DOCTORAL DISSERTATION

M.Sc. Jefferson da Silva Coelho

STRUCTURAL HEALTH MONITORING OF WIND TURBINES COMPONENTS USING MACHINE LEARNING

MONITOROWANIE STANU TECHNICZNEGO TURBIN WIATROWYCH Z WYKORZYSTANIEM UCZENIA MASZYNOWEGO

FIELD: ENGINEERING AND TECHNICAL SCIENCE
DISCIPLINE PBS: CIVIL ENGINEERING, GEODESY AND TRANSPORT
DISCIPLINE UnB: MECHANICAL SCIENCE

SCIENTIFIC SUPERVISOR
dr.hab.inż.prof. Maciej Dutkiewicz
Bydgoszcz University of Science and Technology

SCIENTIFIC SUPERVISOR
dr.hab.inż.prof. Marcela Rodrigues Machado
University of Brasília
Bydgoszcz University of Science and Technology

February 2025

STRUCTURAL HEALTH MONITORING OF WIND TURBINES COMPONENTS USING MACHINE LEARNING

MONITOROWANIE STANU TECHNICZNEGO TURBIN WIATROWYCH Z WYKORZYSTANIEM UCZENIA MASZYNOWEGO

MONITORAMENTO DA INTEGRIDADE ESTRUTURAL DE COMPONENTES DE TURBINA EÓLICA UTILIZANDO APRENDIZADO DE MÁQUINA

M.Sc. Jefferson da Silva Coelho

Dissertation presented in fulfilment of the requirements for the degree of Doctor of Engineering and Technical Science (PhD) from Bydgoszcz University of Science and Technology and Doctor of Mechanical Science from the University of Brasilia.

February 2025

Funding Source

This research is part of the project No. 2022/45/P/ST8/02123 co-funded by the National Science Centre and the European Union Framework Programme for Research and Innovation Horizon 2020 under the Marie Skłodowska-Curie grant agreement no. 945339, CNPq 444595/2024-4, FAPDF.00193-00002143/2023-02, FAPEAM - Fundação de Ampora à pesquisa do estado do Amazonas.

Acknowledgements

I wish to express my sincere thanks to my supervisor, Prof. Dr. Marcela Machado, for her motivation, insightful discussions, knowledge, patience and valuable guidance throughout this project. I also wish to express my gratitude to my supervisor Prof. Dr. hab. Maciej Dutkiewicz for his support, comments, and discussions, as well as for his commitment to making my stay in Poland possible.

I would like to my parents and siblings for their invaluable advice, unwavering support over the years, and for helping me achieve my goals. My special thanks to my wife for her constant support, understanding, and patience at every stage of this journey.

I also thank to my friends, labmates, and professors at UnB's Department of Mechanical Sciences for their guidance, assistance, and support.

I would like also to thank all the friends I made in Poland, for making me feel at home despite being far from Brazil.

I thank all the people who, directly or indirectly, contributed to the completion of this thesis. Last but not least, my sincere thanks to all the academic and non-academic staff at Politechnika Bydgoska who helped me feel welcome in a foreign country.

List of Abbreviations

ACC	Accelerometer
AI	Aerodynamic imbalance on one blade
AIGAC	Average Integration Global Amplitude Criteria
AIGSC	Average Integration Global Shape Criteria
ANN	Artificial Neural Networks
cDI	Combined Damage index
CFM	Corresponding Flow Matching
CNN	Convolutional Neural Network
CTGAN	Conditional Tabular Generative Adversarial Networks
DI	Damage Index
DL	Deep Learning
DT	Decision Tree
DTR	Decision Tree Regression
FEM	Finite Element Model
FC	Flexible coupling of the linear drive of the collective pitch system
FRF	Frequency Response Functions
FRAC	Frequency Response Assurance Criteria
FAAC	Frequency Amplitude Assurance Criteria
GAC	Global Amplitude Criteria
GAN	Generative Adversarial Networks

GBR	Gradient-boosting regression
GMM	Gaussian mixture model
GP	Gaussian Process
GSC	Global Shape Criteria
GWEC	Global Wind Energy Council
GEN	Generator
KNN	K-Nearest Neighbors
KNR	K-Nearest Neighbors Regression
LASSO	Least Absolute Shrinkage and Selection Operator
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
M-DI	Monnier Damage Index
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
MSB	Main shaft bearing
NB	Naïve Bayes
NMF	Nacelle main frame
NO	Normal Operation
PCA	Principal Component Analysis
RBF	Radial Basis Function
R-DI	Banerjee Damage Index

RF	Random Forest
RMS	Root Mean Square
RMSE	Root Mean Squared Error
RI	Rotor icing event
SHM	Structural Heath Monitoring
SPR	Statistical Pattern Recognition
SVM	Support Vector Machine
SVR	Support Vector Regression
TGAN	Tabular Generative Adversarial Networks
UQ	Uncertainty quantification
XGBoost	eXtreme Gradient Boosting

List of Symbols

ω	Frequency spectrum
$H_{ij}^{dam}, H_{ij}^{und}$	Vector on (j) for the damaged and undamaged excited on (i)
$x(t)$	Time-domain signal
τ	Signal rate
$\overline{x}$	Mean value
Δ_f	Difference between each element in the feature vector
μ	Mean of logarithmic values
σ	Standard deviation of logarithmic values
ν	Variance
u_i',d_i'	Generated artificial data
u_i,d_i	Original data
n_c,n_d	Continuous and Discrete variables
k	Number of cluster
k_n	Number of nearest neighbour
k_f	Number of subsets (folds).
ξ	Slack variables
ϵ	Error “insensitive” area
θ	Slope of the line
α	Intercept
x	Independent variable (input)

y	Dependent variable (output)
$\hat{y}$	Average of target values.
λ	Regularization parameter that controls the penalty
β_0,β_j	coefficients to be estimated
δ_i	Expansion coefficient

Contents

1 Introduction

Modern engineering structures and systems face growing challenges due to climate change, ageing infrastructure, and evolving monitoring demands. These factors drive traditional methods to their limits, highlighting the need for more resilient and adaptive solutions. Hence, engineering systems must integrate advanced materials, real-time monitoring, and intelligent design strategies to address these demands. The rapid advancement of digital technologies is transforming structural health monitoring (SHM). Innovations like the Internet of Things, virtual sensors, and digital twin technology enable real-time assessment of structural conditions, creating on-demand tools to model physical assets and predict potential failures. This digital evolution is particularly critical for renewable energy systems, which are expanding and operating under constant environmental exposure. Ensuring their reliability and efficiency requires high-performance materials, rapid diagnostic methods, and advanced data processing tools. As a result, the demand for smarter, more predictive SHM solutions continues to evolve and transform the future of engineering infrastructure.

Wind energy has attracted notable attention over the past decade due to its potential as a clean, renewable energy source and global support for its development, especially in the intensified use of renewable energy [1]. The sector's latest projections evidence the growing importance of wind energy. According to Global Wind Report 2024, [2], published by the Global Wind Energy Council (GWEC), global installed wind power capacity will exceed 791 GW by the end of 2028 (see Figure 1.1), with an average of 158 GW of new installations annually until then. This rapid expansion is concentrated in the key markets of China, the EU, the US, India, and Brazil [3].

Due to the importance of obtaining energy from wind in the entire area of renewable energy sources, proper planning of wind farms is essential. In the world and individual countries, the principles of design, location, construction, and use of wind turbines are regulated in detail. For example, in Poland, the key legal act concerning the construction and operation of wind

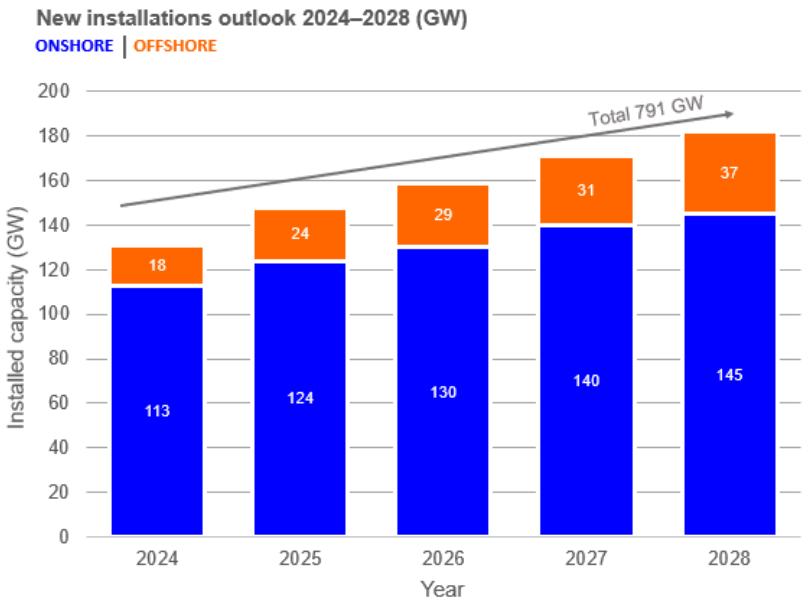


Figure 1.1: Global installed wind capacity [2].

turbines is a document on investments in wind farms [4]. This document introduces the principle of the minimum distance of wind farms from residential buildings and forms of nature conservation. Work on regulations to liberalise the principles of the location of wind turbines is currently underway to amend this document. The proposed changes include, among others, reducing the minimum distance from buildings to 500 meters. The construction law [5] also defines the principles of mounting wind turbines directly on a building, and it is planned to introduce further simplifying solutions. The location of wind turbines must be consistent with the local spatial development plan [6]. In the absence of such a plan, it is necessary to obtain a decision on development conditions. The construction of a wind turbine on a separate foundation is treated as a separate construction object, which requires obtaining appropriate building and use permits.

Wind turbines are energy sources that convert the kinetic energy of the wind (wind energy) into electrical energy from a renewable source [7]. They are generally categorised by their axis orientation, horizontal or vertical,

and by their installation location onshore, turbines on land or offshore, and turbines at sea. Turbines are composed of mechanical and electrical components interlinked [8]. Their main subsystems comprising various components are the nacelle, rotor, tower, blades, and foundation, as shown in Figure 1.2. They are tower structures of considerable height and slenderness, subject to complex aerodynamic loads resulting from the action of the wind. The loads acting on such structures also change in icing and temperature changes. Those turbines are complex structural and building systems that require continuous monitoring to ensure operational efficiency, minimise costs, and prevent failures. With the rapid advance of the wind energy sector, pro-

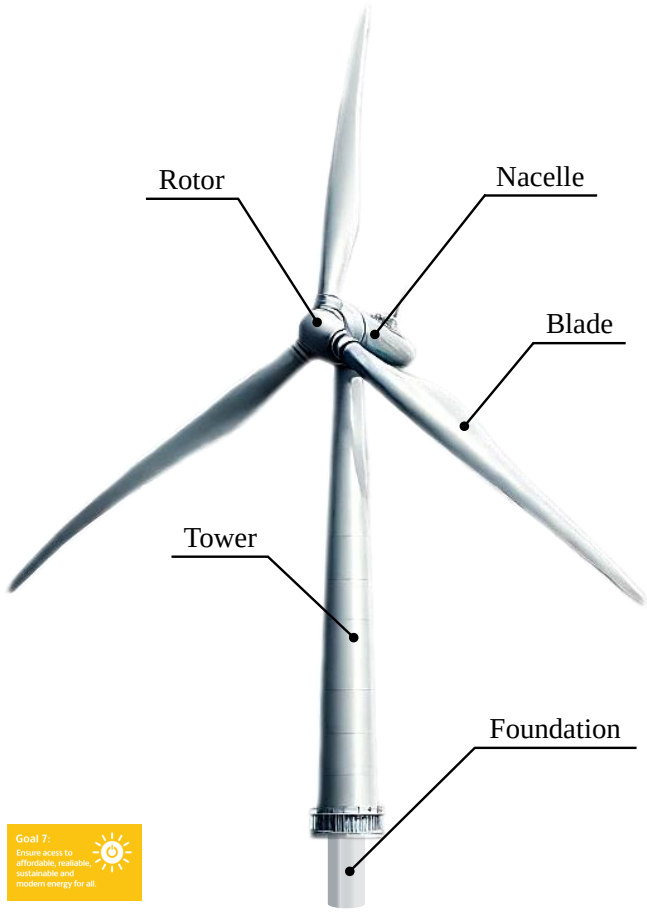


Figure 1.2: Wind turbine mains sub-parts (Source: own study).

duction has increased yearly. This progress drives the construction of ever larger, more powerful, and technologically advanced wind turbines [9]. However, as these turbines evolve in scale and complexity, they face significant challenges. On the one hand, an increase in turbine size and capacity allows for greater efficiency in energy generation. On the other hand, these advances also bring structural and operational implications [10]. Turbines are subject to huge mechanical loads and extreme environmental conditions during operation, which has increased the frequency of failures and maintenance costs [11]. These failures can reduce the reliability of wind turbines and increase unexpected shutdowns and maintenance requirements [9]. Studies indicate that the operating and maintenance costs for onshore and offshore wind turbines represent 10-15% and 20-35% of total lifetime costs in wind turbine systems [12]. As a result, the wind energy industries face a high demand for improvements in the reliability, safety, availability, and productivity of wind turbine systems [13].

Like any other electromechanical system, wind turbines are subject to several unforeseen and serious failures that can result in fatal disasters [18]. With the rapid expansion of wind energy worldwide, there has been a significant increase in the frequency of accidents at wind farms, with several reported cases. In recent years, the wind energy sector has experienced some catastrophic failures, such as the one in the Netherlands in 2013, where a turbine fired up during maintenance, causing a loss of employee life (Figure 1.3a). In 2021, a set of nacelles and blades fell 100m from a Delta 1 wind farm turbine tower in Piaui, Brazil (Figure 1.3b). An accident occurred in Germany in December 2022 when a wind turbine caught fire (Figure 1.3c). The same year, with strong winds, a tornado in Texas, USA, resulted in blade failure in at least three wind turbines (Figure 1.3d).

Failure mechanisms in wind turbines can be classified into three main categories [19]: mechanical, electrical and environmental causes. Mechanical failures generally affect components such as the rotor blades, connections, gearbox, bearings and main shaft, risking the turbine's performance [20]. In the electrical system, which includes generators, converters and control

(a) Netherlands

(b) Brazil

(c) Germany

(d) USA

Figure 1.3: Wind turbine accidents [14, 15, 16, 17].

systems, failures can result from insulation degradation, thermal voltages and electrical transients, impacting the efficiency and safety of the operation [21]. Environmental factors such as extreme temperatures, atmospheric discharges, humidity and corrosion pose significant challenges to turbine reliability and service life [22]. Typical failures of wind turbine components and their causes are described in Table 1.1.

Table 1.1: Typical failures in wind turbines components and causes [12, 19, 23, 24].

Category	**Components**	**Causes**
Mechanical	Blades and rotors	Corrosion of blades and hub; crack; reduced stiffness; increased surface roughness; deformation of the blades; errors of pitch angle; and imbalance of rotors, etc.
	Gearbox	Imbalance and misalignment of shaft; damage of shaft, bearing and gear; broken shaft; high oil temperature; leaking oil; and poor lubrication, etc.
	Bearings	Overheating; and premature wear caused by unpredictable stress, etc.
	Main shaft	Misalignment; crack; corrosion; and coupling failure, etc.
	Hydraulic Faults	Sliding valve blockage; oil leakage, etc.
	Mechanical braking system	Hydraulic failures; and wind speed exceeding the limit, etc.
	Connections system	Flexible coupling; fixed connection (bolts).
	Tower	Poor quality control during the manufacturing process; improper installation and loading; harsh environment, etc.
Electrical	Generator	Excessive vibrations of generator; overheating of generator and bearing; abnormal noises; insulation damage.
	Electrical systems/ devices	Broken buried metal lines; corrosion or crack of traces; board delamination; component misalignment; electrical leaks; cold-solder joints, etc.
	Sensors	Malfunction or physical failure of a sensor; malfunction of hardware or the communication link; and error of data processing or communication software.
Environmental	Lightning strikes	Cause damage to turbine components, e.g. rotor blades and electrical systems.
	Temperatures variations	Low/high temperatures can cause materials and components to expand or contract, accelerating fatigue.
	Humidity	Accelerate corrosion-fatigue; mold; condensation; freezing; and electrical failures.
	Corrosion	Due to the harsh and highly corrosive environmental conditions in which they operate, especially in coastal or offshore installations.

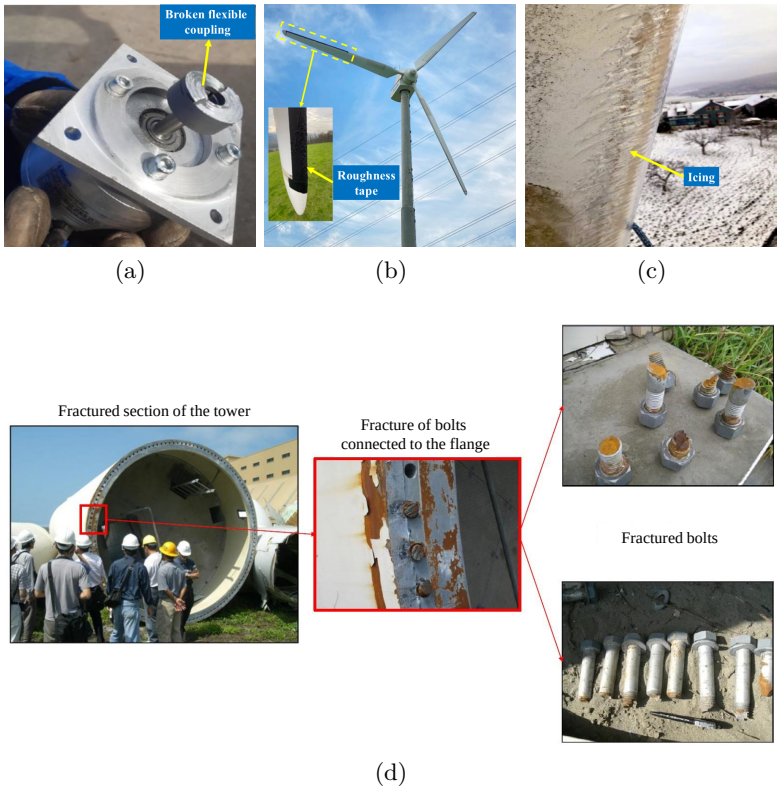


Figure 1.4: Wind turbine mechanical failure events: (a) failure of the pitch drive system; (b) aerodynamic imbalance due to roughness effects; (c) Icing on the blade [25]; (d) Flange and bolts of a tower failing on-site [26] (© Creative Commons Attribution 4.0 International).

Common mechanical failures in wind turbines are related to icing, aerodynamic imbalance, flexure coupling drive failure, and connections associated with bolts, as illustrated in Figure 1.4. Mechanical failure in the pitch drive system, related to flexible coupling responsible for adjusting the blade angles, is shown in Fig. 1.4a. This failure type results in a loss of pitch control. Figure 1.4b shows the aerodynamic imbalance fault, which imposes changes in the blade surface, modifies the airflow and reduces the turbine's aerodynamic efficiency. In the example case, the imbalance is simulated by applying roughness tape to the blade. Icing is a common event due to environmental conditions directly affecting the turbine's mechanical components, such as the blades. Figure 1.4c shows ice formation on the blade,

which compromises aerodynamic performance and causes an operational imbalance, especially in cold weather conditions.

Wind turbines are made of several components, and it is common to use bolted joints to connect those components. Large bolts are used at critical points, such as the tower flange, the blade root, and certain nacelle connections. However, bolts can become loose during normal operation due to prolonged operation and exposure to vibration [27, 28]. Excessive wear of the joints can result in serious failures, including tower collapse, blade separation or even detachment of the rotor from the nacelle [29]. Failures of wind turbine blade root bolts were reported at a wind farm in Inner Mongolia, China. After three years of performance, the turbine blade root bolts suddenly broke, and the blade fell [30]. Another case is a fracture of the screws connecting the blade to the hub of the wind turbine rotor [31]. This failure occurred soon after the start of the wind turbine. The disaster was preceded by strong winds for a few days, which caused the tower's collapse. In 2008, a typhoon hit Taiwan and toppled a wind turbine tower off the coast of the port of Taichung. A study by Chou and Tu [26] identified that strong winds, insufficient strength of bolts and poor-quality control of bolts during construction were the likely causes of the tower's collapse (Figure 1.4d).

Monitoring systems, as wind turbines, have become more complex. It must guarantee greater reliability, efficiency and the ability to identify faults. However, detecting these problems is challenging as turbines operate in extreme and variable environments. Sensors can't always cope well with these challenges, which makes it essential to use more advanced and adaptable methods to analyse large volumes of often abnormal data [32]. In this framework, structural health monitoring of the systems must identify specific faults that could risk the performance and structural integrity of turbines, as this is a developing technology that combines sensors and intelligent computer algorithms capable of carrying out structural monitoring in real-time or in defined time intervals [33]. Since the integrity and safety of the structures depend on monitoring the occurrence, formation, and prop-

agation of structural damage, the need to use the SHM technique to detect early damage in wind turbines is of great importance.

