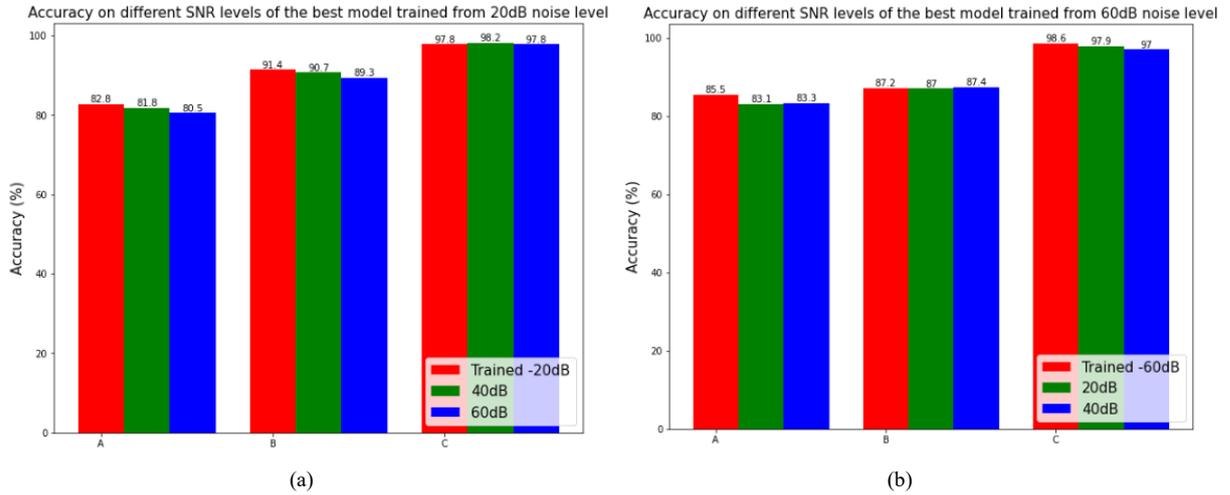


Figure 4.10. Accuracy on different SNR levels of the CNN models trained from 20dB and 60dB SNR levels

can see that, there is not a very large differences, best model has nearly the same performance at different signal-to-noise levels. This is very meaningful. Our research is based on the data obtained from the previously simulated beam model and used to approximate the real-world noise effect. In theory, for different working environments, retraining the models every time based on the collected data can achieve the best results. However, in practice, retraining the models will take a long time and higher cost; in addition, in the real world, although the error of the sensor changes slowly, the environmental noise changes rapidly. For example, in the real working environment of a bridge, there may be a large change in the signal-to-noise ratio within a period with the change of the flow of people, the flow of vehicles, and the time. Therefore, the robustness of the best model has practical significance for improving the generalization ability.

4.2.4 Conclusion

In conclusion, the main relevant results obtained from the current study can be summarized as follows:

- **For (A)(B)(C) method:**

- **Method (A)** which only considers statistical features performed worse than other methods which included subspace-based damage indicators as input of deep learning method.

- **Method (B)** consistently exploiting information contained in the subspace-based damage indicator and statistical features. This result proves the robustness of the proposed method concerning misleading user's parameters choices. Compared with method (A), method (B) has a slight improvement, but still needs to consider the random selection of some parameters

- **Method (C)** exhibited the best performance concerning the other analyzed methods. This method has the advantage of removing the user's arbitrary choice of the parameters which affect the damage indicators calculations.

- **For ANN and CNN method:**

- **CNN models:** When in case (A) and case (B), it has a greater disadvantage than ANN because of the nature shape of the datasets. However, in case (C), the CNN approximates and surpasses the MLP performance, thus to some extent demonstrating the strong effectiveness of Yan's et al. damage indicator.

- **ANN model:** The ANN model has a simple structure, excellent results, and fast running speed. For the current working situation, it is the model we should accept.