Fault detection methods can be categorised into two approaches [34, 35], the model-based methods and the data-based methods. Among these, data-based methods have gained prominence because they do not require precise physical models or advanced knowledge of signal processing [36, 37, 38]. These methods use statistical and signal processing techniques to identify patterns and create turbine fault indicators [39]. With advances in artificial intelligence, especially machine learning, it has become possible to improve these techniques further, enabling the automatic extraction of relevant information from large volumes of data. The data-based approach is used in this research, and advances in state-of-the-art technology are proposed.

The manipulation of the data acquired from the monitoring systems has advanced and become possible with artificial intelligence tools. Machine learning algorithms have made significant advances in structural monitoring, achieving higher levels of precision than traditional methods. These approaches facilitate uncertainty modelling and statistical pattern recognition analysis, supporting decision-making and handling a wider data fusion. However, developing models capable of accurately capturing the various non-linearities and variabilities of the system still represents an area of research with great potential to be explored. Current research into fault detection in turbine components employing machine learning (ML) techniques has advanced over the decade, but it is still ongoing research. Thus, a significant gap exists in applying ML-based condition assessment methods incorporating data augmentation, uncertainty quantification, and raw vibration spectra acquired from structures. An issue in the data-driven approach SHM is the reduced value of data, missing data, and data sensitivity towards the damage. Researchers have addressed this issue by mixing experimental data with simulated or synthetic data. However, these sets often do not accurately represent real-world conditions. Another way to expand the data set is the use of combined features obtained from the signal. There-

fore, it typically will repeat the information among the feature extractor techniques employed in the process.

Aside from the reduced dataset, which can be a factor and potential problem for the machine learning algorithms to perform fault detection accurately, the data size, processing and computational effort can be a limited factor. Applied signal differentiation education, or feature extraction, has been used. Feature extraction is one of the critical steps in the SHM process that will influence the whole monitoring process. Various methods have been used in this process and must be carefully selected depending on the type of signal and information acquired from the structure. Indeed, during the monitoring process, the knowledge of the monitoring systems and our structures associated with the information under search is the main goal previously defined before using any SHM technique.

1.1 Motivation

Mechanical and structural failures in systems, ranging from aerodynamic imbalance and icing to bolted joint failures, can severely impact performance, safety, and operational costs. Traditional monitoring techniques have encountered issues in detecting early-stage faults due to the complexity of turbine dynamics and the harsh environmental conditions in which they operate. Therefore, the demand for advanced, data-driven monitoring solutions has intensified. In this context, SHM integrated with ML offers a transformative approach, enabling real-time fault detection, predictive maintenance, inspection and structure evaluation, and enhanced operational efficiency. However, current data-driven SHM methods face challenges such as limited datasets, high computational costs, and difficulty accurately capturing nonlinear system behaviours. Addressing these limitations requires innovative methodologies that employ machine learning, data augmentation, and advanced signal processing techniques to improve fault detection accuracy and robustness. The study motivation is to bridge these gaps by advancing state-of-the-art in SHM with ML techniques and enhancing the resilience and efficiency of systems and structures.

1.2 Objectives

This research objective[1] is to develop a methodology for monitoring systems and structures from their dynamic response using SHM and ML techniques. The proposed methodology can detect statistical pattern recognition and quantify damage associated with the uncertainties related to the process. Henceforth, the specific objectives are:

- Implement an architecture to pre-process the signal, feature extraction and selection using spectral and statistic methods;
- Develop a virtual sensor integrated into the SHM-ML for data augmentation;
- Design a monitoring route involving a machine learning algorithm for pattern recognition and detection of damage;
- Design a monitoring route involving a machine learning algorithm for damage quantification associated with the uncertainty in the process;
- Evaluate the performance of the data-driven SHM-ML methodology with experimental data tested in laboratory systems and *in-situ* operational wind turbine.

The proposed methodology's efficacy and specific objectives are evaluated through three case studies on mechanical faults in wind turbine components. In the first case, the SHM-ML framework is applied to detect, classify, and recognise patterns of torque loosening in bolted joint structures. To address the challenges posed by the variability of these connections, the proposed approach integrates both supervised and unsupervised machine learning algorithms. The methodology relies on experimental data collected under different test conditions, using raw frequency spectral signals to estimate a damage index (DI) that serves as a feature extractor. This DI is then processed by machine learning algorithms, enabling the approach to

[1]This research aligns with the 7th Sustainable Development Goal established by the World Health Organization, which focuses on ensuring access to affordable, reliable, sustainable, and modern energy for all.

handle measurement variability and uncertainties without requiring prior modal analyses.

The second case study builds upon the same experimental setup but enhances the quantification of bolt torque loss by incorporating data augmentation into the SHM-ML methodology. Here, frequency spectra are also utilised. While many studies rely on simulated or synthetic data, these datasets often fail to capture real-world conditions accurately. Therefore, a key objective of this research is to develop an improved condition assessment model based on machine learning to estimate bolt torque loss directly from raw vibration signal spectra. The proposed approach employs data augmentation and fusion strategies to enrich the dataset, eliminating the need for numerical models and relying exclusively on experimental data.

The third application involves monitoring a real wind turbine. This case study employs an unsupervised clustering algorithm to classify operational states and identify structural patterns without requiring predefined labels. A new feature extraction technique was introduced to enhance classification accuracy, incorporating a relative damage index variation metric. The SHM-ML methodology has demonstrated effectiveness across various monitoring scenarios, from simple to complex systems, emerging as a viable technical solution for integration into real wind turbines. With the increasing demand for remote structural monitoring solutions, particularly for fault detection in critical components, the proposed SHM-ML framework facilitates proactive maintenance, minimises downtime, and enhances operational efficiency. By advancing structural health monitoring and vibration control strategies for wind turbines, this study contributes to improving the efficiency and reliability of renewable energy systems, thereby supporting the transition to cleaner and more sustainable energy sources.

1.3 Organisation of the dissertation

The thesis is organised by presenting an introduction to address the scope first, followed by a literature review. Then, the subsequent chapters describe the methodology and results obtained in case studies with the

application of ML techniques for SHM in wind turbine components and the conclusions. More specifically, this thesis is structured as follows:

- **Chapter 1** presents a general introduction describing the motivation, objectives, and organisation. The importance of SHM and the role of ML in this context are discussed.
- **Chapter 2** presents a literature review covering the fundamentals of SHM, methods, and the application of ML in damage detection. In addition, the chapter discusses different ML approaches and their relevance to SHM in wind turbine components.
- **Chapter 3** describes the methodology used in this research, including data processing, feature extraction, ML strategies and uncertainty quantification. The chapter also details frequency and time domain analysis with data augmentation techniques to increase the robustness of feature extraction.
- **Chapter 4** demonstrates case studies in experimental applications of ML for SHM. It includes the classification of faults in wind turbine components, detecting loosening torque in bolted structures, and quantifying damage using ML-based virtual sensors. The results and discussions highlight the effectiveness of the proposed methods.
- **Chapter 5** concludes the dissertation by summarising the main findings, discussing the study's contributions, and suggesting future research directions.

2 Literature Review

This research proposes monitoring wind turbines and their components using the proposed damage assessment with a machine learning process. The present chapter gathers a literature review of past research and development related to this doctoral dissertation. It starts with an overview of machine learning techniques for structural health monitoring, SHM in turbine component monitoring and bolted structures. The chapter ends with a general analysis and the contribution of this research to the state-of-art.

2.1 Structural health monitoring

Structural health monitoring provides significant advantages across various sectors, providing tools for scheduling maintenance and promoting the safety and reliability of structures and systems. Hence, continuously assessing structural conditions enables early detection of potential damage or anomalies, allowing for timely maintenance and repairs [40]. Furthermore, SHM is a proactive action that minimises the risk of catastrophic failures and extends the lifespan of these structures [41]. Additionally, SHM enhances maintenance efficiency by identifying specific areas that require attention, ultimately reducing operational and maintenance costs.

SHM techniques are commonly defined as a strategy for detecting structural damage through continuous or periodic monitoring [42]. This process involves data acquisition from sensors embedded in the structure or contactless, extracting damage-related features, and analysing them to assess potential failures. This approach can help us estimate the monitored system's remaining lifespan [43]. Enabling the transition from offline damage identification to near real-time online assessment significantly improves the efficiency and speed of damage evaluation. The tasks of SHM technology are categorised into five levels, as defined by [44]:

- **Level 1- Detection:** provide information on the presence of damage.

- **Level 2- Localisation:** provide information on the location of the damage.
- **Level 3- Classification:** provide information on the type of damage.
- **Level 4- Assessment:** provide information on the extent of the damage.
- **Level 5- Prognosis:** provide information on the residual life and safety of the structure.

Advancing to the next level in the hierarchy depends on the successful completion of the previous ones. It is well known that higher levels of detail are increasingly challenging to achieve. For instance, prognosis is particularly complex because it requires a deep understanding of the physics of damage [45]. Therefore, determining the necessary level of identification is a critical decision when developing an SHM strategy. The damage in the monitoring process is defined as any modification or deterioration in a structure's integrity, performance, or behaviour that compromises its safety, functionality, or longevity. This encompasses a range of issues, including cracks, corrosion, deformations, fatigue, and material degradation, all of which can weaken structural capacity or threaten stability. Early detection and monitoring of such damage enable engineers to evaluate its severity and implement preventive measures to avoid catastrophic failures.

Structural damage detection associated with SHM techniques is broadly categorised into global and local methods. Vibration-based methods [46] are typically global, while local methods focus on small-scale damage without relying on vibration response. Most non-destructive testing and evaluation techniques, such as ultrasonic testing, acoustic emissions, and infrared thermography, fall under local methods, such as inspecting structural components in specific areas. Electro-mechanical impedance-based approaches, for instance, use piezoelectric patches to detect local damage in small structures. While local methods aid damage identification, they are insufficient

for large-scale monitoring. A comprehensive SHM system should integrate local and global techniques for a complete structural assessment [46].

Vibration-based methods have been extensively researched for decades, assuming time, frequency, and modal analyses to identify, locate, and evaluate damage in engineering structures. Numerous algorithms have been developed across civil, mechanical, and aerospace fields. Global damage detection methods analyse structural vibration responses using strategically placed accelerometers, with collected data processed to detect damage. While qualitative techniques date back to the 1800s, quantitative methods became viable in the 1980s due to advancements in computing and sensing technologies [46]. Compared to local approaches, the vibration-based method requires fewer sensors, does not rely on predefined damage locations, and employs portable equipment. These methods are broadly categorised into nonparametric and parametric approaches, which are further examined in the following sections. Hence, the fundamental idea behind vibration-based damage identification is that damage-induced changes in physical properties such as mass, damping and stiffness can cause detectable changes in modal properties such as natural frequencies, modal damping and mode shapes [47]. Thus, as an indicator of stiffness reduction, shifts in natural frequencies and variations in modal parameters are commonly used in vibration-based structural health monitoring systems to assess structural integrity.

The advancements in reliable and low-cost sensors capable of measuring various structural responses (e.g., accelerations, displacements, strains, temperatures, and loads) have driven significant scientific and practical progress in SHM application over the past four decades, enabling the processing of raw measurement data into meaningful structural health indicators. However, despite these advancements, SHM remains largely confined to research and has not yet achieved widespread real-world implementation. Addressing this gap requires fast, on-demand monitoring solutions that leverage machine learning techniques to enhance real-time data processing, automate damage detection, and improve the scalability and efficiency of the monitoring methods in practical applications. Hence, with the integration

of artificial intelligence in the price, we can increase the value of information in vibration-based SHM. The value of information further highlights the advantage gained from utilising vibration data for early damage detection, risk mitigation, cost optimisation, informed decision-making, and improved uncertainty quantification, thereby enhancing overall structural safety and maintenance efficiency [48, 49].

2.2 Machine learning for structural health monitoring

Traditionally, SHM relied on physics-based approaches, which were limited by their applicability to simple structures and controlled environments. The advent of machine learning has leveraged the SHM approach by providing advanced tools for data analysis and damage detection, enabling more robust and comprehensive monitoring systems [50]. Conventional SHM methods faced difficulties such as incomplete monitoring data, uncertainties in structural conditions, and complex environmental factors affecting the parameters of the feature, e.g. modal properties and temporal responses. These limitations required the development of novel methodologies that could exploit the vast amounts of data generated by modern monitoring systems [50, 51]. Integrating the Internet of Things (IoT) and big data analytics has recently significantly enhanced SHM systems. These technologies facilitate collecting and processing large datasets, enabling near real-time damage assessment and decision-making [52, 53]. Hence, the evolution of the application of ML in SHM assessed early challenges and progress in integrating IoT and Big Data.

Indeed, machine learning techniques significantly enhanced the efficiency and accuracy of SHM systems by automating data analysis and improving damage detection [54, 52, 55]. These algorithms process large volumes of vibration data, identifying patterns and anomalies that may indicate structural deterioration. Unlike traditional methods that rely on manual interpretation, machine learning enables real-time monitoring and early issue detection by learning from historical and newly acquired data. Beyond vibration analysis, ML can integrate data from various sources,

such as temperature and moisture sensors, and visual inputs from cameras or drones. This multidimensional approach improves anomaly detection and enhances predictive capabilities. Additionally, ML algorithms continuously refine their performance, adapting to new data and increasing the accuracy and efficiency of SHM systems over time. These advancements contribute to early intervention, reducing the risk of structural failures.

ML techniques, particularly data-driven methods, have become essential in SHM for analysing complex structures where physical modelling is challenging. These methods include clustering, regression, and classification algorithms that can efficiently detect and predict damage in structures such as bridges, buildings, wind turbines, machinery, and aeroplanes, among others [52, 54]. Further, Deep learning (DL) has emerged as a powerful tool in SHM, offering advanced capabilities for vibration-based and vision-based monitoring. DL methods, such as deep neural networks and transfer learning, have been successfully applied to enhance the accuracy and reliability of SHM systems [56, 57].

Along with progress in the field of SHM integrated to ML, one still faces significant challenges in applying ML and DL to SHM, such as the lack of comprehensive sensor data for different damage scenarios, affecting ML models' robustness and generalizability. Physics-informed learning, which integrates domain knowledge into the ML process, is proposed as a solution to improve model performance [51]. The solution of sensor fusion and data augmentation [58] has shown promising solutions, but ongoing research remains. Overall, the integration of ML into SHM represents a significant advancement in the field, addressing many of the limitations of traditional methods. As technology continues to evolve, the combination of ML with IoT, big data, and emerging technologies will likely lead to even more sophisticated and effective SHM systems, ensuring the safety and improving the lifespan of the structures and systems.

2.2.1 Statistical pattern recognition

Structural health monitoring with machine learning and statistical pattern recognition (SPR) are closely related, as SHM integrated with ML relies on SPR principles for damage detection, classification, and prognosis [59]. The SPR encompasses various definitions and approaches. It includes all stages of an investigation, from problem formulation and data collection to analysis, classification, evaluation, and interpretation [60], and is further defined as the automatic discovery of regularities in data through algorithms that classify data into different categories [59]. In a general context, SPR is a methodology for identifying patterns in data using statistical techniques involving feature extraction, modelling relationships between features, and probabilistic decision-making.

Figure 2.1 is a scheme of the SPR process [43], which starts with a data acquisition system (e.g., sensors, cameras, etc.) responsible for capturing physical information and translating it into a measured signal. A pre-processing process eliminates noise or distortions, followed by a feature extractor (or attribute). Feature is the information collected from the signals that can represent the information acquired from the monitored system in a reduced or compact format. It is commonly applied to reduce data to attributes, properties, or characteristics. The next step is classifying and clustering these features to achieve the pattern recognition objective.

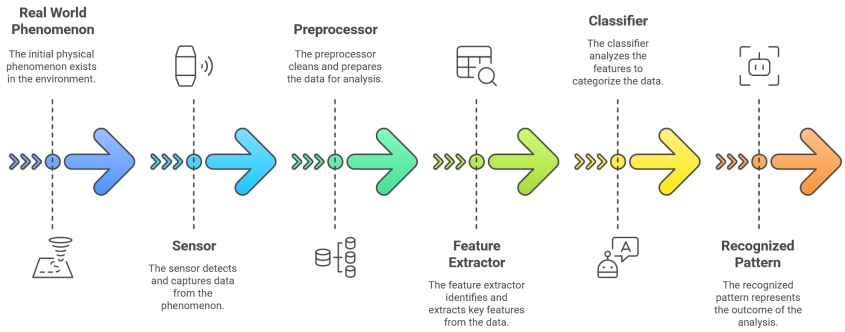


Figure 2.1: Statistic pattern recognition process (Source: own study).

In summary, SPR aims to classify objects into different categories or classes by analysing the characteristics presented by the features. In the context of damage identification, it seeks to detect structural changes by comparing the damaged state to the undamaged state. This process begins with collecting data from the monitored structure and ends with identifying damage to assess the structure's current condition [61]. Data extraction is generally categorised into two main approaches, the physics-based approach and the data-driven approach [55, 62]. The physics-based approach assesses structural integrity by updating a model grounded in physical principles, such as the finite element model (FEM), aiming to minimise discrepancies between model predictions and measured data. This method can provide an accurate and calibrated model for damage prediction. However, simplifications and omissions in the model may introduce errors, affecting parameter estimation, damage detection, and forecasts of structural behaviour. Data-driven approaches leverage pattern recognition to detect damage, determine its location, and assess its severity based only on structural response data [63]. An advantage of this approach is that it eliminates the need for complex numerical model development and validation while adapting to uncertainties caused by measurement variability. However, a significant challenge in data-driven SHM is obtaining sufficient accurately labelled training data to develop a reliable and generalizable statistical model [64]. In particular, damage detection in data-driven SHM is a supervised learning problem, where potential damage locations serve as target class labels for a machine learning classifier. This learning process requires training data from both undamaged and damaged conditions.

In the context of pattern recognition, Figueiredo and Brownjohn [61] recently published a review of the evolution of the SPR paradigm applied in bridge monitoring covering the past three decades. Their study highlighted key developments in detection technology, data analysis, and emerging trends to foster more coordinated and interdisciplinary research. The authors also addressed challenges in transitioning SHM from research to practical application, particularly in obtaining reliable global damage as-

sessments and advancing damage identification levels, including diagnosis (location, type, and severity) and prognosis. Similarly, Sen and Nagarajaiah [65] reviewed statistical learning techniques for SHM and damage detection, discussing the role of supervised and unsupervised learning algorithms. They concluded that statistical learning algorithms enhance the robustness and efficiency of SHM systems and reduce computational effort compared to model-based approaches, which often require high-fidelity simulations. This paves the way for developing real-time, online SHM systems capable of autonomous damage detection with minimal human intervention.

Autoregressive models have been employed to solve the SPR paradigm. Zhang et al. [66] reviewed linear and nonlinear structural identification methods that apply support vector regression (SVR) for pattern recognition. They presented three SVR-based approaches utilising ARMA time series, high-order AR models, and substructuring strategies to identify linear structural parameters from vibration data. The study also discussed SVR coefficient selection and incremental training algorithms. Numerical evaluations confirmed the accuracy of SVR-based methods in identifying structural parameters, highlighting their potential for SHM applications. Later, Gui et al.[67] investigated statistical pattern recognition methods for SHM through experiments on a supported steel beam and a complex steel grid structure. Their approach combined time series modelling, specifically autoregressive (AR) models, with outlier detection algorithms based on Mahalanobis distance to identify structural changes. Similarly, Entezami et al.[68] proposed an approach to efficiently handle big data in structural integrity monitoring using the SPR. Feature extraction was performed through autoregressive moving average (ARMA) modelling, and a novel partition-based Kullback-Leibler divergence nearest neighbour (PKLD-NN) method was introduced for damage detection. The proposed method was validated using high-dimensional experimental data from the Tianjin Yonghe Bridge, demonstrating its effectiveness in SHM applications.

Herrera-Iriarte et al. [69] explored a methodology for SHM based on recognising deformation field patterns using sensors capable of measuring

deformations at discrete points and machine learning techniques to detect structural damage. In this study, deformation data from fibre optic sensors (FOS), specifically fibre Bragg gratings (FBG), acquired through two experiments were used: an aluminium beam with 32 FBGs and a CFRP beam with 20 FBGs, which serves as the main wing of the structure of an unmanned aerial vehicle (UAV). In the experiments presented, the beams were equipped with different numbers of sensors, which were removed one by one to analyse the sensitivity of the PCA-based damage detection methodology to changes in the number of sensors. The results show that only a few sensors contribute significantly to the methodology's performance, and these sensors are validated as those located close to the analysed damage condition.

Machine learning algorithms have been employed in SPR techniques for structural damage detection. Trendafilova and Heylen [70] explored using artificial intelligence techniques for damage detection and localisation in structures. The structure was divided into substructures in their approach, and pattern recognition techniques were applied to identify the damaged substructure. Frequency response functions (FRFs) were used for a defined number of degrees of freedom and frequencies, serving as the foundation for the detection process. A mapping was established between the feature space extracted from the FRFs and the space defined by the dynamic response of the structure in the frequency domain. Based on this mapping, standard vectors and samples representing different damage classes were obtained. Finally, a computer code (classifier) was developed to utilise the pattern recognition information for damage localisation within the structure.

Qiao et al. [71] employed a signal-based SPR procedure for structural damage detection with a limited number of input/output signals. The method involves extracting and selecting sensitive features from the structure's response, creating a unique pattern for each specific damage scenario, and comparing the unknown damage pattern with a known database to identify the damage's location and severity. In the study, two transformation algorithms, Continuous Wavelet Transform (CWT) and Wavelet

Packet Transform (WPT), were separately implemented for feature extraction, while three pattern matching algorithms—correlation, least squares distance, and Cosh spectral distance—were used for pattern recognition. Experimental studies conducted on a simple three-storey steel structure showed that the signal characteristics for different damage scenarios could be uniquely identified, with the correlation algorithm providing the best performance in recognising the unknown damage pattern. This method is not only effective for damage location identification but also has the potential to detect damage type, making it suitable for online structural health monitoring applications. Later, in an experimental study, Shan et al. [72] proposed a method for detecting structural damage in a continuous girder railway bridge by combining a step-by-step damage detection approach with statistical pattern recognition, the identification process covered early warning of damage, localisation of the damage, and determining its extent. The Support Vector Machine (SVM) multiclass classification algorithm was used for damage location identification, while the Support Vector Regression (SVR) algorithm was employed to assess the severity of the damage. The results demonstrated that the proposed method successfully and accurately identified the damage's location and extent with high accuracy.

Deep learning algorithms have also been employed in the SPR process. Goswami and Bhattacharya [73] proposed an SPR-based damage detection scheme for aerospace vehicle structures. This method involves collecting mechanical vibration signals from plate-like structures using displacement sensors, followed by noise removal and feature extraction through Wavelet Transform-based signal processing techniques. A set of neural networks is then trained to classify and identify the damage present in the structure. While the case studies showed promising results for classifying individual damages, identifying multiple damages in the same structure revealed a decline in success rates. To enhance these results, the authors developed a sensor positioning strategy that significantly improved the accuracy of detecting multiple damages.

Perafán-López and Sierra-Pérez [40] presented a pattern recognition

methodology for clustering operating conditions in a structural sample using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm in the context of SHM. This methodology was validated using data from an experiment with 32 fibre optic Bragg sensors attached to an aluminium cantilever beam subjected to cyclic bending loads at 13 different operating conditions (inclination angles). Additionally, the computational cost and accuracy of the machine learning pipeline, FA+GA-DBSCAN (which combines factor analysis for dimensionality reduction and a genetic algorithm for automatic DBSCAN parameter selection), were evaluated. The results demonstrated good performance, detecting 12 out of the 13 operating conditions with an overall accuracy exceeding 90%. Li et al. [74] proposed a new Generalised Automatic Encoder (NGAE) integrated with a statistical pattern recognition approach, utilising cepstral power coefficients of structural acceleration responses as damage-sensitive features (DSFs) for structural damage assessment. The method was validated through numerical simulations and experimental data, demonstrating superior performance to traditional Auto-Encoder (TAE) and Principal Component Analysis (PCA) methods.

In recent years, the application of SHM and ML has experienced vast growth in methods and applications aimed at improving the reliability and maintenance of these systems. In this study, we focus our SHM-ML technique developments and tests on detecting and diagnosing faults in wind turbine components, such as blades and rotors, and coupling mechanisms, such as flexible coupling components and fixed bolted joints.

2.3 Application of SHM integrated with ML in wind turbine components

The wind energy industry faces the challenge of ensuring the efficient and safe operation of wind turbines subjected to extreme mechanical loads and harsh environmental conditions during operation. In this regard, damage detection methodologies play a key role, as they make it possible to identify structural problems before they become critical, guaranteeing the

integrity and longevity of these structures. To reduce costs and improve operational efficiency, it is vital to use a reliable SHM methodology capable of detecting structural defects. Thus, the use of SHM associated with ML has been successful in several applications in monitoring wind turbines [28, 75, 76, 77], allowing precise and automated procedures to be developed. Among the various techniques available, systems based on vibration analysis show great potential for structural monitoring and fault diagnosis [28, 75, 76, 77].

The wind turbine is a sophisticated mechanical-electrical system. A turbine's main mechanical components and structures that have been monitored include the blades, main bearing, main shaft, gearbox, nacelle, tower, foundation, yaw system and bolted systems. The common failure types and cases of the main components have been introduced in Table 1.1. SHM methods applied to wind turbines were investigated and summarised. Thus, wind turbine component fault detection using the proposed SHM-ML process is the main focus of this study, specifically in damage conditions associated with the blade, rotor and flexible and fixed coupling.

2.3.1 Blade monitoring

SHM integrated with ML applied to wind turbine blades have been extensively studied by several researchers. In one of the earlier studies, [78] explored machine learning techniques for monitoring turbine blades using vibration data, particularly Frequency Response Function (FRF) measurements. The research emphasised low-level fault estimation to determine the presence or absence of damage using a Multilayer Perceptron (MLP) and a novel approach to self-association using Radial Basis Function (RBF) networks. Building on this, [79] investigated new turbine blade fault detection using experimental vibration analysis and machine learning techniques. The study explored Nonlinear Neural Networks, including Auto-Associative Neural Networks (AANNs) and Radial Basis Function (RBF) network models. In [80], the authors examined machine learning algorithms based on Auto-Associative Artificial Neural Networks (AANNs), proposing a stan-

dard ANN-based AANN and a novel approach for auto-association using RBF networks, and in [81], it is discussed advancements in SHM and condition monitoring of wind turbines, focusing on damage detection through data-driven vibration analysis. The study presented existing technologies for turbine blades, employing a pattern recognition and machine learning approach, such as ANNs and Gaussian processes, alongside SCADA data.

In [82], the authors presented an algorithmic classification of vibration signals for assessing the condition of wind turbine blades considering five faults (blade crack, erosion, loose hub-blade connection, pitch angle twist and blade bend) for the diagnosis of wind turbine blade faults. ML techniques were used for feature extraction, selection, and classification based on a decision tree algorithm. The functional trees algorithm is suggested for diagnosing wind turbine blade faults. In the work by [83], a three-layer structural integrity monitoring framework is employed on experimental data from a 34 m rotor blade for damage and icing detection. Modal parameters identifying the system and other damage characteristics, also called condition parameters, are presented and compared with each other. The focus of the study was to investigate the effect of varying environmental and operating conditions (EOCs) on structural dynamics and to explore the contribution of unsupervised ML by data clustering to increasing detection performance. It is shown that detection performance in the case of data clustering according to the equivalent damage load applied is higher than without data clustering. Subsequently, [84] used twelve rule-based feature classifiers using WEKA for wind turbine blade fault diagnosis, considering the following methods: Conjunctive Rule (CR), Decision Table, Decision Table and Naive Bayes hybrid classifier (DTNB), JAVA implemented repeated incremental pruning to produce error reduction (JRip), Non-Nested generalised exemplars (NNge), One Rule (OneR), Projective Adaptive Resonance Theory (PART), Ripple down rule learner (Ridor), Zero Rule (ZeroR), Fuzzy Unordered Rule Induction Algorithm (FURIA), modified learnable evolution model (MODLEM) and classifier Ordinal Learning Method (OLM).

Chandrasekhar et al. [85] proposed a diagnostic methodology for oper-

ational wind turbine blades using Gaussian Processes (GPs) for predictive purposes, and the residuals between the actual signals and the predicted signals can be used as an informative indicator of damage. The proposed SHM methodology can identify when blades begin to behave differently from each other over time. More recently, [86] contributed to this field by focusing on detecting structural faults in turbine blades by analysing tower vibrations. The study developed a Convolutional Neural Network (CNN) classifier to differentiate between tower vibrations collected under healthy and faulty blade conditions. In [87], the authors present the structure of signal collection, feature extraction and classification techniques for predicting blade failures. Classifier models such as Naive Bayes (NB), multilayer perceptron (MLP), linear support vector machine (linear SVM), single depth convolutional neural network (1DCNN), bagging, random forest (RF), XGBoosts, and decision tree (DT) were used, and the results were compared according to their parameters to propose a better fault diagnosis model.

2.3.2 Foundation and tower monitoring

Wind turbines rely on various support structures, including foundations and towers, which are also susceptible to damage [3]. SHM applied to these components provides crucial insights into their condition, facilitates maintenance, and helps prevent catastrophic failures. Vidal et al. [88] propose a methodology for diagnosing structural damage in jacket-type foundations, specifically investigating crack damage at four locations. Their approach employs a damage detection and localisation method, where the latter is treated as a classification problem, using k-NN and SVM on vibration response data from accelerometers. Similarly, Jersson et al. [89] develop an SHM methodology that integrates data pre-processing, principal component analysis (PCA) for dimensionality reduction and feature extraction, and XGBoost-based classification for damage detection in offshore wind turbine support structures. Their method was validated on a small-scale model with five structural states: an intact structure and a 5mm crack at four locations. Additionally, Liying [90] explores tower failures, introducing

a K-means fault clustering classification model optimised through a dynamic weight algorithm.

Throughout their operational life, wind turbines are continuously exposed to multiple hazards from various load types, including torsional vibration and tower side-to-side and fore-aft bending. These factors can significantly impact their long-term structural health and performance [3]. Hoxha et al. [91] propose an SHM strategy that relies solely on vibration response, employing machine learning methods for damage diagnosis. Their study utilises KNN, quadratic SVM, and Gaussian SVM for classification. Similarly, Nguyen et al. [92] investigate wind turbine tower damage assessment through vibration-based artificial neural networks (ANNs). Their approach uses modal parameters, such as mode shapes and frequencies, as inputs, with element stiffness indices as outputs. A FE of a real wind turbine tower serves as the test structure, and the trained ANNs are then applied to detect damaged elements and assess severity levels.

2.3.3 Gearbox and rotor monitoring

The nacelle of a wind turbine accommodates the drivetrain system, which includes the gearbox, rotor, main bearing, main shaft, yaw system, hub, and generator. Drivetrain component failures are among the most critical challenges throughout a wind turbine's operational lifespan. These failures vary in nature and are often exacerbated by prolonged exposure to harsh conditions, such as heavy loads, wind gusts, and dust-induced corrosion. Common issues, including rotor imbalance, rotor icing, misalignment, structural damage, bearing and gear failures, and generator breakdowns, are typically associated with excessive vibration, oil leakage, elevated oil temperatures, inadequate lubrication, and impact forces[3]. Analysing wind turbine gearbox vibrations for SHM is explored in [93], where a neuromorphic machine learning model is developed to analyse time-series accelerometer data for fault detection. A neuromorphic neural network was trained on the back to classify accelerometer data from both healthy and damaged gearboxes. Similarly, Praveen et al. [94] propose a simplified signal

segmentation technique that aligns non-stationary vibration signals with specific speed stages and gearbox components. This technique is validated using machine learning algorithms, including Decision Tree, Support Vector Machine, and Deep Neural Network.

Elforjani et al. [95] investigate fault classification and detection in wind turbine gearboxes using ANNs, Decision Trees (DTs), Gaussian Processes (GPs), Mixture Discriminant Analysis (MDA), and Support Vector Machines (SVMs). Their study first extracts twelve statistical features from the vibration dataset, followed by Principal Component Analysis (PCA) to enhance data visualisation and reduce dimensionality, improving classification accuracy. Gao et al. [96] propose a linear discriminant diagnostic method based on convolutional neural networks (CNNs) for detecting and diagnosing coexisting mechanical faults in operational wind turbine bearings and gearboxes using vibration signals.

Vives et al. [97] apply KNN and SVM techniques for diagnosing and preventing faults in wind turbine bearings through vibration analysis, achieving high predictive accuracy. In a related study, Vives et al. [98] implement SVM and deep learning for fault monitoring and diagnosis. Abdelrahman et al. [99] introduce a combined vibration analysis and CNN-based approach (Cyclostationary-based CNN and Kurtogram-based CNN) for detecting and classifying faults in wind turbine gearboxes. Lastly, Angela et al. [100] integrate autonomous data-driven learning of fault signatures with integrity state classification using CNNs and isolation forests.

2.3.4 Flexible and fixed assembly coupling

Flexible coupling failure in wind turbines is a critical issue affecting the reliability and efficiency of wind energy systems. This topic encompasses the study of dynamic interactions between various components of wind turbines, particularly focusing on the flexible couplings that connect the gearbox and generator. These couplings are essential for accommodating misalignments and reducing stress on the turbine components. Flexible couplings in wind turbines often fail due to misalignment and the resulting reaction loads.

Joint kinematics, metal disk pack deformations, and axial and angular shaft misalignments influence these loads. Such misalignments can lead to early, unplanned bearing failures in the gearbox and generator high-speed shafts, significantly reducing their lifespan [101].

The dynamics of rigid-flexible coupling in wind turbines are complex and involve interactions between various structural components. Studies have shown that the dynamic response of wind turbine components, such as blades and towers, is significantly affected by external forces like wind loads and gravity, leading to deformation and vibration. These dynamics are crucial for understanding the stability and reliability of wind turbines [102]. Understanding and addressing flexible coupling failures are vital for improving the reliability and efficiency of wind turbines. By accurately modelling the dynamic responses and optimising the design of flexible couplings, the operational lifespan of wind turbine components can be extended, reducing downtime and maintenance costs [101, 103, 104]. These insights are crucial for developing and deploying wind energy technologies.

Another important coupling mechanism used in wind turbines is bolts. It is crucial in wind turbine towers, connecting all tower sections and blade root connections. However, bolt looseness is a common deficiency and is prone to occur due to long-term vibrations, fatigue, corrosion, bolt relaxation and incorrect bolt pre-tensioning procedures. These failures can lead to the collapse of wind turbine towers in severe cases, which makes bolt looseness a major concern. Damage detection methods applied to bolted joints have been studied, focusing mainly on two components/structures, the tower flange [105, 106, 107, 108, 109, 110, 111, 28, 112, 113] and blade root connections [114, 115, 116, 117, 118, 119]. Machine learning techniques were investigated for detecting bolt loosening in wind turbines based on vibration analysis in [120]. They developed a FEM of the bolted connection, tower flange, to study the bolt loosening process under transverse vibration conditions. A prediction model for bolt loosening based on machine learning Gaussian process regression (GPR) was developed to obtain confidence intervals for the preload variation in a probabilistic sense with vibration cy-

cles in different working conditions. The result shows that, under the action of transverse vibration loading, the magnitude of the vibration load is the main factor affecting loosening, and the greater the magnitude of the load, the greater the probability of loosening occurring.

2.3.5 Bolted structures monitoring

Machines and structures are assembled and fixed together from individual parts, which can be joined through mechanical, chemical, or physical means [121]. There are various fastening techniques, some of which are permanent, while others allow joints to be repeatedly assembled and disassembled [122]. Mechanical joining involves connecting two or more engineering components (or elements) to form a functional unit. This process is achieved using devices called fasteners, which result in mechanical fixing [123]. Various types of fasteners, such as screws, nails, bolts, rivets, and others, are used to secure the parts of an assembly. The joint is typically formed by an ordered matrix of fixing points or lines, creating a discontinuity in the structure. There are many types of joints, some designed for specific applications and others, like overlap joints, being more common. Mechanical joints are crucial and often critical to any assembly or engineering structure, as they rigidly connect substructures and enhance the dynamics of the assembled system [124].

Bolted joints are widely used in engineering structures, such as civil and mechanical structures, due to their simplicity of design, ease of assembly and disassembly, reliability, high load capacity and relatively low cost. They consist of a bolt, a nut, two contact parts and sometimes washers, as illustrated in Figure 2.2. The tightening force, provided by the tension in the bolt, connects the components and applies a preload or pre-tension, placing them in compression to increase resistance to static or cyclic loads. Tightening is important to ensure the proper functioning of the joint and the force required to hold it together. The tightening process guarantees the quality and integrity of bolted joints. However, maintaining tightening accuracy is a challenge, as there is a degree of uncertainty in the preload

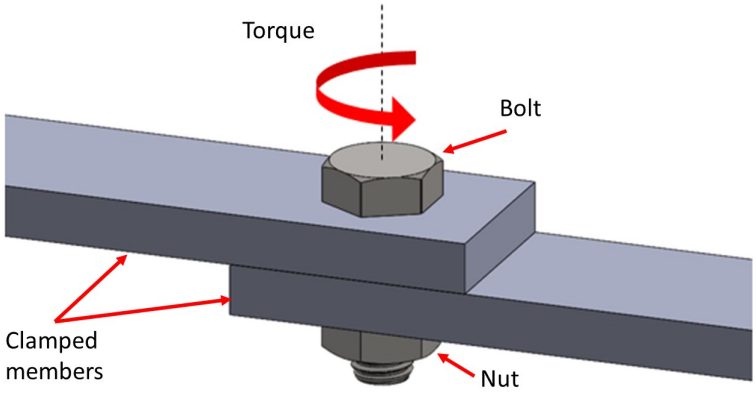


Figure 2.2: Schematic representation of bolted joints considering the principal elements: structure, bolt, nut, and torque force (Source: own study).

on each joint [125]. In addition, it is difficult to tighten several bolts simultaneously due to the limited number of devices available, and tightening is done alternately in many passes to approximate the design load [126]. It is important to ensure an adequate tightening force for the bolt; in many cases, inadequate force can cause damage to the structure due to looseness, while excessive force can also impose stresses on the bolt, which reduces its efficiency. Therefore, it is crucial to accurately measure the bolt tension and detect and quantify the degree of tightness of the joint.

Failure of bolted joints

Although bolted joints offer advantages, it is important to note that they can fail during operation. In some situations, failures in bolted joints have serious consequences and are safety-critical. Fatigue, stress concentration, cracking, corrosion, high temperatures and mechanical failure during assembly are the main causes of bolt failure. Too tight bolts can contribute to bolted joint failure, but the event is uncommon. Excessive loads can stimulate stress corrosion cracks, crushing or damaging surfaces [125]. Bolt loosening is a form of damage or failure in structures and is the most common and relevant problem in bolted joints [126]. Loosening can be defined as a gradual loss of preload after the tightening process has been completed,

which over time is inevitable, making the fastening unstable due to external dynamic loads such as impact, vibration and thermal loads [127]. Early detection is, therefore, essential to ensure the safety of the connection.

Vibration is widely recognised as the primary cause of bolt loosening [128]. When a joint is subjected to unintended movement, the preload can diminish or be entirely lost, leading to mechanical failure [129]. Additionally, insufficient preload can result in fatigue fracture of the bolt under vibrational stress. Research into vibration-induced loosening has been conducted for decades. A recent review study by Gong et al. [130] examines the factors that initiate rotational loosening, including axial, transverse, torsional, and bending vibration loads. The authors suggest that their work will significantly advance the understanding of the subject and serve as a valuable resource for engineers. Bickford and Oliver [131] also explore vibration loosening in their book.

Experimental studies presented in [132] used a specialised test machine to demonstrate that transverse vibrations have more detrimental effects on bolt loosening than axial vibrations. Similarly, Pai and Hess [133, 134] conducted experimental studies on screw loosening under dynamic loads. Their findings revealed that fasteners can loosen at lower loads than anticipated due to localised sliding on the contact surfaces. Later, the authors [134] developed a three-dimensional FEM to simulate loosening. This model accurately represented the characteristics observed in the experimental data, including the key factors contributing to loosening. Chen, Gao, and Guan [135] proposed a FEM using a hexahedral mesh, integrating both the tightening and loosening processes to analyse the effects of different fastening media.

Machine learning applied in bolted joints monitoring

Over the years, significant advancements have been made in detecting loosening in bolted structures [136, 126]. These techniques are in situ inspection, computational vision, and sensor-based techniques. In-situ inspection techniques, such as using a torque wrench and hammer, can be

visual or mechanical [127, 129, 137]. Digital cameras or images are used in computer vision-based techniques, while the vibration-based method, wave propagation [138, 139, 140], acoustoelastic effect-based method, piezoelectric sensor-based methods, and impedance-based method are used in sensor-based techniques [129].

ML algorithms have recently been utilised to detect and monitor bolt torque loosening. Machine learning is a set of methods that can automatically detect patterns in known problem data, which can then be used to develop predictive models and carry out decision-making in uncertain conditions [141]. The ML technique learns from the available data and obtains a model that can make accurate predictions [142]. These algorithms are categorised into supervised and unsupervised classification learning techniques and regression algorithms. The supervised learning technique creates a model from a set of labelled training data using previously known input and output values, and it is subdivided into classification and regression problems. Using multiclass supervised machine learning algorithms, Sousa et al. [143] accurately assess the damage of a beam reinforced by masses from its spectral response. The authors used ML to classify the beam's damage, where the methodology involves experimental measuring and numerical calculation of the dynamic features, such as natural frequency and frequency response function, to construct two DIs. In contrast, the unsupervised learning approach does not require target class labels in the training data[141]. The regression algorithms help in defining the relationship between labels and data points. An overview of the literature related to the work is presented here to explain the application of the ML technique in detecting looseness in bolted joints.

In situ inspection techniques to monitor torque loosening have been explored by Zhou et al. [144], where the authors describe a study that used percussive methods and machine learning to detect loosening in bolts. The experiment was conducted on a four-bolt steel beam-column joint, where laser Doppler vibrometry was used to capture the vibration information of the test bolt, while microphones collected acoustic sounds generated by an

automatic hammer. The authors then transformed the reconstructed sound database into spectrograms and trained a 2D-CNN to identify bolts' loosening conditions. Wang and Song [145] presented a novel one-dimensional training interference capsule neural network (1D-TICapsNet) to process and classify percussion-induced sound signals to detect bolt early looseness in two steel pieces tightened using four bolts. They also employed in [146] the multifractal analysis and joint mutual information maximisation method to extract feature sets and detect bolt loosening using the gradient boosting decision tree (GBDT) algorithm. Tran et al. [147] investigated the application of a deep convolutional neural network (DCNN) algorithm to detect and estimate looseness in bolted joints using a laser ultrasound technique. Zang et al. [148] presented a method of detecting screw loosening in iron plates based on audio classification using SVM. Kong et al. [149] proposed a new approach to identifying bolt clearance levels in a twelve-bolt subsea flange using an ML model with the decision tree method, similar to the percussive diagnostic techniques used in clinical examinations.