- **For noise effect:**

- **CNN and ANN models** can obtain almost similar results under different noise levels, even in harsh noise environments. From both the results and some literatures, we verify the robustness of the deep learning model to noise

- **Different statistical parameters** will make the results vary greatly. In particular, the damage index based on the SSI method proposed by Yan et al, after combining with deep learning tools, can obtain more than 95% accuracy in various noise environments, and has considerable anti-noise ability.

- **The best models** trained by deep learning have good generalization ability. Whether under low or high noise conditions, the trained model is able to obtain similar results to the training noise level at any other test SNR level. This has good promotion and practical significance.

4.3 Conclusions and Future Remarks

In current study, three different Damage indicator methods have been proposed to achieve the Level 1 Structure health monitoring task. All the proposed methods are implementing through multi-layer perceptron and Convolutional Neural Network to perform three-class damage detection on datasets calculated from vibration data collected with real-world sensor monitoring system placed on a numerical beam. Specifically, the first method (A) adopts input statistical features only calculated on raw time series data, whereas the second method (B) attempts to enhance the classification performance considering the most informative subspace-based damage indicator. These damage sensitive features present the advantage of not requiring a prior operational modal analysis, but they directly work on residues of raw data comparing a reference state (undamaged) and a current (possible damaged) state. On the other hand, the most informative damage indicator calculations rely on the user's experience and capacity to identify the best optimal parameters (time shift and active and null subspace dimensions), which represent the trade-off between computational effort and the damage detectability resolution. Therefore, method (C) represents an innovative solution to the arbitrary definition of the parameters that affect the subspace-based damage indicator calculations. A numerical beam model has been built to simulate a realistic monitoring system placed on the structure to benchmark the three damaged levels. The sensors' locations are related to the

position of the mesh nodes of the FE model, which has been achieved through the OpenSeesPy module. In addition to validating feature selection and building a deep learning model, we further investigate the effects of the acquisition SNR levels, which affect actual sensors. The effects of different signal-to-noise ratio levels on the classification performance are evaluated. Both deep learning models and random subspace parameters are based on data, the most common parameters for structural health detection come from vibration. Therefore, studying the performance brought by accelerometers is the key to the data preparation in this topic. Through the comparison and research of various common sensors, MEMS sensors are the most frequently used in the future trend - that is, in wireless sensing systems. We set the research target on MEMS-accelerometers. Determine the most common accelerometer ideal working signal-to-noise ratio level and set three reasonable signal-to-noise ratios, 20dB, 40dB, 60dB. Gaussian white noise is the theory widely used in the deep learning field, is added to the signal collected from OpenSeesPy through a python program, aims to simulate signals collected by real Accelerometers from the real world, normal working condition, and environment. An innovative extended empirical sensitivity analysis has been conducted to explore the capabilities of some subspace-based damaged indicators applied in the current study, mainly referring to Yan's et al. damage indicator. Finally, ANN models and CNN models has been trained to perform the 3-class classification problem to correctly identify the damaged situation starting from the vibrational data monitored both in a reference undamaged state and a possible damaged. Three possible class have been set: undamaged, low level of damage (cross section reduction of damaged elements of 25%) and high level of damage (cross-section reduction of 50%). To increase the generalization of the deep learning models, 5000 simulations have been considered with a random choice of the number of possible damaged elements and damaged levels. The MLP performances appear good for all three current proposed methods in terms of overall accuracy (consistently higher than 90%), while the CNN model has accuracy lower (higher than 80%) in case (A) and (B). For MLP models, we have almost same shape for every case , and for CNN models, 3 different shapes have been designed to fit different situation. Those results are obtained from different signal-to-noise ratio levels, so it has good generalization performance.

In conclusion, the trained models exploit artificial intelligence and machine learning to provide a reliable system on real working condition even have rather good or bad noise situation that recognizes hidden patterns inside multivariate raw acceleration time-series because it considers statistical and/ or damage-sensitive features coming from all the sensors' acquisitions simultaneously. The implemented system could be applied for damage detection, such as a warning system when the damage occurs. Thus, the CNN and ANN models could work as first-level anomaly detection to distinguish from the undamaged situation and a possible damaged one by only recalling the trained model, which is, therefore, quite fast in usage. Indeed, the greatest computational effort is related to the training phase, not the prediction recall phase. Furthermore, not only at the ideal signal-to-noise ratio level, we can load the best model trained before at any reasonable signal-to-noise ratio level, and we can also get ideal results. This is due to the robustness of the best machine learning models to noise.