Computer vision-based machine learning techniques to torque loosening were proposed by Gong [130]. In [150], the authors use a combination of deep learning algorithms and geometric image theory for detecting loosened bolts in a steel pedestal through vision-based bolt loosening. The method uses a faster regional convolutional neural network (Faster-RCNN) and a waterfall pyramid network (CPN) algorithm. Zhang has conducted similar research [151] and Yu [152] using the Faster R-CNN and single-shot multibox detector (SSD) algorithm, respectively, for detecting bolt loosening angles. Pham et al. [153] used synthetic images of bolts generated from a graphical model to train a deep learning model based on the Region-based Convolutional Neural Network (R-CNN) algorithm for detecting loose bolts. Ramana et al. [154] used machine learning techniques, including the Viola-Jones algorithm and SVM, to detect loosened bolts on a steel I-section. Similarly, Chan et al. [155] used the Hough transform and SVM to build a classifier for detecting loosened bolts.

Several investigations have employed various techniques and methods

to detect anomalies and identify bolt loosening in engineering structures. For instance, Razi et al.[156] have utilised sensor-based techniques, ML based on wave propagation, and modal methods. In the study by Ziaja et al.[157], elastic wave propagation was employed to detect anomalies in the prestressed connections of engineering structures, utilising a combination of Artificial Neural Networks (ANN). Eraliev et al.[127] detected and identified loosening bolts in a multi-bolt structure using seven ML algorithms, namely Random Forest, Bagged Trees, Decision Tree, k-neighbour, Linear Discriminant Analysis, SVM, and XGBoost. The author utilised the Short-Time Fourier Transform (STFT) method for feature extraction from acquired vibration data. Miguel et al.[158] observed the loss of tightening torque in bolted joints by employing modal parameters. Teloli et al. [159] utilised two probabilistic ML methods, namely the Gaussian mixture model (GMM) for damage detection and Gaussian process regression (GPR) for quantifying loosening torque in lap-joint structures. Chen et al. [135] proposed a diagnostic method for detecting looseness in fan foundation bolts based on the mixed domain characteristics of excitation response and multiple learning. The study employed the K-weight nearest neighbour classifier (WKNNC) to identify slacks. Zhuang et al. [160] employed the acoustoelastic effect-based method along with several ML algorithms, such as the recurrent neural network LSTM, one-dimensional WideResnet40, one-dimensional Densenet121, XGBOOST tree classification model, LightGBM, and the SAX-VSM algorithm. Wang et al.[161, 162] proposed the Siamese Double-path CapsNet (SD-CapsNet) and the Genetic Algorithm-based Least Square Support Vector Machine (GA-based LSSVM) for bolt loosening detection, using piezoelectric sensor-based methods. Zhou et al. [163] applied an impedance-based method using the Graph convolutional networks (GCN) model in another approach. Hence, these studies have utilised various ML algorithms to tackle complex real-world problems in diverse applications. Most proposed techniques or processes combining SHM with ML are based on hybridising multiple interacting numerical procedures [164]. This complexity poses challenges when implementing an effective solution.

2.4 Chapter final remarks

Based on the discussions presented in this chapter, it is evident that the field of Structural Health Monitoring still holds significant potential for further exploration, particularly in detecting wind turbine component faults and, specifically, bolt loosening. This work uses machine learning and SHM techniques, focusing on vibration signature analysis, to identify bolt torque loosening and wind turbine operation faults under various damage scenarios, such as blade aerodynamic imbalance, rotor icing, and flexible coupling failure. These advanced techniques offer several advantages, including early damage detection and operation fault, precise identification of damage patterns, enhanced inspection accuracy, service life prediction, and improved maintenance efficiency. Collectively, these contributions can significantly enhance the reliability and durability of wind turbines. Given the limited number of studies conducted in this area relative to its importance, there is a pressing need to develop and utilise accurate and reliable tools. This research seeks to advance the understanding and application of SHM in wind turbine maintenance by addressing this gap, ultimately supporting safer and more efficient wind energy systems.

3 Methodology and Methods

The methodology process for wind turbine components' condition assessment, including detection and quantification of damage using ML model. The methodology strategy includes processing the existing data acquired from vibration tests, feature extraction, data augmentation strategies through the virtual sensor, feature selection, classification damage, and performing the detection and regression algorithms to quantify the damage with its uncertainty quantification.

The SHM-ML process proposed in this work turns to an Open-code tool design in Python language and named "Machine learning for Damage Assessment" [165], patented under number INPI-BR10202401528 [166] and computed code registration number BR512024001008-4. The proposed SHM-ML process encompasses eight steps in total, comprising receiving the normalised acquired data (Step 1), supervised data processing (Step 2), feature selection (Step 3), data augmentation (Step 4), and unsupervised pattern recognition, labelling, and clustering (Step 5). These steps form the data-driven processing and SPR. Subsequently, the data splitting is performed in Step 6. In the supervised stage, classification ML algorithms (Steps C7 and C8) are used for damage detection, and regression (Steps R7 and R8) is applied for damage and uncertainty quantification. The algorithm supplies information regarding the damage condition based on the classification and regression algorithm outcomes.

Figure 3.1 shows the SHM-ML methodology for damage detection and estimation, listing each process step. The two main routes described are related to damage detection and classification and quantifying damage severity and uncertainty propagation. For the classification, the algorithms k-nearest neighbour (k-NN), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB) and XGBoost are employed. The regression algorithms implemented are the Linear. For damage quantification, we employ nine regression algorithms: linear regression, Lasso, KNR, DTR, GBR, SVR-linear, SVR-RBF, SVR-Poly, and MLP.

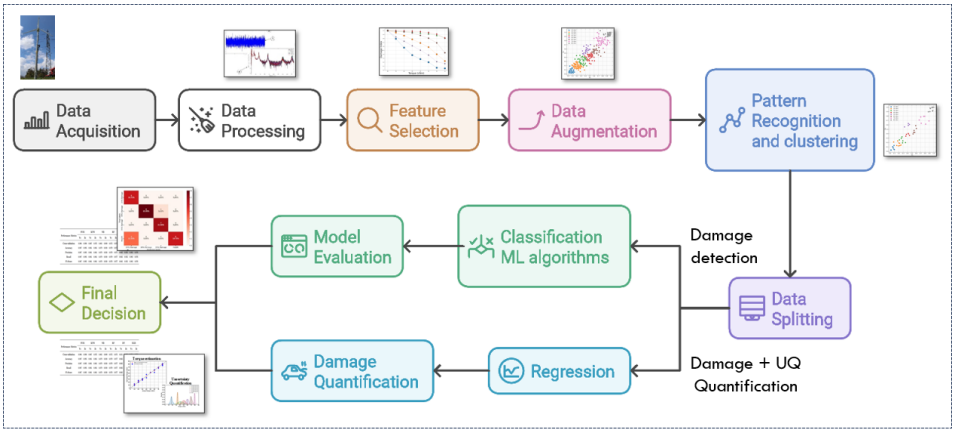


Figure 3.1: Schematic representation of the Structural Health Monitoring methodology using machine learning, outlining each step of the damage detection and quantification methodology[165].

In brief, the SHM-ML architecture is designed for the detection, quantification and prognosis of structural integrity based on the measured dynamic response of the structure [165]. It integrates the ML algorithms into the SHM process. This approach exclusively relies on driven data from the monitored system, eliminating the need for numerical models. The proposed method significantly improves torque estimation accuracy, enhances model performance in damage-level predictions, and supports decision-making. This section details each step of the SHM-ML method and the methods and techniques used to model.

3.1 Algorithms descriptions

Structures and systems vary, and the approach used in this work focuses on individual monitoring. Therefore, each system must be evaluated independently, taking the following considerations into account beforehand:

- Identify the specific damage and faults under consideration.
- Determine the information available from the system, e.g., temporal responses, frequency spectrum, or modal properties.
- Assess whether both normal and failure conditions are provided.

- Gather information about the damage or fault, including its natural causes, probable components affected, and possible locations.
- Recognize that while the existence and location of damage can be identified using unsupervised learning, determining the type and severity of damage generally requires supervised learning (Axiom III) [167].
- Acknowledge that sensors do not directly measure damage; feature extraction through signal processing and statistical classification is necessary to convert sensor data into meaningful damage information (Axiom IVa) [167].
- Ensure appropriate feature extraction, as improper methods can lead to high sensitivity to operational and environmental variations rather than actual damage (Axiom IVb) [167].
- Consider whether data scarcity will pose challenges for monitoring.

After mapping and planning the monitoring strategy for the system under evaluation, the proposed SHM-ML method is recommended. This study investigates four applications that utilize dynamic input information: one based on modal properties, specifically natural frequencies, as detailed in [143], two relying on frequency spectrum analysis, and one using temporal response data. Beyond damage classification and quantification, the challenge of data scarcity is addressed by integrating a virtual sensor into the model for data augmentation. The SHM-ML subroutine to process temporal responses for fault detection and classification in Algorithm 3 (Alg3), utilize frequency response data for damage detection and quantification in Algorithm 1 (Alg1), and enhance damage quantification with data augmentation in Algorithm 2 (Alg2).

Algorithm 1 SHM-ML for damage classification (Alg1)

1: **Data Acquisition:** Receive structural modal parameters data as the natural frequency or frequency response $[H(\omega)]$.
2: **Data Processing:** Correcting inconsistencies, handling missing values, and implementing signal processing techniques to convert raw data into structured and reliable.
3: **Feature Extraction:** Employing damage index, e.g. FRAC-DI as:

$$FRAC_{ij} = \frac{\left\| H_{ij}^{dam}(\omega) \left(H_{ij}^{und}(\omega) \right)^* \right\|^2}{\left[H_{ij}^{und}(\omega) \left(H_{ij}^{und}(\omega) \right)^* \right] \left[H_{ij}^{dam}(\omega) \left(H_{ij}^{dam}(\omega) \right)^* \right]}$$

5: **Pattern Recognition and Clustering:** Use the K-means unsupervised algorithm to cluster the data.

$$J = \sum_{i=1}^{n} min_k \left(\|x_i - \overline{x}_k\|^2 \right)$$

6: **Data Splitting:** Split the dataset into training and testing, e.g.80-20%, respectively.
C7: **Classification ML algorithms:** Applied ML algorithms for detection (SVM, K-NN, RF, NB, DT and XGBoost).
C8: **Model Evaluation:** Calculate the Performance Measure for the ML classifiers (Cross-validation, Accuracy, Precision, Recall, and F1-score).
9: **Final decision:** Information about damage state based on classification and regression algorithm outcomes.

Algorithm 2 SHM-ML for damage quantification and data augmentation (Alg2)

1: **Data Acquisition:** Receive structural modal parameters data as the natural frequency or frequency response $[H(\omega)]$.
2: **Data Processing:** Repairing raw data, checking for missing values, applying signal processing techniques to obtain a structured and usable format, and truncating the signal among the mode shape more influenced by the torque loosening;
3: **Feature Extraction:** Employing damage indices metrics based on the transmissibility response using FRAC, FAAC, AIGSC, AIGAC, M-DI, R-DI method discussed in section 3.2.2, and normalise the data and defining a threshold to identify outliers;
4: **Feature augmentation and fusion:** Employing statistic method, Tabular GAN, Forest Diffusion, and multiple DIs to increase the volume of data in the dataset, aiming to improve the performance of the regression algorithms on damage quantification. The proposed data augmentation architecture is detailed in section 3.2.4
5: **Clustering:** Use the K-means unsupervised algorithm to cluster the data, see subsection .

$$J = \sum_{i=1}^{n} min_k \left(\|x_i - \overline{x}_k\|^2 \right)$$

6: **Data Splitting:** Split the dataset into training and testing, e.g.80-20%, respectively.
R7: **Regression ML algorithms**: Applied ML algorithms for quantification (Linear regression, Lasso, KNR, SVR-kernels, DTR, Gradient Boosting Regressor, see section 3.3.4).
R8: **Model Evaluation:** Calculates evaluation metrics for regression models, including the coefficient of determination (R^2), MAE, MSE, and RMSE.
9: **Final decision:** Information about damage state based on classification and regression algorithm outcomes.

Algorithm 3 SHM-ML for damage assessment - Time series data (Al3)

1: **Data Acquisition:** Receive structural from the time-domain responses.
2: **Data Processing:** Involves correcting raw data by checking for missing values and applying signal processing techniques.
3: **Feature extraction and normalisation:** Fourteen techniques are applied to extract features from the time-domain signal, see section 3.2.3
5: **Pattern Recognition and Clustering:** Use the K-means unsupervised algorithm to cluster the data.

$$J = \sum_{i=1}^{n} min_k \left(\|x_i - \overline{x}_k\|^2 \right)$$

6: **Data Splitting:** Split the dataset into training and testing, e.g.80-20%, respectively.
C7: **Classification ML algorithms:** Applied ML algorithms for detection (SVM, K-NN, RF, NB, DT and XGBoost, see section 3.3.3).
C8: **Model Evaluation:** Calculate the Performance Measure for the ML classifiers (Cross-validation, Accuracy, Precision, Recall, and F1-score).
9: **Final decision:** Information about damage state based on classification and regression algorithm outcomes.

The overview of the algorithm's steps is detailed in the following:

2 **Data processing:** This stage involves repairing raw data, checking for missing values and applying signal processing techniques to transform data into structured and reliable information. In addition, inconsistencies are corrected, and missing values are dealt with to ensure data integrity. Duplicated and unbalanced data is also performed in the supervised step. (Applied to Alg1, Alg2, and Alg3)

3 **Feature extraction:** Using a data-driven approach involves understanding and examining its inherent features before integrating it into the machine learning model for accurate classification and estimation. Specifically, when the data involves employing the spectrum response of the structure, as normalisation is indicated, this step, or feature extraction, involves transforming the original data variables to create a new dataset, as described in [59]. This data-processing step converts raw experimental data into a normalised value using the DIs [168, 143, 38]. The DIs are split into two feature attributes DI_1 and

DI_2, serving as input data to the ML algorithms. (Applied to Alg1, Alg2, and Alg3)

4 **Feature augmentation and fusion by virtual sensor:** Employing statistic method, Tabular GAN, Forest Diffusion, and multiple DIs to increase the volume of data in the dataset, aiming to improve the performance of the regression algorithms on damage quantification. Section 3.2.4 details the proposed data augmentation architecture. Data fusion is the process of combining information from multiple sensors to enhance the fidelity of the damage detection process [55]. In our model, data fusion is performed using experimental and augmented data from the virtual sensor.(Alg2)

5 **Pattern recognition and clustering:** In the generated dataset, there is prior knowledge regarding the number of clusters to be chosen (e.g. $k = 5$). However, the Python code of the elbow method can also be applied to determine the most appropriate number of clusters for K-means. Thus, in the clustering step, the elbow method was applied to the dataset to validate the result and closely aligned with our assumed value of $k = 5$, reinforcing the empirical choice. The K-means algorithm receives DI_1 and DI_2 attributes and returns clusters with samples grouped between healthy and damaged. As a result of this step, one generates a clustered dataset saved by the algorithm into an Excel file named "*dataset.to_excel*", which is employed in the next steps. (Applied to Alg1, Alg2, and Alg3)

6 **Data splitting:** The new dataset generated by K-means (two or more attributes and a group cluster) is divided into training and testing sets for model construction and evaluation following a ratio, e.g. 80-20%, respectively. (Applied to Alg1, Alg2, and Alg3)

Damage detection:

7C **Classification:** Six ML-supervised classification algorithms are applied to the training data using the sci-kit-learn library. This pro-

cess allows the model to learn patterns and relationships within the dataset. (Applied to Alg1 and Alg3)

8C **Classification model evaluation metrics:** This step assesses the model's performance by testing it on the previously separated test dataset (as per step 6). During this stage, model hyperparameters can be fine-tuned to improve metrics such as Accuracy, Precision, Recall, and F1-score. Additionally, the confusion matrix is examined for insights. Cross-validation is employed to prevent overfitting and promote model generalisation on the training set, with 5-fold cross-validation being utilised in the current study. The confusion matrix graphic is generated and saved in a high-resolution PDF figure by the function *"fig.saving"*. (Applied to Alg1 and Alg3)

Damage and uncertainty quantification:

7R **Regression:** The supervised regression ML algorithm is implemented using the Scikit-learn library. The nine supervised regression machine learning algorithms are applied to this software version. The same dataset previously clustered by the K-means model (two attributes and a group cluster) is used as input in this process, addressing the regression problem, which returns the damage level values. (Applied to Alg1 and Alg2)

8R **Regression evaluation metrics:** Predicting future data trends using the trained model based on machine learning algorithms. Performance metrics are fundamental for supervised machine learning models, as they allow us to assess the quality of predictions. The aim is to evaluate the model's performance when dealing with new data, ensuring its effectiveness. In the case of regression models, performance is assessed using the evaluation metrics R^2, MAE, MSE, and RMSE (see Section 3.3.4. (Applied to Alg1 and Alg2)

9 **Final decision and interpretation:** The last step of the proposed algorithm offers information into the damage state through the clas-

sification algorithm and provides damage quantification via the regression algorithm. It also includes uncertainty quantification in damage estimation. These outcomes play a pivotal role in the interpretation and decision-making processes related to detecting damage in the structure analysed. (Applied to Alg1, Alg2, and Alg3)

3.2 Feature extraction and augmentation

Feature extraction involves transforming and reducing data dimensions through a multi-step fusion process to eliminate noise and highlight structural signatures from the original data. This process is often seen as essential in ensuring the reliability of SHM systems, as it significantly impacts the accuracy and effectiveness of detecting damage [61]. Obtaining enough experimental data for predictive applications and implementing data-driven monitoring remains a challenge in many fields[169]. Consequently, the scarcity of data can pose problems for ML algorithms performing on experimental system's effectiveness, overfitting, and limited exploitation of the feature space during model training [170]. Data augmentation methods can minimise such issues and provide larger datasets by generating artificial data to improve accuracy and robustness.

3.2.1 Data analysis and processing

The first step of any damage detection system is collecting and efficiently analysing and processing the sensor's data, which serves as input to identify changes in the monitored system. The system's effectiveness depends directly on the accuracy and reliability of the information provided by the sensors, emphasising the importance of the quality and relevance of the data collected [171]. Data can be classified into different types, depending on its organisation and structure [172]:

- Structured: Data with a well-defined structure follows a standardised order. Examples include names, dates, and addresses, among others.

- Unstructured: Data that does not have a predefined format or organisation. Examples include sensor data, emails, text documents, PDF files, audio, videos, images, and other similar types.

- Semi-structured: Data combining structured and unstructured elements, displaying certain organisational properties without a rigid schema. Examples include HTML files, XML, JSON documents and NoSQL databases.

- Metadata: Data that describes other data, providing information about characteristics such as authorship, file type, size, date and time of creation, or last modification. Examples include attributes such as author, file format, and document size.

This categorisation is useful for understanding the data's characteristics and determining appropriate collection, processing, and analysis approaches.

Algorithms are the core of data processing in damage detection models. They transform raw data into useful information for identifying and characterising damage [173]. The choice of algorithm depends on factors such as the type of data available (e.g. vibration, temperature, acoustics), the complexity of the system being monitored, operating conditions, and the requirements for accuracy and speed of detection. In addition, effective algorithm implementation must consider computational limitations and the conditions of the system's environment. Therefore, choosing and implementing algorithms to process the data and carry out identification is the central element of a damage detection system. Before selecting the interpretation algorithm, choosing between model-based or data-driven approaches is essential. This work will focus on data-driven models.

Data-driven modelling, driven by machine learning [174], has become essential for structural dynamics and SHM [175, 176]. These models are generally built based on collected input and output data [177]. A general process for a data-driven SHM is described below and illustrated in Figure 3.2.

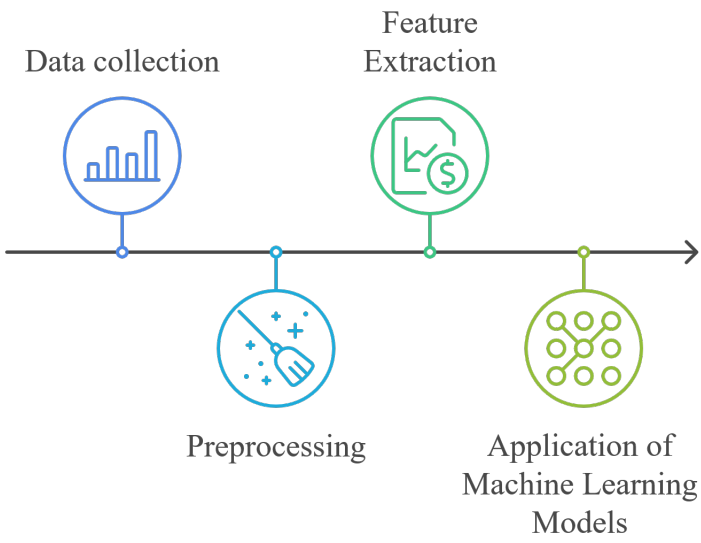


Figure 3.2: A general process for a data-driven SHM (Source: own study).

Firstly, the relevant data is collected from various sensors installed on the monitored structure. These sensors can include accelerometers, strain gauges, microphones, cameras, or other specialised devices. The aim is to collect information that can reflect the state of the structure, such as vibrations, stresses, temperatures or images. The data must vary during the phenomenon, show some causal relationships, and correlate with each other. To apply the techniques successfully, the variables monitored during the phenomenon that influence the prediction must be recorded and used as inputs in the model [178]. After collection, the data undergoes pre-processing (removal of noise, inconsistent data, invalid features, normalisation to standardise different measurement scales, feature dimension reduction, etc) to improve its quality and prepare it for analysis. Data pre-processing results in a cleaner experimental set with more relevant features and reduced feature space, facilitating the application of data-driven methods. The aim is to ensure that the data is consistent and usable for the following stages, increasing the accuracy and efficiency of the system.

Feature extraction involves identifying and isolating important charac-

teristics in the raw data indicative of the structure's state [179]. This process can be improved with appropriate pre-processing, statistical techniques (mean, standard deviation), analyses in the time and frequency domains (Fourier transform, wavelets) and identifying specific characteristics such as amplitudes, natural frequencies and vibrational modes. This process makes the data more representative, facilitating the model learning and improving its efficacy. The features must be specific, sensitive to structural integrity changes, and defined with a cause-effect relationship, allowing damage to be detected accurately [178]. The features are then categorised to identify patterns or anomalies that may indicate the presence of damage. In the final stage, conventional ML or DL models make predictions and diagnoses [180].

3.2.2 Frequency-domain features

Vibration-based damage detection techniques investigate the problem of locating and quantifying damage in a structure from changes in its dynamic characteristics [168]. Structure vibration signature has been used as a sensitive indicator of structural integrity and can be employed to monitor the procreation and propagation of damage. In some SHM methods, damage detection is performed by comparing the vibration signature in two states of the structure, one considered undamaged and the other damaged [143]. The identification of damage by vibration-based methods is based on the fact that the damage causes changes in the physical properties of a structure, such as mass, damping and stiffness, and can induce changes in the dynamic response, like the natural frequency, mode shape, and resonant frequency. Therefore, changes in dynamic characteristics can be used as damage indicators compared with the original response.

The damage index (DI) is formulated by comparing a reference signal, usually derived from the system considered undamaged, to the one provided by the system under the presence of discontinuing or damage [181]. Various DI approaches have been developed to extract signal features in different domains, aiming at identifying structural damage using an indicator that describes the damage. The DIs are associated with the estimation tech-

niques for damage quantification and reveal important information about the structural health condition. Therefore, the DI is normally presented in values between zero and unity, where the unit accuses no damage. A lower value up to zero indicates the presence of a crack and its severity within the analysis scenario. This work uses the DI as structure information for the training and testing data in the multiclass ML algorithms. The DIs can be used to classify the damaged and undamaged state from the loosening torque conditions of a bolted joint. Resonant frequencies vary according to the torque levels, enabling the DI to identify and quantify the torque loss. This work uses DIs to build an ML dataset from the experimental bolted beams-driven data.

The literature describes a range of DI developed over time. In this work, we used the Frequency Response Assurance Criteria (FRAC), Frequency Amplitude Assurance Criteria (FAAC), Global Shape Criteria (GSC), Global Amplitude Criteria (GAC), Average Integration GSC/GAC (AIGSC, AIGAC), and Monnier's Damage Index (DI). The FRAC is a damage index representing the correlation between tested frequency responses. It references FRF signals [182], where a unity indicates a strong correlation in case no damage is found. In contrast, the lowest correlation reaches zero, depending on the damage severity. The FRAC is defined by

$$FRAC_{ij} = \frac{\left\| H_{ij}^{dam}(\omega) \left(H_{ij}^{und}(\omega) \right)^* \right\|^2}{\left[H_{ij}^{und}(\omega) \left(H_{ij}^{und}(\omega) \right)^* \right] \left[H_{ij}^{dam}(\omega) \left(H_{ij}^{dam}(\omega) \right)^* \right]} \tag{3.1}$$

where (*) defines the complex conjugate operator, $(H_{ij}^{dam}(\omega))$ is the FRF vector on (j) for the damaged excited on (i) and $(H_{ij}^{und}(\omega))$ is the FRF vector for the undamaged, on the same aforementioned coordinates. Another damage indicator that uses the correlation function in the frequency domain is the frequency amplitude assurance criterion [183, 184], which measures differences in response amplitude. FAAC follows the same idea as FRAC and is denoted as

$$FAAC_{ij} = \frac{2 \left\| H_{ij}^{dam}(\omega) \left(H_{ij}^{und}(\omega) \right)^* \right\|}{\left[\left(H_{ij}^{und}(\omega) \right) \left(H_{ij}^{und}(\omega) \right)^* \right] + \left[\left(H_{ij}^{dam}(\omega) \right) \left(H_{ij}^{dam}(\omega) \right)^* \right]} \tag{3.2}$$

Global Shape Criterion (GSC) and Global Amplitude Criterion (GAC) proposed by Zang [183, 185, 186] were used as a damage index. The GSC and GAC must return values between zero and one for all frequencies similar to the FRAC and FAAC. The GSC and GAC DI are defined, respectively, by

$$GSC(\omega) = \frac{\|H^*_{und}(\omega)H_{dam}(\omega)\|^2}{\left[H^*_{und}(\omega)H_{und}(\omega)\right]\left[H^*_{dam}(\omega)H_{dam}(\omega)\right]} \tag{3.3}$$

and

$$GAC(\omega) = \frac{2\,\|H^*_{und}(\omega)H_{dam}(\omega)\|}{\left[H^*_{und}(\omega)H_{und}(\omega)\right] + \left[H^*_{dam}(\omega)H_{dam}(\omega)\right]} \tag{3.4}$$

In addition to GAC and GSC, Zang [186] calculates a single damage index value defined as the Mean Integration of the GAC and GSC functions, defined as

$$AIGSC(\omega) = \frac{1}{N}\sum_{i=1}^{N} GSC(\omega_i) \tag{3.5}$$

and

$$AIGAC(\omega) = \frac{1}{N}\sum_{i=1}^{N} GAC(\omega_i) \tag{3.6}$$

where N is the frequency band number. The AIGSC and AIGAC indicators are real constants between zero and unity, indicating total damage or undamaged structure. The Monnier Damage Index[187] and its modified version proposed by Banerjee [188, 189] are the normalised difference in module between two FRFs in different structural states. The Monnier DI return a single real value between zero and unity for a given frequency band of interest. However, values closer to zero represent minor damage (no damage or healthy indication), while values closer to unity represent greater damage.

$$DI = \frac{\sum_{i=1}^{n}\left\|H^{und}_{ij}(\omega)H^{dam}_{ij}(\omega)\right\|}{\sum_{i=1}^{n}\left\|H^{und}_{ij}(\omega)\right\|} \qquad n = 1, 2, 3...n. \tag{3.7}$$

$$DIs = \left\| 1 - \frac{\left(H^{dam}_{ij}(\omega)\right)^T * H^{dam}_{ij}(\omega)}{\left(H^{und}_{ij}(\omega)\right)^T * H^{und}_{ij}(\omega)} \right\| \tag{3.8}$$

where n is the captured frequency band of the spectrum and T indicates the transposition of the FRF vector

3.2.3 Time-domain features

The signals collected by sensors in monitoring and fault diagnosis systems are usually acquired in the time domain. However, due to their oscillation nature, it is difficult to identify changes caused by damages directly. However, time domain analysis methods are widely used to extract features from signals, as it is the first information measured from the structures, allowing differentiation between damaged and undamaged states[190, 191]. In the feature extraction stage, it is possible to identify damage-sensitive features using statistical analyses [192]. In addition, statistical features are easier to estimate directly from the data.

Vibration-based damage detection and structure localisation using time indicators are also used for SHM [193, 192]. This approach allows a temporal signal to be characterised representing it by a single value [194]. In other words, the raw signal data is compressed into a shorter version using features that describe it. In addition, different features in the time domain have different information in the vibration signal [195], providing information on the statistical and physical properties of the signal, such as variability, peaks, and energy. These aspects are essential for detecting faults and identifying irregular patterns. In this study, fourteen techniques are applied to extract features from the time-domain signal, $x(t)$, as shown in Table 3.1, including [196]:

Table 3.1: Time domain feature extraction methods.

Description	Formula
01. Maximum value of $x(t)$	$\max[x(t)]$
02. Minimum value of $x(t)$	$\min[x(t)]$
03. Amplitude range	$\max[x(t)] - \min[x(t)]$
04. Median value of $x(t)$	$Median = \begin{cases} x\left(\frac{n+1}{2}\right), & \text{if } n \text{ is odd} \\ \frac{x\left(\frac{n}{2}\right)+x\left(\frac{n}{2}+1\right)}{2}, & \text{if } n \text{ is even} \end{cases}$
05. Mean value of $x(t)$	$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
06. Variance of $x(t)$	$\nu = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$
07. Energy of the signal	$E_s(x) = \sum_{i=1}^{n} x_i^2$
08. Energy of the centred signal	$E_c = \sum_{i=1}^{n}(x_i - \overline{x})^2$
9. Skewness of $x(t)$	$Skew = [E((x - \overline{x})^3)]/\nu^{3/2}$
10. Kurtosis of $x(t)$	$Kurt = [E((x - \overline{x})^4)]/\nu^2$
11. Moment order (m_i)	$m_i = [E((x - \overline{x})^i)]/\nu^{i/2}, \quad (i = 5:10)$
12. Shannon Entropy	$H_S(x) = -\sum_{i=1}^{n} x_i^2 \log_2(x_i^2)$
13. Signal rate (τ)	$\tau = [\max(x_{1:n}) - \min(x_{1:n})]/\overline{x}$
14. Root mean square of $x(t)$	$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$

where x_i in Table 3.1 represents vibration signals, n is the total amount of sampling points and $E()$ represents the expected values. The amplitude variation among the features can be an issue due to different amplitude levels and small differences between operations thresholds. Such issues pose great challenges for the ML algorithm in finding a pattern and performing the classification further. To cope with this issue, we proposed a relative change damage index (RC_f) in the feature condition extraction and incorporated a normalisation in the feature expressed as

$$\Delta_f = [\max[feature] - feature] \tag{3.9a}$$

$$RC_f = \frac{\Delta_f}{\max[\Delta_f]} \tag{3.9b}$$

where Δ_f represents the difference between each element in the $feature$ vector and its maximum values, and RC_f is the feature's relative change, calculated by dividing Δ_f by this maximum value. This normalisation method ensures that the features are scaled between zero and one, preserving their essential characteristics while enabling consistent feature comparison.

3.2.4 Data argumentation

Data augmentation artificially expands datasets by introducing variations or using deep learning to generate new data points while preserving the core characteristics of the original data [197]. Virtual sensors are software-based tools that indirectly estimate process variables or unknown conditions by exploiting data from physical sensors combined with data fusion techniques [198]. In this work, the proposed virtual sensor performs data augmentation and fusion of the new synthetic data with the real one. Hence, our virtual sensor augments damage indices obtained from the raw measurements to increase dataset volume. Furthermore, the generated synthetic data is combined with the real data, and the sensor output is the augmented DI. In practice, the virtual sensor increases the volume of data guided by physical sensor input derived from the experimental vibration signals of the bolted being in different health conditions.

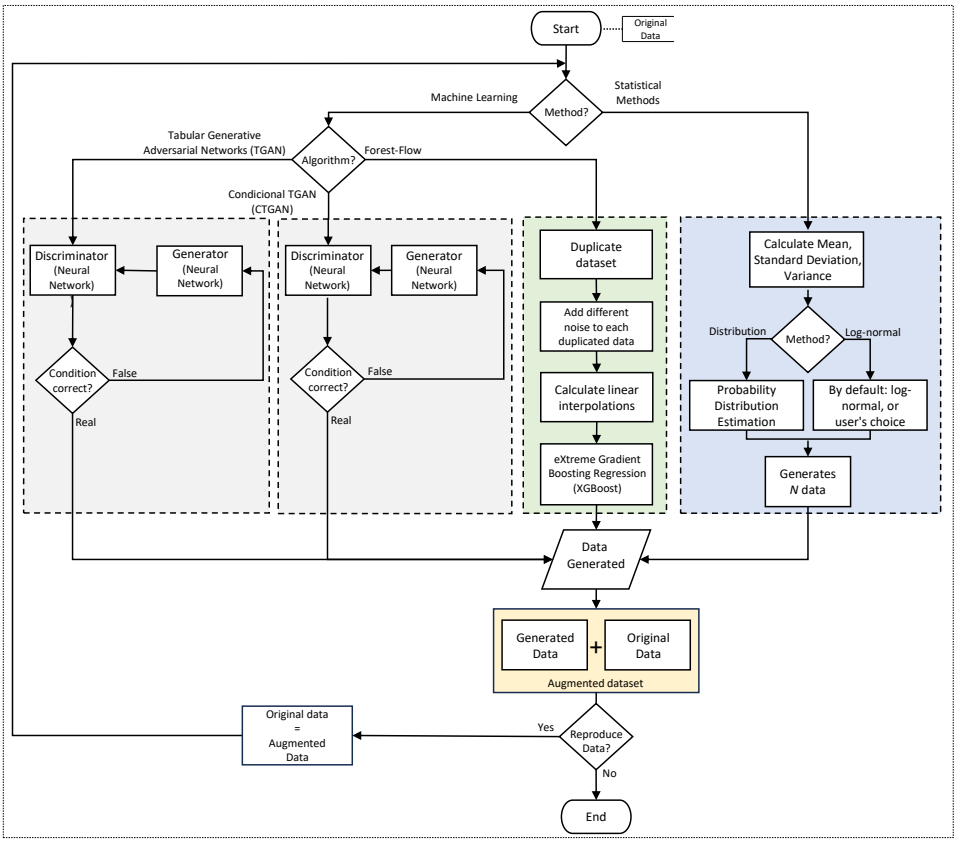


Figure 3.3: Virtual sensor flowchart for data fusion and augmentation (Source: own study).

This section details the virtual sensor algorithm for data augmentation. The architecture includes four methods based on statistical moments (Section 3.2.4), Tabular Generative Adversarial Networks (TGAN) [199] (Section 3.2.4), Forest Diffusion [197] (Section 3.2.4), and the combination of the multiples DIs presented in Section 3.2.4.

Figure 3.3 presents the virtual sensor flowchart proposed in this paper. The process begins with calculating DIs from experimental transmissibilities and then selecting a method that involves either a statistical, deep learning-based approach for data augmentation or both. Each pathway has distinct variations, which are detailed in the subsequent subsections. The augmented data is then combined with the original dataset for further reproduction or

as input to the next step of the condition assessment for torque loosening. Hence, the virtual sensor is designed to streamline the data augmentation process, enhancing both the performance and accuracy of machine learning algorithms, particularly in scenarios with limited data. The overall virtual sensor algorithmic workflow data fusion and augments are as follows

1. **Start of the process**: The process starts with the original data input provided by the user. In our case, the Damage indexes.

2. **Method**: The user can choose between 'Statistical Methods' or 'Machine Learning'.

3. **Statistical Methods**: In the statistical route, the algorithm calculates the mean, standard deviation and variance of the original(input) data. The next step is to determine or select a probabilistic distribution. The log-normal distribution is used by default, but the user can choose a different distribution, which may be more suitable depending on the nature of the data. Both ways end in generation N-samples, as defined by the user.

    a) **Log-Normal**: Using the log-normal distribution, new data samples are generated using Monte Carlo simulation (N-samples).

    b) **Distribution**: Based on the estimated or given probability density function, new samples are generated using Monte Carlo simulation.

4. **Machine Learning**: The machine Learning route is chosen, and the user can select among the deep learning TGAN, CTGAN (Conditional TGAN) or Forest Diffusion. Independent of the choice, those methods are based on the generative adversative techniques that reproduce the original data in an unsupervised approach.

    a) **TGAN**: is a method for modelling the distribution of tabular data and sampling lines from the distribution.

b) **CTGAN**: is specific to dealing with non-Gaussian and multimodal distributions, mode-specific normalisation and conditional training to handle unbalanced discrete columns.

c) **Forest Diffusion**: This model combines diffusion methods, such as Corresponding Flow Matching (CFM), with XGBoost, an augmented gradient tree method, to produce and attribute tabular data, both continuous and categorical.

5. **Data augmentation**: Regardless of the previous choice, the augmented data is combined with the original data to enhance the dataset that will be input in the ML algorithms.

6. **Reproducing data**: The user can decide whether to round another round of the argument process to increase the dataset further or to proceed to the end. In the case of another round, the augmented data is considered as 'original data', and the process starts again.

7. **End of the process**: The end of the process returns the new dataset, which combines the original and new synthetic data.

Statistical Data Generation

In this subsection, we want to generate synthetic data from an original dataset using a lognormal distribution. The lognormal distribution is a probability distribution where the logarithm of the variable follows a normal distribution. A key feature of the lognormal distribution is that it only allows positive values, making it particularly suitable for modelling phenomena restricted to non-negative outcomes [200]. For a lognormal random variable, the mean (μ) and standard deviation (σ) are determined by of the associated distribution as

$$f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right),$$
$$\mu = \ln\left(\frac{\overline{x}^2}{\sqrt{v + \overline{x}^2}}\right), \quad \sigma = \sqrt{\ln\left(1 + \frac{v}{\overline{x}^2}\right)} \tag{3.10}$$

where $x > 0$ is the value of the random variable, μ is the average of the variable's natural logarithm, and σ is the standard deviation of the natural logarithm of the variable. Based on this statistical information, sample generation is carried out according to the sample size defined by the user. The samples are summed to the original dataset depending on the data generation. For other distributions, the users can easily use it to declare the selected distribution.

Tabular generative adversarial networks

Generative adversarial networks (GANs) are a generative modelling technique based on neural networks, first introduced by Goodfellow et al. in 2014 [201]. GANs can generate new data that closely resembles the original data by learning the underlying probability distribution from the training set. In a GAN, the discriminator (D) attempts to distinguish between real data and synthetic data, while the generator (G) creates realistic synthetic data to deceive the discriminator [202]. A synthetic data generator based on GANs for tabular data, known as TGAN, addresses the scarcity of experimental data. The basic architecture of TGAN is illustrated in Fig. 3.4.

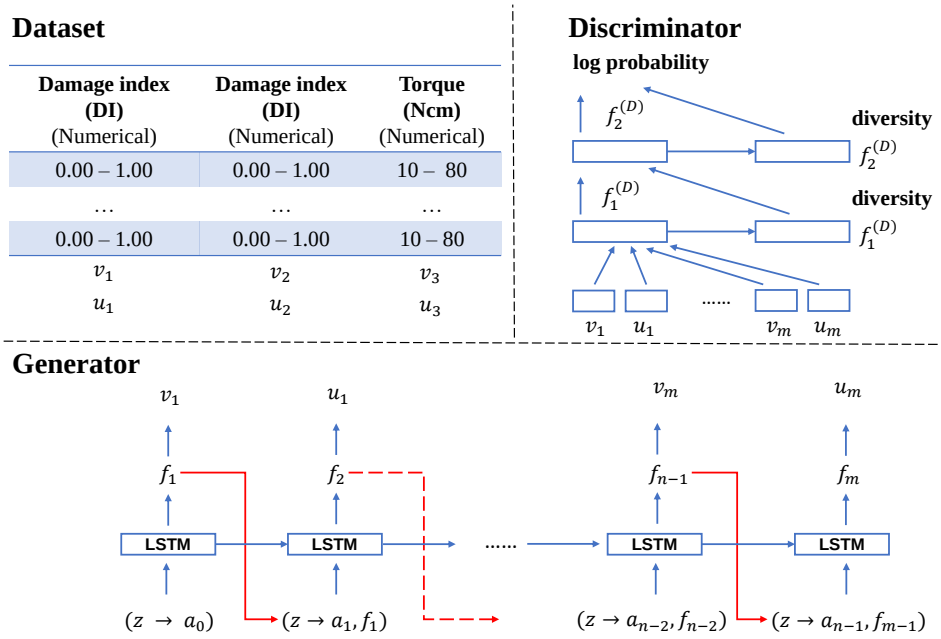


Figure 3.4: The structure of the TGAN model for generating tabular data (Source: own study).