In the future study, the team and I are trying to complicate the current beam model, that is, to build a close-to-real truss bridge model and collect data to verify the effectiveness of the method provided in this thesis

in some complex structures. Then, I can research on the method of optimization and parameter adjustment in the machine learning field, to improve the accuracy of the CNN models in more complex environments and structures; finally, the three cases currently used are already existing in theory, so according to the characteristics of deep learning technology , we should research on whether it can directly learn some parameters that have not discovered so far from the original data, so as to be fit the concepts of ‘artificial intelligence’ and ‘deep learning’.

Reference

- [1] H. Ren, X. Chen, and Y. Chen, Eds., “References,” in *Reliability Based Aircraft Maintenance Optimization and Applications*, Academic Press, 2017, pp. 223–227. doi: <https://doi.org/10.1016/B978-0-12-812668-4.00012-5>.
- [2] “SAE ARP 6461-2013 (SAE ARP6461-2013) - Guidelines for Implementation of Structural Health Monitoring on Fixed Wing Aircraft.” <https://webstore.ansi.org/Standards/SAE/SAEARP64612013ARP6461> (accessed May 24, 2022).
- [3] A. Güemes, A. Fernandez-Lopez, A. R. Pozo, and J. Sierra-Pérez, “Structural Health Monitoring for Advanced Composite Structures: A Review,” *Journal of Composites Science*, vol. 4, no. 1, Art. no. 1, Mar. 2020, doi: 10.3390/jcs4010013.
- [4] M. Flah, “Classification, Localization, and Quantification of Structural Damage in Concrete Structures using Convolutional Neural Networks,” 2021. doi: 10.13140/RG.2.2.10317.36320.
- [5] G. Bernagozzi, C. E. Ventura, S. Allahdadian, Y. Kaya, L. Landi, and P. P. Diotallevi, “Output-only damage diagnosis for plan-symmetric buildings with asymmetric damage using modal flexibility-based deflections,” *Engineering Structures*, vol. 207, p. 110015, Mar. 2020, doi: 10.1016/j.engstruct.2019.110015.
- [6] “Vibrational Based Inspection of Civil Engineering Structures — Aalborg University’s Research Portal.” <https://vbn.aau.dk/en/publications/vibrational-based-inspection-of-civil-engineering-structures> (accessed May 24, 2022).
- [7] A. Mendler, M. Döhler, and C. E. Ventura, “Sensor placement with optimal damage detectability for statistical damage detection,” *Mechanical Systems and Signal Processing*, vol. 170, p. 108767, May 2022, doi: 10.1016/j.ymsp.2021.108767.
- [8] Z. He, W. Li, H. Salehi, H. Zhang, H. Zhou, and P. Jiao, “Integrated structural health monitoring in bridge engineering,” *Automation in Construction*, vol. 136, p. 104168, Apr. 2022, doi: 10.1016/j.autcon.2022.104168.
- [9] X. Zhou, C.-W. Kim, F.-L. Zhang, and K.-C. Chang, “Vibration-based Bayesian model updating of an actual steel truss bridge subjected to incremental damage,” *Engineering Structures*, vol. 260, p. 114226, Jun. 2022, doi: 10.1016/j.engstruct.2022.114226.
- [10] Z. Liu, S. Li, A. Guo, and H. Li, “Comprehensive functional resilience assessment methodology for bridge networks using data-driven fragility models,” *Soil Dynamics and Earthquake Engineering*, vol. 159, p. 107326, Aug. 2022, doi: 10.1016/j.soildyn.2022.107326.
- [11] J. Wang, C. Xu, J. Zhang, and R. Zhong, “Big data analytics for intelligent manufacturing systems: A review,” *Journal of Manufacturing Systems*, vol. 62, pp. 738–752, Jan. 2022, doi: 10.1016/j.jmsy.2021.03.005.