The TGAN proposed by Xu and Veeramachaneni [199] increased the ability of the GAN-based model to generate tabular data that includes continuous, discrete, and categorical variables. TGAN employs a Long Short-Term Memory (LSTM) network to generate synthetic data column by column, with each column in the original dataset having its own dedicated LSTM cell. The role of each LSTM cell is to produce data for its respective column, and the process is sequential. After one column's data is generated, its LSTM cell's output is used as input for the next column's LSTM cell, and so on. This ensures that the dependencies between columns are preserved. A Multi-Layer Perceptron (MLP) is the discriminator, distinguishing between real and artificial data. The generator is optimised using the ADAM optimiser, which trains the network to generate artificial data that can fool the discriminator. During the optimisation process, the Kullback-Leibler (KL) divergence and the cluster vector are incorporated into the loss function, enhancing both the efficiency of the training process and the stability of the

model. The loss functions for the generator and discriminator are defined as follows

$$Loss_G = -\mathbb{E}_{z\sim\mathcal{N}(0,1)} \log D(G(z)) + \sum_{i=1}^{n_c} \mathrm{KL}(u'_i, u_i) + \sum_{i=1}^{n_d} \mathrm{KL}(d'_i, d_i), \quad (3.11)$$

$$Loss_D = -\mathbb{E}_{v_{1:n_c}, u_{1:n_c}, d_{1:n_d}\sim\mathbb{P}(T)} \log D(v_{1:n_c}, u_{1:n_c}, d_{1:n_d}) + \mathbb{E}_{z\sim\mathcal{N}(0,1)} \log D(G(z)) \quad (3.12)$$

where u'_i and d'_i are generated artificial data, u_i and d_i are original data, n_c and n_d are the continuous and discrete variables, respectively. The performance of the generator and discriminator is improved through iterative training, where both work to minimise their respective loss functions. This process continues until the discriminator can no longer distinguish between synthetic and real data.

Generating tabular data via Diffusion and XGBoost

Diffusion models are a recent generative method that estimates a scoring function and uses stochastic differential equations (SDEs) to generate samples. This process involves transforming real data into Gaussian noise through a direct stochastic process and learning to reverse the noise back into data using SDEs [197]. In contrast, Conditional Flow Matching (CFM) is a newer approach that estimates a vector field and employs ordinary differential equations to generate data. CFM works similarly to diffusion models, but deterministically, converting data to noise and noise to data [197]. Both methods have been successful in data generation tasks and typically use deep neural networks to estimate the scoring function or vector field, as neural networks are considered universal function approximations. In this study, we employ an approach to generate artificial tabular data using diffusion and Independent-Conditional Flow Matching [203] with XGBoost [204], as proposed by [197]. The method employs a Gradient-Boosted Tree (GBT) model instead of neural networks to estimate the vector field or

scoring function. This technique generates realistic tabular data, which can be trained on complete or incomplete data, covering continuous and categorical variables. The stages of the Forest Diffusion method are described below and illustrated in Figure 3.5, following the steps:

1. The first stage duplicates the original dataset, represented as x by n-times.

2. The second stage adds a different noise to each duplicated dataset. Each duplicate data receives a different Z noise vector, creating several noisy versions of the original data.

3. The third step calculates the linear interpolation between the duplicate dataset x and its corresponding noise Z for different times t. The interpolation is given by:

$$x(t) = tx + (1-t)Z, \qquad t \in \left\{0, \frac{1}{3}, \frac{2}{3}, 1\right\} \tag{3.13}$$

4. The final step involves regressing a Gradient-Boosted Tree (GBT) model for each noise level against the vector field. During this phase, the models are trained to map the interpolated data $x(t)$ to the corresponding vector field, minimising the difference between the model output and the expected data. The optimisation is performed to minimise the following loss function:

$$min_f \|f(x(t)) - (x - Z)\|_2^2 \tag{3.14}$$

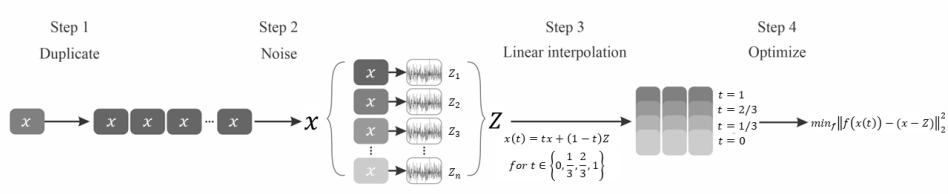


Figure 3.5: Steps of Forest-Flow method (Adapted from [197]).

Damage Index combined

The combination of the damage indices, detailed in Section 3.2.2, increases the volume of features from the original data. The advantage of this approach is that it is based only on the system's physical properties, thereby improving the robustness and accuracy of monitoring. As the machine learning algorithms input are often represented as vectors or matrices [205], our data features are organised in tabular and defined as a matrix of vectors, where each row of the table represents an instance or data point, the DI value, and each column represents a resource associated with a particular DIs attribute (DI_1 and DI_2). For a particular damage index, e.g. FRAC DI, the tubular matrix yields

$$\mathbf{DI} = \begin{bmatrix} DI_{1,11} & DI_{2,21} \\ DI_{1,21} & DI_{2,22} \\ \vdots & \vdots \\ DI_{1,k1} & DI_{2,2k} \end{bmatrix} \tag{3.15}$$

where k represents the table's number of rows containing the damage index values. The combined DIs in the tabular matrix consider more columns of damage indices derived from other theories, represented by

$$\mathbf{DI}_{multi} = \begin{bmatrix} DI_1^{\mathrm{FRAC}} & DI_1^{\mathrm{FAAC}} & \cdots & DI_1^{\mathrm{n}} \\ DI_2^{\mathrm{FRAC}} & DI_2^{\mathrm{FAAC}} & \cdots & DI_2^{\mathrm{n}} \\ \vdots & \vdots & & \vdots \\ DI_k^{\mathrm{FRAC}} & DI_k^{\mathrm{FAAC}} & \cdots & DI_k^{\mathrm{n}} \end{bmatrix} \tag{3.16}$$

where FAAC expresses the damage indices methods used considering in the feature extraction. There is no limitation of the feature added. , but in this case, for a good performance of the K-means, the columns must added in pairs of attributes (DI_1 and DI_2).

3.3 Machine learning algorithms

Machine learning is a technique within the field of artificial intelligence. It is defined as a process that automatically extracts patterns from

data [141], then uses the discovered patterns to make predictions about future data or to carry out other types of decision-making under uncertainty. This technique aims to determine a model that examines data to identify patterns or make predictions [206]. The model results from the training process, which involves a set of data and an algorithm that can be used to analyse and learn from that data. During training, the model adjusts its parameters to find the most relevant patterns in the data and becomes an accurate prediction based on new data. Machine learning has become increasingly popular in recent years due to its ability to analyse large amounts of data, identify patterns and make predictions or decisions based on that data [207]. With the rapid increase in data availability and the constant advancement of computing capabilities and programming methods, ML tools are increasingly used in various engineering areas [208].

The ML algorithms provide the tools needed to expand the capabilities of SHM systems [51]. It offers efficient solutions for building models or representations to map input patterns in measured sensor data to output targets for a damage assessment at different levels [209]. ML is part of this feature selection paradigm and statistical modelling for feature discrimination described in [55, 210]. In the literature, there are different types of machine learning methods [211]. The most commonly used approaches are Methods based on the amount of human supervision in the learning process (Supervised learning, Unsupervised learning, Semi-supervised learning and Reinforcement learning), Methods based on the ability to learn from incremental data samples (Batch learning or *offline* learning and *online* learning) and Methods based on their approach to generalising from data samples (Instance-based learning and Model-based learning).

The most commonly used categories of machine learning are supervised and unsupervised learning [212]. The main difference between the two categories lies in how the rules for classifying patterns are modelled. For example, when information on a structure's damaged and undamaged state is available, pattern recognition can be based on the supervised learning approach. When information on the damaged state is not available, un-

supervised learning is applied. Supervised learning best suits scenarios in which damaged and undamaged structure data is available for engineering structures [43].

3.3.1 Unsupervised learning

In unsupervised learning, the pattern is determined by an unknown class 'boundary', i.e. there is no prior information about the class to which the patterns belong, as there is no class associated with the pattern. All data is unlabelled [64]. This usually involves finding hidden structures in the data based solely on their characteristics and similar patterns. In this case, as there is no prior information about the desired result, i.e. the training data is missing, it can only be used to detect and possibly localise the damage [55]. Unsupervised learning is widely used to detect outliers due to the unavailability of the training data set [213]. Outliers, or anomalies, are patterns in data that do not follow a typical, well-defined behaviour [214, 215]. They can arise due to mechanical failures, changes in system behaviour, fraudulent behaviour, human errors, instrumental errors or simply natural deviations in populations [216].

There are many different types of unsupervised learning, including K-means Clustering, Hierarchical Clustering [217], GMM [158], Hidden Markov model [218] and PCA in the context of dimensionality reduction [219]. The K-Means clustering algorithm used in this work is an unsupervised ML in which data objects are distributed into a specified number of k clusters [220]. The k is a hyperparameter that specifies the number of clusters that should be created. It is a useful approach for clustering (labelling) or partitioning the data before feeding the labelled data as the output of a supervised ML algorithm. The aim is to find centroids that measure the cluster's centre point, such that the sum of the squared distances of each data sample to its nearest cluster centre is minimal. The nearest here is concerning the Euclidean norm (L2 norm). Thus, the objective function is

$$J = \sum_{i=1}^{n} min_k \left(\|x_i - \overline{x}_k\|^2 \right) \tag{3.17}$$

where x_i represents the ith instance in cluster k, and $\overline{x}_k$ denotes the mean of the samples or "centroid" of cluster k.

The K-Means algorithm is widely used due to its simplicity of implementation and low computational complexity. Still, one of the biggest problems of K-Means clustering algorithms is the initial definition of the number of clusters that must be used. When dealing with highly complex problems where the cluster count is hard to define, the "elbow" method can provide insights into the potential number of required clusters. Another disadvantage of K-means is that it is very sensitive to outlier points, which can distort the centroids and the clusters [221]. This work employs K-means for feature selection, clustering, and pattern recognition.

3.3.2 Supervised learning

In supervised learning, the input pattern is identified based on available information, i.e., the class is defined from a knowledge base of known patterns. This way, it assigns a class to this unknown object through a similarity measure with previously classified known objects. The known information forms a set of 'labelled' patterns. There is a very high demand for data in this case, as all possible damage situations must be available. The supervised learning approach is divided into a Regression problem and a classification problem [219].

Regression is a predictive learning problem that maps a data item to a real-value predictor variable. In other words, it has to predict a numerical characteristic of the data [222]. Regression predicts likely future or desirable outcomes from a set of labelled data. Regression models are widely used in various fields, including financial forecasting [223], price estimation [224], production [225], thermoelectric performance estimation [226], concrete compressive strength prediction [227] and many more. Classification is also a predictive learning problem. In this case, a label has a previous classification for a given example, and we want to predict which class an unclassified piece of data belongs to, i.e. it is the division of data into several known categories. Two types of classifiers are available considering their

output characteristics, binary and multiclass classification [228, 229].

- Binary Classification: This refers to classifying data with two class labels or groups/categories, for example, 'true and false' or 'positive and negative'. In binary classification problems, one class can be the undamaged state, while the other class can be the damaged state.
- Multi-class classification: This refers to classifying data with more than two class labels or groups/categories. In multiclass classification problems, objects are assigned to a category within a specified range without the distinction of normal or abnormal results, as found in binary classification.

Many classification algorithms are proposed in the machine learning literature, such as those described in [217, 230]. Herein, we highlight the ones we have used in our method.

3.3.3 ML algorithm used in damage detection and classification

The classifier algorithms employed to classify the damage are the supervised Naive Bayes, Decision Tree (DT), Random Forest (RF), K-Nearest Neighbours (KNN), Support Vector Machine (SVM), and extreme Gradient Boosting (XGBoost). To cluster the data and perform the SPR, the unsupervised K-means.

K-Nearest-Neighbour classifier

K-nearest neighbour is one of the simplest supervised learner methods [221, 53] and widely used for pattern recognition[231]. KNN can be used for classification and regression, where data with discrete labels usually uses classification and data with continuous labels regression. The classification is calculated from a simple majority vote of the nearest neighbours of each point: a query point is assigned the data class with more representatives within the nearest neighbours of the point. A metric between the points is used spaces[221].

The KNN algorithm, in its simplest version, only considers exactly one nearest neighbour, which is the closest training data point to the point we want to predict. The prediction is then simply the known output for this training point. Depending on the value of 'k_n', each sample is compared to find similarity or closeness with 'k_n' surrounding samples. For example, when $k_n = 5$, the individual samples compare with the nearest five samples; hence, the unknown sample is classified accordingly [221]. The optimal choice of 'k_n' value is highly data-dependent. In general, a larger suppresses the effects of noise but makes the classification boundaries less distinct.

Decision Tree and Random Forest

Decision tree supervised algorithm can target categorical variables such as the classification of a damaged or undamaged statement and continuous variables as regression to compare the signal with the healthy state of the system [53]. Learning a decision tree means learning the sequence of if/else questions that gets us to the true answer most quickly. A tree contains a root node representing the input feature(s) and the internal nodes with significant data information. Each node (a leaf or terminal node) represents a question containing the answer. The interactive process is repeated until the last node (leaf node) is reached such that the node becomes impure [221]. The data get into the form of binary features in our application, and a classification procedure is performed.

The random forest ML algorithm is an ensemble classifier that consists of many decision trees, and the class output is the node composed of individual trees. The RF has high prediction accuracy, robust stability, and good tolerance for noisy data. The law of large numbers does not overfit and has been used for structural damage detection. It has shown a better performance [232].

Support Vector Machine

Support Vector Machines are supervised machine learning techniques developed from the statistical learning theory that can be used for classifying

and regressing clustered data. In the case of linear classification, with two classes, let $\{(x_i, y_i), ..., (x_n, y_n)\}$, a training dataset with n observations, where x_i represents the set of input vectors and $y_i(+1, -1)$ is the class label of x_i, the hyperplane is a straight line that separates the two classes with a marginal distance (as seen in Fig. 3.6). The purpose of an SVM is to construct a hyperplane using a margin, defined as the distance between the hyperplane and the nearest points that lie along the marginal line termed as support vectors [233].

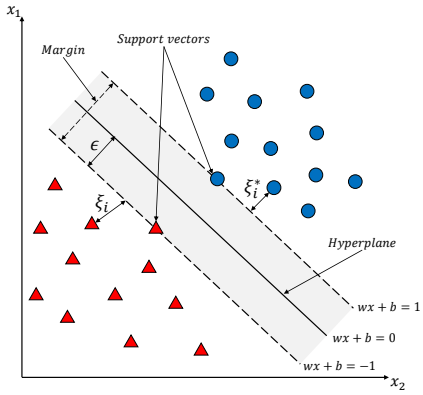


Figure 3.6: SVM algorithm operation (Source: own study).

One can define the hyperplane by Eq. (3.18), where we have the dot product between x and w added to the term b:

$$D(x) = w^T.x + b = c \quad for \quad -1 < c < 1 \tag{3.18}$$

where x represents the points within the hyperplane, w is the weights that determine the orientation of the hyperplane, and b is the bias or displacement of the hyperplane. When $c = 0$, the separating hyperplane is in the middle of the two hyperplanes with $c = 1$ and -1. An SVM aims to maximise the data separation margin by minimising $||w||$. This optimisation problem can be obtained as the quadratic programming problem given by

$$min \quad \frac{||w||^2}{2} \quad s.t \quad y_i(w^T.x_i + b) \geq 1 \quad for \quad i = 1, 2, ..., n \tag{3.19}$$

where $||w||$ is the Euclidean norm. The SVM algorithm encompasses linear and nonlinear classification and linear and nonlinear regression. The main idea of the algorithm consists of fitting as many instances as possible a "tube" while limiting margin violations. Therefore, SVR wants to find a hyperplane that minimises the distance from all data to this hyperplane. The width of the "tube" is controlled by a hyperparameter, which has an error "insensitive" area, defined by ϵ, as illustrated by Figure 3.6. The larger the ϵ, the larger the diameter of this tube, and the less sensitive the model is in predicting points within it. In contrast, the smaller ϵ, the smaller the diameter of the tube, the greater the chances of points being on the edges of the tube, making the model more robust. The samples that fall into the ϵ-margin do not incur any loss. Points outside the tube are examined and considered concerning the ϵ-insensitive region. Compared to a previously defined error called slack variables (ξ). This approach is similar to the "soft margin" concept in SVM classification because the slack variables allow regression errors to exist up to the value of ξ and ξ_i^*, yet still satisfy the required conditions. Including slack variables leads to the objective function given by Eq. (3.27).

$$\begin{aligned} Minimize: \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ Constraints: \quad & y_i - w^T.x_i - b \leq \epsilon + \xi_i \\ & w^T.x_i + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, ..., n \end{aligned} \tag{3.20}$$

Naïve Bayes

Naïve Bayes classification is a probabilistic classification method based on Bayes theorem with the assumption of independence between features, considered a simple technique for constructing classifiers with models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There are three classes in sk-learn, the Gaussian-NB, Multinomial-NB, and Bernoulli-NB.

The first assumes a Gaussian distribution, the second is for discrete occurrence counters, and the third is for discrete boolean attributes [234]. Naive Bayes classifiers are highly scalable, requiring several linear parameters in the number of variables in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression. In other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods. An advantage of naive Bayes is to train a model with few samples [235].

Extreme Gradient Boosting (XGBoost)

XGBoost (short for Extreme Gradient Boosting) is an efficient implementation of Gradient Boosting Machines (GBM), developed by Tianqi Chen [204], widely recognised for its superior performance in supervised learning. This versatile algorithm is also considered an ensemble tree technique that can be used for regression and classification tasks. XGBoost follows the concept of weak-learner, where each predictor could be improved by sequentially training new trees to the model [236]. In other words, the XGBoost makes predictions by creating numerous smaller decision trees, also known as subtrees. Each subtree makes predictions for the data, combining their predictions to form the final prediction for the given input. This ensemble approach helps improve the accuracy and generalisation ability of the predictive model. The process involves iteratively training these subtrees to correct the errors made by the previous subtrees, gradually refining the overall prediction as more trees are added.

Another feature related to XGBoost is that it uses $L1$ and $L2$ regularisation, which helps with model generalisation and reduction of overfitting. It uses an optimisation strategy that produces better weights as it calculates the weights of the component models. It also uses slightly less tiny component models.

Evaluation metrics for classification models

The performance of the classification models was evaluated using key metrics such as Accuracy in Eq. 3.21, Precision in Eq. 3.22, Recall in Eq. 3.23, and F1-score in Eq. 3.24, which are based on True Positive (TP), True Negative (TN), False sitive (FP) and False Negative (FN) samples [237, 238]. These metrics are crucial at various stages of the modelling process, such as model type selection, final evaluation and ongoing performance monitoring [239].

$$\text{Accuracy}(y, \hat{y}) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{3.21}$$

$$\text{Precision}(y, \hat{y}) = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.22}$$

$$\text{Recall}(y, \hat{y}) = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.23}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.24}$$

where y_i and $\hat{y}$ represent the true and predicted label, respectively.

Actual	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative TN
		Positive	**Negative**
		Predicted	

Figure 3.7: Confusion Matrix (Source: own study).

The confusion matrix [240] is also vastly employed to verify the data classification, which provides the correct configurations of the classified data. It compares actual values with predicted values, categorizing them into two labels: “Positive” and “Negative” [241]. The matrix’s main diagonal values show how many correct model predictions are for each class. To illustrate, consider a 2×2 confusion matrix in Figure 3.7, where:

- True Positive (TP): The number of instances where the ML model correctly classifies a sample as belonging to the “Positive” class.
- False Negative (FN): The number of instances where the ML model incorrectly classifies a sample as belonging to the “Negative” class when it belongs to the “Positive” class.
- False Positive (FP): The number of instances where the ML model incorrectly classifies a sample as belonging to the “Positive” class when it belongs to the “Negative” class.
- True Negative (TN): The number of instances where the ML model correctly classifies a sample as belonging to the “Negative” class.

Cross Validation

When developing a machine learning model, it is essential to check that the model is well-adjusted. A simple approach is to split the data into two sets: training and test [242]. The model is trained with the training set to fit the model and is evaluated in terms of accuracy using the test set [243]. The data split usually follows proportions, such as 80% for training and 20% for testing, and can be adjusted according to the problem. However, when evaluating the model only once, the question of whether the good performance observed was due to random factors may arise. To obtain greater confidence in the quality of the model, it is necessary to evaluate it several times using methods such as cross-validation, which allows for a more consistent and robust analysis of the model’s performance.

Cross-validation is a widely used statistical method for evaluating performance, comparing machine learning algorithm models on unseen data samples and preventing overfitting [244, 38, 245]. This approach provides a more accurate estimate of the generalisation error, especially in small datasets. By evaluating the model on multiple validation subsets, cross-validation offers a more realistic view of its performance, helping to identify and mitigate overfitting, which occurs when the model provides accurate

predictions for the training data but not for the new data in the test set [246]. It also provides a reliable estimate of the model's expected performance on new and unseen data. The method splits the data into two segments: one for training the model and the other for validation (testing) [247]. The most common cross-validation approach is k-fold cross-validation [248].

In k-fold cross-validation, the dataset is randomly divided into k_f subsets called folds (where k_f is defined in advance), with an approximately equal sample in each subset. The model is trained and evaluated k_f times, using a different fold as a test set in each iteration. In the first iteration, the first fold serves as the test set, while the remaining $k_f - 1$ folds are used for training. The process is then repeated with the second fold as the test set, and so on, until each fold has been used once for validation, ensuring a comprehensive evaluation of the model. Each iteration generates an evaluation metric, such as accuracy, to monitor the performance of each learning algorithm. These metrics assess the model's ability to generalise to new and unseen data. Cross-validation is a fundamental tool in machine learning for evaluating model performance, preventing overfitting and selecting the most suitable model. Systematically dividing the data into training and validation sets and repeating this process several times provides a more robust and reliable estimate of the model's generalisation ability.

3.3.4 ML algorithms used to damage quantification

For damage quantification, we employ nine regression algorithms: linear regression, Lasso, KNR, DTR, GBR, SVR-linear, SVR-RBF, SVR-Poly, and MLP. This approach exclusively relies on experimental data from the monitored system, eliminating the need for numerical models. The proposed method significantly improves damage level quantification accuracy, enhances model performance in fault-level predictions, and supports decision-making.

Linear Regression

Linear Regression fits a linear model to the dataset by adjusting a set of parameters to minimise the sum of squared residuals of the model. It is considered a straightforward and commonly used statistical regression method for predictive analysis in machine learning [249]. The Equation 3.25 for linear regression is as follows

$$y = \theta x + \alpha \tag{3.25}$$

where θ is the slope of the line, α is the intercept, x is the independent variable (input feature), and y is the dependent variable (output feature).

LASSO Regression

Lasso regression (Least Absolute Shrinkage and Selection Operator) [250] is a regression technique that combines linear regression with L1 regularisation. The main aim of Lasso is to improve the generalisation capacity of the model by performing both regularisation and variable selection. L1 regularisation adds a penalty to the sum of the absolute values of the parameter coefficients, which reduces the coefficients of less significant characteristics to zero. This allows Lasso to automatically select the most important features, eliminating irrelevant ones and reducing the dimensionality of the model. Lasso's objective is to minimise the loss function:

$$\hat{\beta}_{L_1,\lambda} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j Y_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{3.26}$$

where y and Y_{ij} are the response and predictor variables, respectively; β_0 and β_j are the coefficients to be estimated; λ is the regularization parameter that controls the strength of the $L1$ penalty; p is the number of predictors (features); n is the number of observations. This technique is particularly effective for dealing with high-dimensional data [251], where many variables may be insignificant. Lasso helps solve multicollinearity problems by better distributing the coefficients [252]. Lasso regression is ideal for predictive

problems due to its ability to automatically select variables, simplify models and increase the accuracy of predictions.

Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a supervised learning algorithm that learns a function by training on a dataset. The MLP Regressor class implements a multilayer that trains using backpropagation without an activation function in the output layer, which can also be seen as using the identity function as the activation function. Therefore, it uses the mean squared error as the loss function, and the output is a set of continuous values. MLP Regressor also supports multi-output Regression, where a sample can have more than one target [249].

Support Vector Regression (SVR)

Support Vector Regression (SVR) is a supervised learning algorithm used for the regression of linear and nonlinear tasks. SVR can be applied to nonlinear problems using kernel functions such as Polynomial and Gaussian radial basis function (RBF) [253], which project the sample space into a higher-dimensional space where the data become linearly separable. It works by finding a hyperplane that minimises the distance from all data to this hyperplane. The width of the "tube" is controlled by a hyperparameter, which has an error "insensitive" area, defined by ϵ. The samples that fall into the *epsilon* margin do not incur any loss. Points outside the tube are examined and considered concerning the ϵ-insensitive region. Compared to a previously defined error called slack variables (ξ). Including slack variables leads to the objective function given by

$$\begin{aligned} Minimize: \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ Constraints: \quad & y_i - w^T.x_i - b \leq \epsilon + \xi_i \\ & w^T.x_i + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, ..., n \end{aligned} \tag{3.27}$$

where, w is the weight vector, b is the bias term, C is the regularisation parameter, ϵ is the epsilon-insensitive loss parameter, x_i are the input features, y_i are the target values, and ξ_i and ξ_i^* are slack variables that allow for errors.

Decision Tree Regressor (DTR)

Introduced by Breiman et al. [254], the Decision Tree algorithm transforms data into a tree structure where each internal node represents an attribute, and each leaf represents a class label. Used for both classification and regression, these trees categorise discrete and continuous data or predict numerical values. The common loss function is the squared error, which needs to be differentiable for regression problems [255, 256]. Inspired by real trees, they consist of a root node representing the input feature(s) connected to internal nodes that end in leaves, making them effective for segmenting complex data. The interactive process is repeated until the final leaf node is reached, and then the node becomes impure. The average response value of the observations in each leaf is used as the final prediction [257]. However, as the size of the data increases, the branches proliferate, increasing the processing time. Modifications to the algorithms help reduce the leaf count to mitigate this difficulty.

Gradiente Boosting Regressor (GBR)

Gradient Boosting (GB) is a popular ensemble method in the machine learning community, proposed by Friedman [258]. This technique combines multiple decision trees to create robust and effective models for classification and regression tasks [259]. GB repeatedly adds decision trees so that each new tree corrects the errors of the previous tree. Unlike other methods, GB does not adjust the weights of the training examples. Instead, each predictor is trained using the residual errors of the previous model as labels. The decision trees used in GB are generally shallow, with depths ranging from one to five, which makes the model lighter and the predictions faster. These shallow trees, called weak learners, improve the model's performance

as more trees are added [256]. GB effectively categorises data into discrete classes and predicts numerical values in regression tasks. It uses a stepwise additive model, where only one new weak student is added at a time, while the previous ones remain unchanged. This is similar to gradient descent, where the model is fitted by minimising the gradient of the loss function. Equation 3.28 presents the Gradient Boosting function for regression problems, where the gradient of the loss function is minimized.

$$F(x) = \sum_{i=0}^{n} \delta_i h(x; \gamma) \tag{3.28}$$

where $h(x; \gamma)$ is a parameterised function of the input variables x, characterised by the parameters γ, and δ_i is the expansion coefficient.

K-Neighbors Regressor (KNR)

The K-Nearest Neighbours algorithm is a simple, non-parametric method used for classification and regression [260]. In the context of regression, KNN makes predictions by identifying the k_n data points closest to a given input and calculating the average of their target values for numerical regression. For classification, it selects the majority class among the nearest neighbours. The choice of parameter k_n is crucial: smaller values of k_n result in more flexible and less biased models, while larger values produce smoother and more robust models. KNN is called a 'Lazy Learner' because it does not go through a traditional model learning phase, memorising the entire training dataset [261]. Although effective at capturing local patterns, KNN faces high-dimensional difficulties and depends heavily on the distance metric chosen (Euclidean or Manhattan) [262]. The choice of k_n and the distance metric must consider the data's characteristics and the problem in question.

Evaluation metrics for regression models

The Assessment of regression models is measured using common metrics like the Coefficient of Determination (R^2), Mean Absolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE). These

metrics are widely used to compare predicted values to actual ones and show how well the model performs [263]. The R^2, or the coefficient of determination (Eq. 4.22a), measure the variation in the actual values relative to the model's prediction. It ranges from 0 to 1 (or from 0% to 100%) and indicates how well the model explains the variability of the response variable [239, 264].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{3.29}$$

where y_i is target value, $\hat{y}_i$ is model's prediction, and $\bar{y}_i$ is the average of all the target values.

The Mean Absolute Error (MAE) is the mean of the absolute differences between the actual and predicted values, calculated by Eq. 4.22b [264].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\|y_i - \hat{y}_i\| \tag{3.30}$$

The Mean Squared Error (MSE) is the average of the squared differences between the predicted values, as calculated by Eq. 4.22c [264].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.31}$$

The Root Mean Squared Error (RMSE) (Eq. 4.22d) is the square root of the MSE. The MAE, MSE, and RMSE values range from 0 to ∞, with lower values indicating better performance of the regression model [264].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.32}$$

3.3.5 Uncertainty quantification

Despite advances in machine learning, which have increased the accuracy of predictions in various fields, especially engineering, the relationship between the inputs (x) and the target variable (y) is still subject to uncertainty [265]. This uncertainty represents a lack of confidence in the model's

predictions [266]. It can stem from the limitations of the models themselves and the natural unpredictability of the phenomena analysed, even with abundant data and available resources [267].

Machine learning models must provide more than a result to ensure safer and more informed decision-making. They need to show, as accurately as possible, the confidence level around their predictions before they are used in practice. This means presenting the results with clear information about the associated uncertainty and whether it is low enough to make the result reliable. If the uncertainty is high, the algorithm can request additional data or human intervention to assist in decision-making. The uncertainty that affects machine learning models can be categorised into two main types [268, 269, 270]: Aleatoric uncertainty (data uncertainty) and Epistemic uncertainty (parameter/model uncertainty).

- **Epistemic uncertainty (parameter/model uncertainty):** This uncertainty arises from incomplete or inadequate knowledge about the system [266]. A lack of training data can cause this, poor quality information or simplifications and assumptions made during the modelling process [271]. This uncertainty is reducible, i.e. it can be minimised by increasing the dataset and gaining a more detailed understanding of the model's structure and constraints [265]. However, this approach does not affect random uncertainty [267]. In the context of ML, epistemic uncertainty is particularly relevant and can be classified into two categories [272]:

    1. **Model shape uncertainty:** This arises from simplifications or architectural choices, such as activation functions in neural networks or kernel function shapes in GPR models.
    2. **Parameter uncertainty:** Related to the calibration and training processes, caused by insufficient training data, bias in the data or difficulties in the optimal adjustment of the parameters by the algorithms.

Epistemic uncertainty is higher in regions with little training data and lower in areas with higher data density, highlighting the importance of enriching datasets and improving training quality to reduce the impact of these limitations [265].

- **Aleatoric uncertainty (data uncertainty):** This type of uncertainty is associated with the inherent noise or stochastic variability of the process that generates the data [271]. Unlike epistemic uncertainty, it is not related to the model but rather to the nature of the data itself and is therefore irreducible, even with the addition of more data for training [266, 265].

  This uncertainty stems from the natural variability of physical systems, such as noise in sensor measurements, variability between samples or the dispersion of responses in replicated experiments. In machine learning problems, it reflects the stochastic nature of inputs, outputs or the dependencies between them and is often modelled in the likelihood function of probabilistic models [272]. Therefore, in the context of ML, random uncertainty arises from the intrinsic variability of the data, where the same vector of characteristics x can be related to different labels y [267]. Examples include variability in classes in classification problems and in outputs in regression problems. As an intrinsic property of data, this uncertainty represents a limit on the accuracy that ML models can achieve [272].

Uncertainty quantification (UQ) is crucial in machine learning, especially in critical scenarios where incorrect decisions can have severe consequences. UQ ensures more reliable results by providing confidence intervals for predictions, recognising the probabilistic nature of the results rather than just seeking greater accuracy [267]. One advantage of UQ is that it helps users set confidence limits on model predictions, which can sometimes become dangerous. In this sense, UQ extends classical statistical analysis to include the uncertainty generated by noise, incompleteness and scarcity of data, offering robust support for risk management [272].

In fault diagnosis, UQ is of paramount importance as it provides a measure of the confidence associated with the results, which is essential for making informed decisions in critical systems. It allows engineers and operators to interpret diagnoses and understand the degree of reliability of this information. Quantifying the uncertainty associated with the process is relevant when machine learning techniques are evaluated by metrics such as accuracy, precision, recall, F1-score, and the confusion matrix, which consider false positives and false negatives and can have significant consequences. These include unnecessary maintenance actions or failure to identify critical problems [273].

Employing UQ in the final SHM process increases the reliability and robustness of ML models. UQ is interested in providing model security and transforming ML solutions into critical decision problems that bring greater gains and less exposure to the risks arising from algorithm failures or limitations. Although there is no way of achieving absolute certainty, it is essential to understand, quantify, and, whenever possible, reduce this uncertainty so that the chances of error or deviations from the model's projections can be accurately estimated. In this study, uncertainty in projections is investigated using the Mean (Eq. 3.33), Standard Deviation (Eq. 3.34) and the application of the Probability Density Function (PDF) (Eq. 3.35).

$$\text{Mean } (\overline{x}) = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3.33}$$

where x represents the values of the variable, $\sum$ indicates "the sum of," and n denotes the total number of observations.

$$\text{Standard deviation} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \tag{3.34}$$

The PDF is a function defined in the sample space S, where $S \subseteq \mathbb{R}$, of a continuous random variable X. It can be used to determine the probability of the variable X taking on values within a certain interval. The PDF of a continuous variable X is a function $f(x)$ such that [274]:

$$P[s \leq X \leq t] = \int_{s}^{t} f(x)\, dx \tag{3.35}$$

where $P(s \leq X \leq t)$ represents the probability of the variable X taking on values in the interval $[s, t]$, and $f(x)$ is the PDF. Here, s and t are real numbers. The PDF must fulfil the following conditions:

1. For all values x in the sample space, $f(x)$ is a nonnegative function,

$$f(x) \geq 0 \quad \text{for all } x \in \mathbb{R},$$

2. The integral of $f(x)$ over the entire sample space is equal to 1. This guarantees that the sum of the probabilities is unitary:

$$\int_{-\infty}^{\infty} f(x)\, dx = 1.$$

3.3.6 Chapter final remarks

This chapter details the methodological framework employed in this study, which includes data processing, feature extraction, machine learning strategies, and uncertainty quantification. The proposed approach integrates frequency- and time-domain analyses with data augmentation techniques to enhance the robustness of feature extraction. These steps ensure the input data is well-structured and representative of real-world SHM conditions.

The selection of machine learning algorithms, including supervised and unsupervised learning techniques, was motivated by their ability to detect, classify, and quantify structural damage based on vibration spectral data. Using both traditional and advanced regression models for damage quantification reinforces the reliability of the proposed methodology. Furthermore, uncertainty quantification techniques were incorporated to address variability in real-world conditions and improve the interpretability of results. While this methodology provides the basis for the study, challenges such as data imbalance, model interpretability, and computational costs were carefully considered. The subsequent chapter presents the results of applying these methods, offering insights into their effectiveness and limitations in real-case scenarios.

4 Data-driven machine learning structural health monitoring: Experimental case studies

A wind turbine system comprises subsystems connected mainly by bolts or weddings. This section presents case studies applying the SHM-ML process proposed in this work. The first study (Section 4.1) aims to monitor and evaluate three component faults during the real operation of the Aventa wind turbine. In this case, the raw data employed in the monitoring is the accelerance temporal spectrum, and the Al 3 is used. Bolted connections, presented in many parts of the turbine, are evaluated using a laboratory experimental setup. The second case(Section 4.2) aims to identify and classify torque loosening in structure bolted joints using raw spectral signals in the frequency domain from experimental tests. The approach combined supervised and unsupervised techniques and employed the Al 1, using a damage index calculated from the frequency response of the joint system as input data. The third case (Section 4.3) integrates regression algorithms with data augmentation techniques to more accurately estimate torque loosening using raw vibration spectra in a bolted structure. These studies demonstrate the efficiency of the developed SHM-ML process, which contributes to fault detection and improves the reliability of complex mechanical structures and systems.

4.1 Case I: Failure classification in wind turbine components during operation

Despite recent advances, challenges persist in adapting SHM techniques to complex operational and environmental conditions and improving detection accuracy and reliability. While many studies employ supervised and unsupervised learning techniques to enhance anomaly detection or optimise classifiers for specific turbine components, integrating these techniques into a unique SHM framework remains a significant research challenge. In this case study, we used the unsupervised clustering technique k-means to clas-

sify and group data into homogeneous clusters, enabling pattern identification without predefined labels. Algorithm 3, described in Section 3.1, is used to classify the four real operational conditions of the Aventa wind turbine. The proposed model consists of receiving the data, processing, feature extraction, feature selection and normalisation, unsupervised classification and clustering, data splitting, supervised classification, and model evaluation.

4.1.1 Wind turbine experimental benchmark

The wind turbine dataset utilised in this study is from the Aventa AV-7 model, manufactured by Aventa AG and commissioned by the ETH Zurich Department of Structural Health Monitoring. It is located in Taggenberg, Switzerland, at coordinates 47°31'12.2"N 8°40'55.7"E [25]. This 6.7 kW-rated power turbine operates via a belt-driven generator and a frequency converter with a variable-speed drive. It initiates power generation at wind speeds of 2 m/s, with a cut-off speed of 14 m/s. The turbine features a 12.8-meter rotor diameter comprising three blades and is mounted at an 18-meter hub height. The maximum rotational speed reaches 63 rpm. Structurally, the tower is composed of tubular steel reinforced with concrete, supported by a concrete foundation, while the blades are constructed from fibreglass with a tubular steel main spar. Turbine control is achieved through a variable-speed, variable-pitch mechanism. The instrumentation on the tower and nacelle included 11 accelerometers strategically positioned along the tower length, the nacelle main frame, the main bearing, and the generator. Additionally, two full-bridge strain gauges are mounted at the tower base to measure fore-aft and side-to-side strain, which can be converted into bending moments. All acceleration and strain data are sampled at 200 Hz. Environmental data, including temperature and humidity, are recorded at the tower base with a sampling rate of 1 Hz. Furthermore, operational performance data (SCADA), which encompasses wind speed, nacelle yaw orientation, rotor RPM, power output, and turbine status, is collected and sampled at 10 Hz. The model's data preprocessing steps consist of se-

lecting the sensors to be included in the monitoring process by analysing their sensibility to failure events, extracting features from the temporal signal, normalising the features, and preparing the dataset.

4.1.2 Signal analysis and sensor selection

The data preprocessing starts with analysing and selecting the sensor for operational and failure identification. The pre-established failures are the rotor icing event (RI), the flexible coupling of the linear drive of the collective pitch system (FC), and the aerodynamic imbalance on one blade (AI). From our previous experience using vibration-based damage detection, we found that the sensors most sensitive to failure are the ones allocated close to the anomaly spot. Considering the number of sensors and data to process that influence time and computational cost, our choice is to work only with the sensors installed in the nacelle because they are the most likely to capture changes in the signal due to the failures considered in this study (RI, FC and AI) and not be masked by other system components, e.g., tower our boundary condition.

The selected accelerometers GEN_ACC (orange mark), NMF_ACC (blue mark), and MSB_ACC (yellow mark) capture signals for each event, including normal operation, as shown in Fig. 4.1. The Aventa dataset provides three-axis acceleration signals, with the X-axis (side-to-side turbine motion) and Y-axis (fore-aft turbine motion). Figure 4.1 includes a schematic representation of the sensor locations in the nacelle. In the acceleration graphs, black lines represent normal turbine operation, while coloured lines indicate different failure conditions. To ensure consistency, signals for both operational states were selected on the same day, minimising the impact of varying environmental conditions. The top row of graphs shows the x-axis response of the accelerometer installed on the generator (GEN). From left to right, it compares normal operation (NO) with FC, RI, and AI failures, respectively. The middle row presents the X-axis response of the accelerometer on the main shaft bearing (MSB), again comparing NO with FC, RI, and AI failures from left to right. Finally, the bottom row displays the X-axis

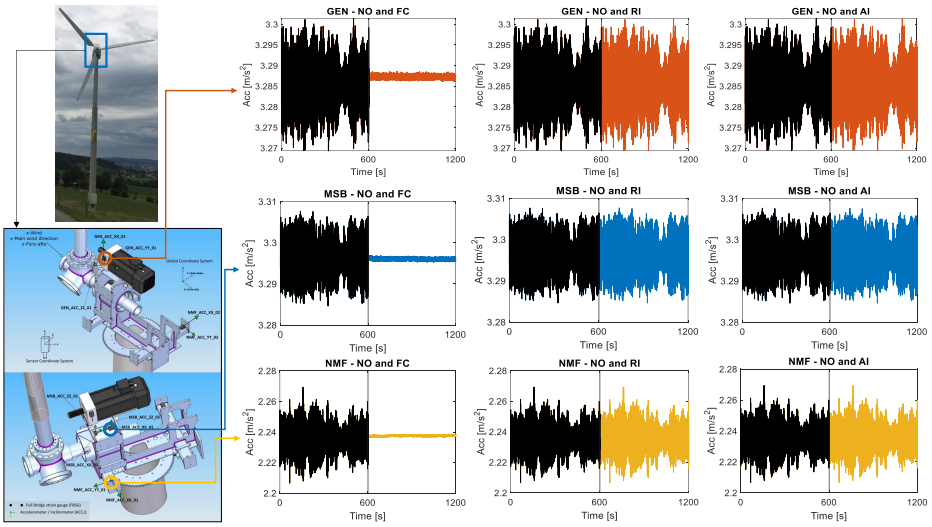


Figure 4.1: Schematic representation of the turbine and sensors position in the nacelle. Temporal section comparison of the turbine's normal operation and operation with failures: (TOP) sensor GEN NO-FC, NO-RI, and NO-AI placed from left to right, respectively. (Middle) sensor MSB NO-FC, NO-RI, and NO-AI; and (Bottom) sensor NMF NO-FC, NO-RI, and NO-AI.

response of the sensor on the nacelle main frame (NMF), with comparisons between NO and FC, RI, and AI failures. The sensors' X- and Y-axis signals were analysed. The curves followed a consistent pattern, with the FC failure causing a significant drop in amplitude compared to normal operation. In contrast, the other failure modes showed only minor differences in signal amplitude. These signal variations can fluctuate daily due to changing weather conditions.

From this initial analysis, changes in the turbine's dynamic behaviour appear to correlate with specific events. For subsequent procedures, only the generator sensor will be used, as the generator is responsible for converting the mechanical energy from the rotor blades into electrical energy. The generator's performance is critical to the wind turbine's energy production and reliability. Monitoring the generator's vibration response can provide early signs of potential turbine malfunctions.