- [12] M. Kalyana Chakravarthy and S. Gowri, “Interfacing advanced NoSQL database with Python for internet of things and Big data Analytics,” *Materials Today: Proceedings*, Mar. 2021, doi: 10.1016/j.matpr.2021.01.724.
- [13] J. McCarthy, “WHAT IS ARTIFICIAL INTELLIGENCE?,” p. 14.
- [14] I. Lauriola, A. Lavelli, and F. Aioli, “An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools,” *Neurocomputing*, vol. 470, pp. 443–456, Jan. 2022, doi: 10.1016/j.neucom.2021.05.103.
- [15] R. Wang, T. F. Edgar, M. Baldea, M. Nixon, W. Wojsznis, and R. Dunia, “A geometric method for batch data visualization, process monitoring and fault detection,” *Journal of Process Control*, vol. 67, pp. 197–205, Jul. 2018, doi: 10.1016/j.jprocont.2017.05.011.
- [16] J. Yang *et al.*, “Intelligent bridge management via big data knowledge engineering,” *Automation in Construction*, vol. 135, p. 104118, Mar. 2022, doi: 10.1016/j.autcon.2021.104118.
- [17] S. Taheri, “A review on five key sensors for monitoring of concrete structures,” *Construction and Building Materials*, vol. 204, pp. 492–509, Apr. 2019, doi: 10.1016/j.conbuildmat.2019.01.172.
- [18] A. Barrias, J. R. Casas, and S. Villalba, “A Review of Distributed Optical Fiber Sensors for Civil Engineering Applications,” *Sensors*, vol. 16, no. 5, Art. no. 5, May 2016, doi: 10.3390/s16050748.
- [19] “(PDF) Polymer Fiber Optic Sensors – A Mini Review of their Synthesis and Applications.” https://www.researchgate.net/publication/301281352_Polymer_Fiber_Optic_Sensors_-_A_Mini_Review_of_their_Synthesis_and_Applications (accessed May 25, 2022).
- [20] “Smart Sensing Technologies for Structural Health Monitoring of Civil Engineering Structures.” <https://www.hindawi.com/journals/ace/2010/724962/> (accessed May 25, 2022).
- [21] J. López-Higuera, L. R. Cobo, A. Q. Incera, and A. Cobo, “Fiber Optic Sensors in Structural Health Monitoring,” *Journal of Lightwave Technology*, 2011, doi: 10.1109/JLT.2011.2106479.
- [22] “Overview of Fiber Optic Sensor Technologies for Strain/Temperature Sensing Applications in Composite Materials - PMC.” <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4732132/> (accessed May 25, 2022).
- [23] “BS EN 50324-3:2002 | API Standards Store.” https://www.techstreet.com/api/standards/bs-en-50324-3-2002?product_id=1115474 (accessed May 25, 2022).
- [24] “On the repair of a cracked beam with a piezoelectric patch - CityU Scholars | A Research Hub of Excellence.” [https://scholars.cityu.edu.hk/en/publications/publication\(e3c15e5c-5c3f-4ebd-991c-cff498658ca9\).html](https://scholars.cityu.edu.hk/en/publications/publication(e3c15e5c-5c3f-4ebd-991c-cff498658ca9).html) (accessed May 25, 2022).
- [25] “A wireless piezoelectric sensor network for distributed structural health monitoring | Semantic Scholar.” <https://www.semanticscholar.org/paper/A-wireless-piezoelectric-sensor-network-for-health-Gao-Dai/40e93f06af7debf5ad3657452dda2a6f54b572d> (accessed May 25, 2022).

- [26] “Nyquist Theorem - an overview | ScienceDirect Topics.” <https://www.sciencedirect.com/topics/engineering/nyquist-theorem> (accessed Jul. 13, 2022).
- [27] J. Tian *et al.*, “Zero drift suppression for PdNi nano-film hydrogen sensor by vacuum annealing,” *International Journal of Hydrogen Energy*, vol. 45, no. 28, pp. 14594–14601, May 2020, doi: 10.1016/j.ijhydene.2020.03.185.
- [28] J. Kullaa, “Detection, identification, and quantification of sensor fault in a sensor network,” *Mechanical Systems and Signal Processing*, vol. 40, no. 1, pp. 208–221, Oct. 2013, doi: 10.1016/j.ymsp.2013.05.007.
- [29] L.-Y. Shao *et al.*, “Data-Driven Distributed Optical Vibration Sensors: A Review,” *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6224–6239, Jun. 2020, doi: 10.1109/JSEN.2019.2939486.
- [30] H. Y. Teh, A. W. Kempa-Liehr, and K. I.-K. Wang, “Sensor data quality: a systematic review,” *Journal of Big Data*, vol. 7, no. 1, p. 11, Feb. 2020, doi: 10.1186/s40537-020-0285-1.
- [31] “KB12VD-KS48C.pdf.” Accessed: May 25, 2022. [Online]. Available: <https://marmatek.com/wp-content/uploads/2021/06/KB12VD-KS48C.pdf>
- [32] “LUNA-Data-Sheet-Micron-os7500-1.pdf.” Accessed: May 25, 2022. [Online]. Available: <https://lunainc.com/sites/default/files/assets/files/resource-library/LUNA-Data-Sheet-Micron-os7500-1.pdf>
- [33] J. Zhu *et al.*, “Development Trends and Perspectives of Future Sensors and MEMS/NEMS,” *Micromachines*, vol. 11, no. 1, Art. no. 1, Jan. 2020, doi: 10.3390/mi11010007.
- [34] Z. Chen, X. Zhou, X. Wang, L. Dong, and Y. Qian, “Deployment of a Smart Structural Health Monitoring System for Long-Span Arch Bridges: A Review and a Case Study,” *Sensors*, vol. 17, p. 2151, Sep. 2017, doi: 10.3390/s17092151.
- [35] A.-M. Yan and J.-C. Golinval, “Null subspace-based damage detection of structures using vibration measurements,” *Mechanical Systems and Signal Processing*, vol. 20, no. 3, pp. 611–626, Apr. 2006, doi: 10.1016/j.ymsp.2005.04.010.
- [36] M. M. Rosso and A. Aloisio, “Enhancing the classification performance of multi-layer perceptron network for damage-detection using subspace-based metrics”.
- [37] “The OpenSeesPy Library — OpenSeesPy 3.4.0.1 documentation.” <https://openseespydoc.readthedocs.io/en/latest/> (accessed May 26, 2022).
- [38] E. B. Groth, T. G. R. Clarke, G. Schumacher da Silva, I. Iturrioz, and G. Lacidogna, “The Elastic Wave Propagation in Rectangular Waveguide Structure: Determination of Dispersion Curves and Their Application in Nondestructive Techniques,” *Applied Sciences*, vol. 10, no. 12, Art. no. 12, Jan. 2020, doi: 10.3390/app10124401.

- [39] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, “Applications of Wireless Sensor Networks: An Up-to-Date Survey,” *Applied System Innovation*, vol. 3, no. 1, Art. no. 1, Mar. 2020, doi: 10.3390/asi3010014.
- [40] L. Zhu, Y. Fu, R. Chow, B. F. Spencer, J. W. Park, and K. Mechitov, “Development of a High-Sensitivity Wireless Accelerometer for Structural Health Monitoring,” *Sensors*, vol. 18, no. 1, Art. no. 1, Jan. 2018, doi: 10.3390/s18010262.
- [41] D. Tang *et al.*, “Research on sampling rate selection of sensors in offshore platform shm based on vibration,” *Applied Ocean Research*, vol. 101, p. 102192, Aug. 2020, doi: 10.1016/j.apor.2020.102192.
- [42] M. R. Bachute and J. M. Subhedar, “Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms,” *Machine Learning with Applications*, vol. 6, p. 100164, Dec. 2021, doi: 10.1016/j.mlwa.2021.100164.
- [43] R. Bardenet and A. Hardy, “Time-frequency transforms of white noises and Gaussian analytic functions,” *Applied and Computational Harmonic Analysis*, vol. 50, pp. 73–104, Jan. 2021, doi: 10.1016/j.acha.2019.07.003.
- [44] L. Zhou, G. Sun, Y. Li, W. Li, and Z. Su, “Point cloud denoising review: from classical to deep learning-based approaches,” *Graphical Models*, vol. 121, p. 101140, May 2022, doi: 10.1016/j.gmod.2022.101140.
- [45] M. S. Reis, P. M. Saraiva, and B. R. Bakshi, “3.10 - Data Quality and Denoising: A Review☆,” in *Comprehensive Chemometrics (Second Edition)*, S. Brown, R. Tauler, and B. Walczak, Eds. Oxford: Elsevier, 2020, pp. 179–204. doi: 10.1016/B978-0-12-409547-2.14874-7.
- [46] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” *Information Fusion*, vol. 55, pp. 220–244, Mar. 2020, doi: 10.1016/j.inffus.2019.09.003.
- [47] Y. Yu, C. Wang, X. Gu, and J. Li, “A novel deep learning-based method for damage identification of smart building structures,” *Structural Health Monitoring*, vol. 18, no. 1, pp. 143–163, Jan. 2019, doi: 10.1177/1475921718804132.
- [48] “NumPy.” <https://numpy.org/> (accessed Jul. 13, 2022).
- [49] “OMG! What is OMA? Operational Modal Analysis.” <https://community.sw.siemens.com/s/article/OMG-What-is-OMA-Operating-Modal-Analysis> (accessed Jul. 13, 2022).
- [50] “Synergistic bridge modal analysis using frequency domain decomposition, observer Kalman filter identification, stochastic subspace identification, system realization using information matrix, and autoregressive exogenous model - ScienceDirect.” <https://www.sciencedirect.com.ezproxy.biblio.polito.it/science/article/pii/S0888327021002132> (accessed Jul. 13, 2022).

- [51] K. A. Kvåle, O. Øiseth, and A. Rønnquist, “Operational modal analysis of an end-supported pontoon bridge,” *Engineering Structures*, vol. 148, pp. 410–423, Oct. 2017, doi: 10.1016/j.engstruct.2017.06.069.
- [52] M. Basseville, M. Abdelghani, and A. Benveniste, “Subspace-based fault detection algorithms for vibration monitoring,” *Automatica*, vol. 36, no. 1, pp. 101–109, Jan. 2000, doi: 10.1016/S0005-1098(99)00093-X.
- [53] “Understanding standards | IEC.” <https://www.iec.ch/understanding-standards> (accessed Jul. 18, 2022).
- [54] A. Rahaman, C. H. Park, and B. Kim, “Design and characterization of a MEMS piezoelectric acoustic sensor with the enhanced signal-to-noise ratio,” *Sensors and Actuators A: Physical*, vol. 311, p. 112087, Aug. 2020, doi: 10.1016/j.sna.2020.112087.
- [55] “SciELO - Brazil - An SHM approach using machine learning and statistical indicators extracted from raw dynamic measurements An SHM approach using machine learning and statistical indicators extracted from raw dynamic measurements.” <https://www.scielo.br/j/lajss/a/LxWRCnsCqTv73ZJVwtRznKh/?lang=en> (accessed Jul. 14, 2022).
- [56] “Operational Modal Analysis of Civil Engineering Structures | SpringerLink.” <https://link.springer.com/book/10.1007/978-1-4939-0767-0> (accessed Jul. 14, 2022).
- [57] “Metrics - Keras .” <https://keras.io/zh/metrics/> (accessed Jul. 14, 2022).
- [58] K. Team, “Keras documentation: Developer guides.” <https://keras.io/guides/> (accessed Jul. 14, 2022).