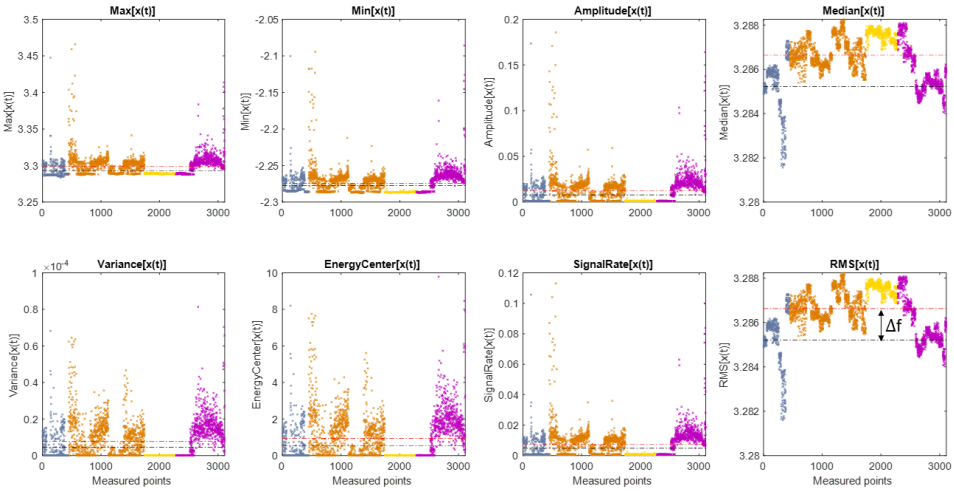
4.1.3 Feature extraction and normalisation

The spectral data comprise four distinct classes: NO, FC, RI, and AI, where features are derived from the time-domain responses of the generator accelerometer's raw signal, shown in Figure 4.1(TOP), along the x- and y-axis. Since the signal sensitivity displays a close pattern, both directions are considered and analysed. Fourteen techniques, including spectral and statistic information, are used as features from the time-domain signal, $x(t)$ (see subsection 3.2.3).

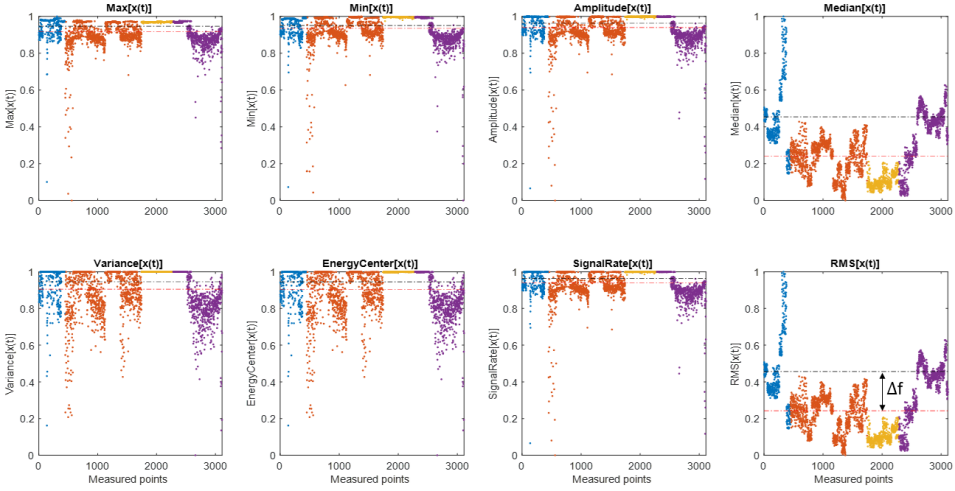
The turbine data acquisition consists of nine days of normal operation in a total of 446 measurements, nine days of operation with FC failure in a total of 1298 measurements, ten days of operation with RI failure in a total of 539 spectra, and seven days of operation with AI failure summing 896 measurements. The total features comprised 3179 measurements and summaries of operation state classification schemas. The features are grouped in a dataset containing the four operational states and illustrated in Fig 4.2. Some methods for feature extraction are insensitive to failure, such as the energy, kurtosis, moment order, and Shannon entropy, which are discarded in the following normalisation and analysis. Aside from this issue, the amplitude variation among the features is small, as shown in 4.1.

Table 4.1: Features relative change obtained with the normal operation and failure condition threshold with and without normalisation.

Δ_f	Features							
	Max	Min	Amplitude	Median	Variance	Energy	Signal rate	RMS
Non normalised	0.0056	0.0031	0.0043	0.0014	0.0000	0.003978	0.0026	0.0014
Normalised	0.0305	0.0154	0.0235	0.2126	0.0405	0.0407	0.0235	0.2137



(a)



(b)

Figure 4.2: Eight feature extractors techniques applied in the raw temporal signals. (a) Non-normalised 'raw' features and (b) normalised features. NO is represented by blue dots, FC by orange dots, RI by yellow dots, and AI by purple dots. The black dashed-dot line is the threshold of the NO mean value, and the red dashed-dot line is the reference of the mean value of the failure condition.

Table 4.1 imposes a great challenge for the ML algorithm to find a pattern and further perform the classification. To cope with this issue, the proposed damage index for the feature condition extraction incorporates normalisation as the feature relative change expressed as in Eq 3.9. This

normalisation method ensures that the features are scaled between zero and one, preserving their essential characteristics while enabling consistent feature comparison. Table 4.1 quantifies the distance between normal operation and failure thresholds applied to the 'raw' features computed with the previous techniques, where, in all cases, the Δf returns minimum values towards zero. Using the normalised damage index related to the relative change proposed in Eq. 3.9, we could scale the Δf values without losing the intrinsic dynamic behaviour over the observation time and impose the features' normalisation between unity and zero. Where close to unity, it is considered NO and toward zero failure operation.

The selected features for evaluation include the max and min values, amplitude range, skewness, RMS, variance, energy centre and signal rate. These features best capture the variations in signals across different operational states. The eight normalised features are then used to generate the global dataset, the input data for the unsupervised k-means algorithm. Figure 4.3 demonstrates the dataset's tabular and visual organisation, which involves binary and multiclass classifications of the turbine's operational conditions. The RMS and median features are placed on top, and the others are grouped at the bottom. For binary classification, three datasets containing the eight features are prepared, each containing information for pairs of states: NO and FC (Fig 4.3a), NO and RI (Fig. 4.3b), and NO and AI (Fig. 4.3c). For multiclass classification, the dataset includes information for all four operational states: NO, FC, RI, and AI (Fig. 4.3d). The black dashed-dot line represents the threshold of the NO condition, while the red dashed-dot line serves as the failure condition reference threshold. The NO condition threshold is the reference to determine the normal and failure data points.

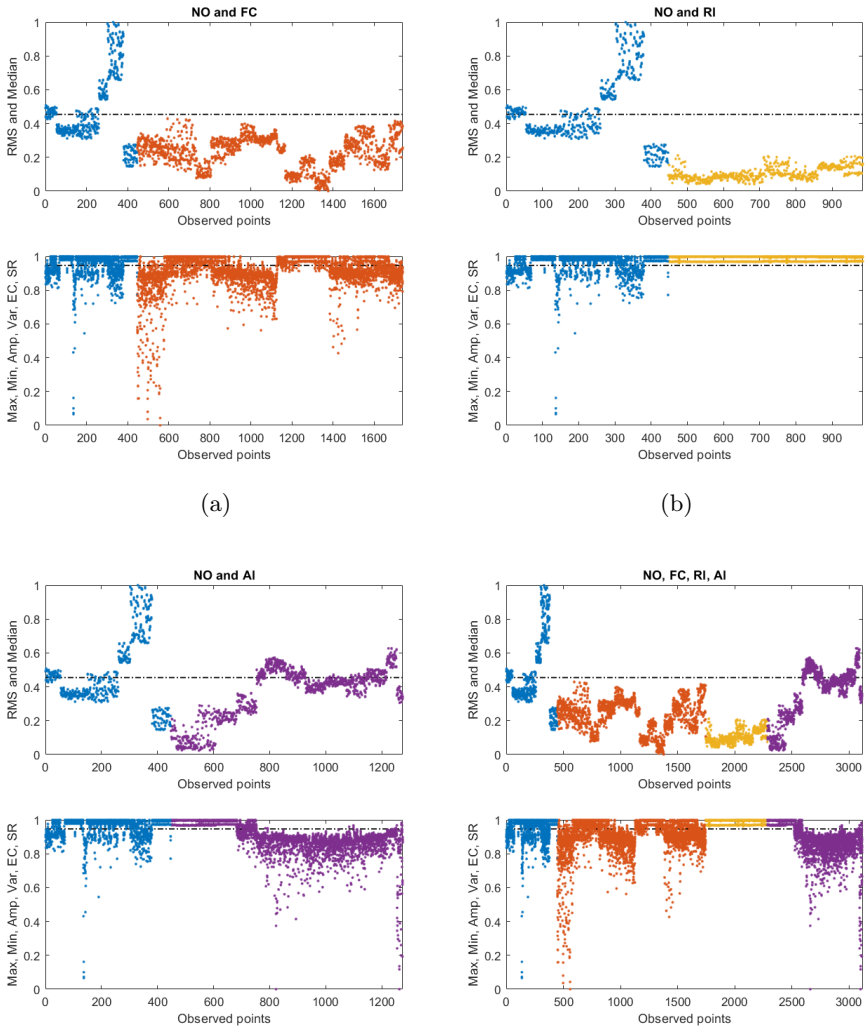


Figure 4.3: Representation of the binary dataset composed of NO-FC condition (a), NO-RI (b), NO-AI (c), and Multiclass dataset, including the four turbine operation condition (d). The black dashed-dot line is the threshold of the NO condition, and the red dashed-dot line is the mean value of the failure condition reference.

After data reorganisation and analysis, we treated our dataset as containing unknown conditions. We applied the unsupervised K-means model to evaluate each binary and multiclass case, classifying and clustering the

samples. The clusters obtained from the K-means algorithm were then used as inputs to assess evaluation metrics, including the confusion matrix, accuracy, precision, recall, and F-score. These metrics were calculated using SVM, KNN, NB, RF, DT, and XGB algorithms. The dataset was split into 75% for training and validation and 25% for testing. The training set was further divided, with 25% of it allocated for validation, resulting in 56.25% of the total data used for training, 18.75% for validation, and 25% for testing. A stratified sampling approach ensured consistent fault distribution across all subsets, minimising sample bias.

The explicit description of the datasets used for training, testing and validation is described in Table 4.2. For the binary classification tasks, data splits were applied uniformly: NO-FC included 981 samples for training, 436 for testing, and 327 for validation; NO-RI had 726 for training, 323 for testing, and 243 for validation; and NO-AI comprised 545 for training, 243 for testing, and 182 for validation. The multiclass scenario covered all turbine operational states (NO, FC, RI, and AI). The dataset consisted of 1,751 training samples, 779 testing samples, and 584 validation samples. This data-splitting strategy ensures that each fault type is adequately represented, providing a robust and accurate evaluation of the model's performance.

Table 4.2: Explicit description of the datasets used for training, testing and validation.

Classification	Cases	Training	Test	Validation
Binary	NO and FC	981	436	327
	NO and RI	726	323	243
	NO and AI	545	243	182
Multiclass	NO, FC, RI and AI	1751	779	584

4.1.4 Results and discussion

Using the K-means algorithm, the dataset was grouped into classes and visualised in scatter plots to illustrate pattern recognition and cluster identification. This representation enabled us to observe correlations and recog-

nise data patterns from normal and fault operations. The distinct patterns emerging in each category indicate how different faults impact the turbine's dynamic response, highlighting the most informative variables for each condition. Variables with minimal overlap between normal and fault conditions are particularly favourable for machine learning algorithms, which can influence these distinct patterns to classify and differentiate the operational states. These feature clusters indicate that these variables exhibit consistent grouping and dispersion patterns, making it easier to identify each operational condition.

Unsupervised classification and clustering using K-means

In binary classification, the dataset is divided into two classes, normal and failure operation, using the k-means algorithm. The three NO-AI, NO-RI, and NO-FC classifications compare the normal state (NO) and a specific type of fault (AI, RI, or FC). The k-means algorithm utilises the combined information from all provided features to cluster attributes according to identified labels. Thus, increasing the number of features enhances the pattern recognition among the features and classification accuracy. Each feature is presented separately for easy visual identification and analysis. Therefore, the classification assumes the combinations of the eight features.

Figures 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9 show scatter plots that highlight the correlation between features in the binary classification, where the model identifies only two types: normal operation and failure. This statistical pattern recognition analysis helped us identify patterns and clusters in the dataset. The blue dots indicate the normal operation, the orange dots FC failure, the yellow dots RI failure, and the purple dots AI failure. The features representing the NO state are concentrated in higher graphics values that tend to unity. This indicates that, under normal conditions, the system tends to cluster at high levels for eight features used in the analysis. In contrast, lower values could indicate more irregular or unstable behaviour, typical of fault conditions, as shown in Fig. 4.3.

Features are estimated using signals from the GEN sensor's Ys- and

Zs-axis directions. Figures 4.4, 4.6 and 4.8 show the features' correlation with the Ys direction. The data dispersion concentrates more on a few features, demonstrating clearer groupings and defined correlations. In the Zs direction, represented by Fig. 4.5, 4.7 and 4.9, there is greater dispersion in some features, indicating less evident separability or correlation for these variables. These feature clustering patterns provide valuable insights into the system's behaviour and help distinguish the turbine's operational conditions. In this case, K-means clustering classification differentiates effectively between normal (NO) and fault states in certain features. The feature data points cluster around a normalised value range for each operational failure. This indicates that K-means can accurately group the data and recognise patterns conducive to reliable classification and robust machine learning model evaluation.

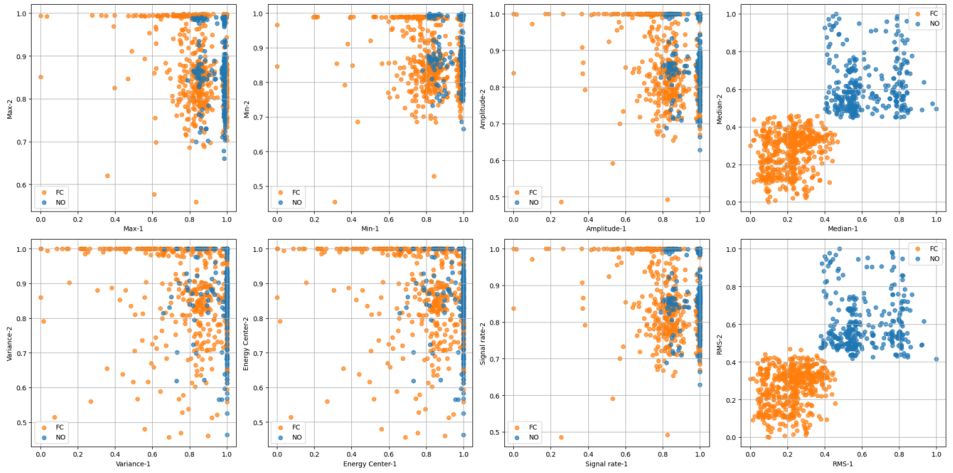


Figure 4.4: Correlation scatter plots between NO and FC failure obtained for the eight normalised features, direction Ys.

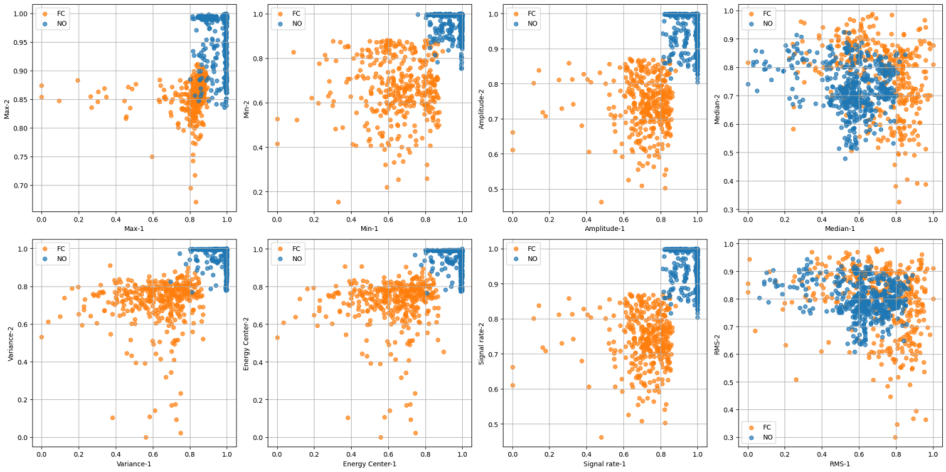


Figure 4.5: Correlation scatter plots between NO and FC failure obtained for the eight normalised features, direction Zs.

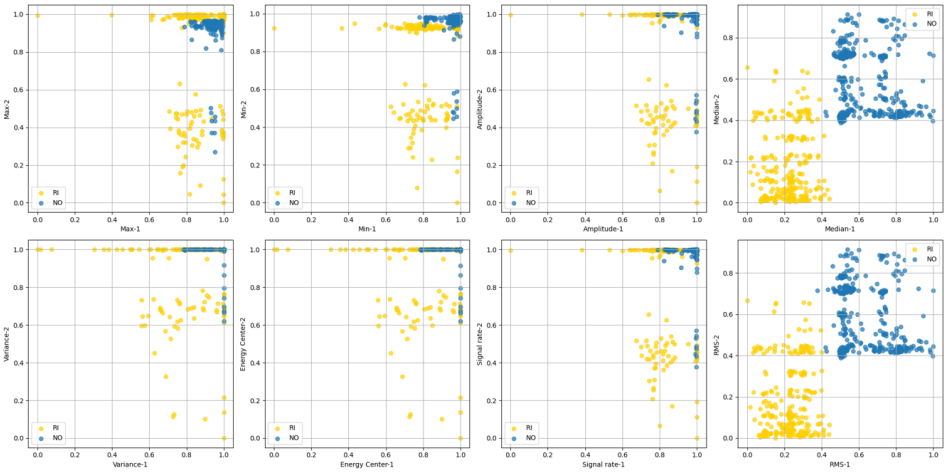


Figure 4.6: Correlation scatter plots between NO and RI failure obtained for the eight normalised features, direction Ys.

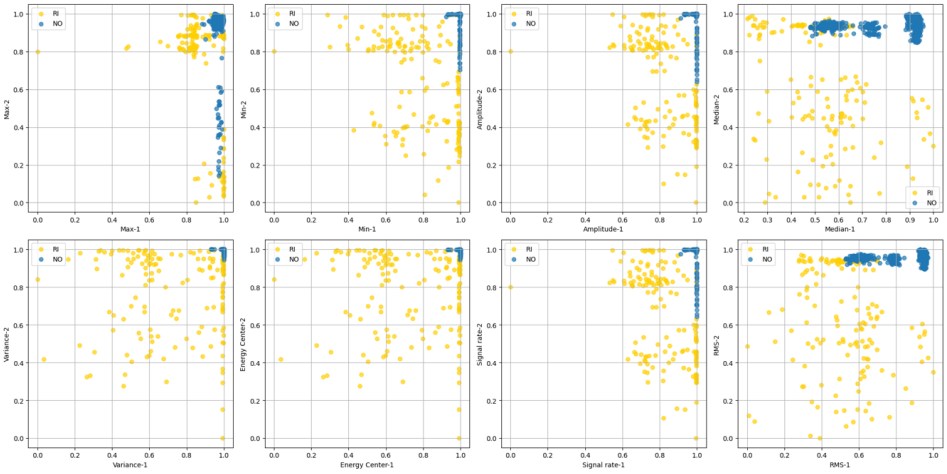


Figure 4.7: Correlation scatter plots between NO and RI failure obtained for the eight normalised features, direction Zs.

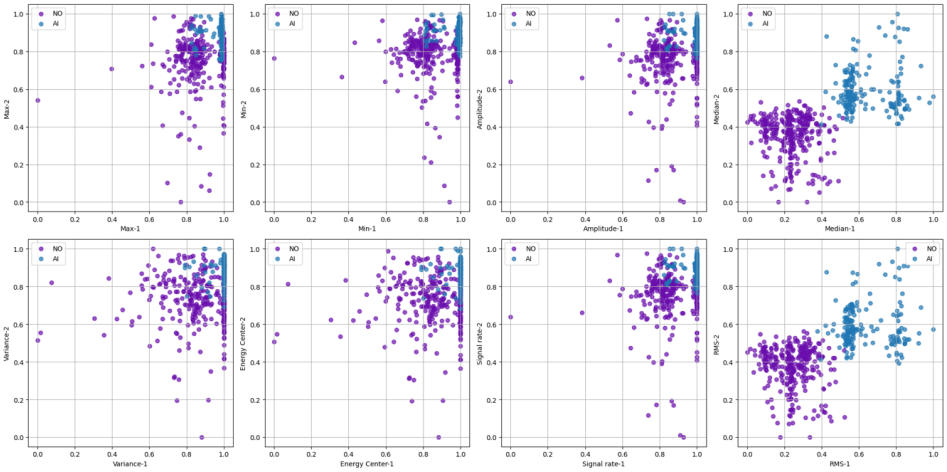


Figure 4.8: Correlation scatter plots between NO and AI failure obtained for the eight normalised features, direction Ys.

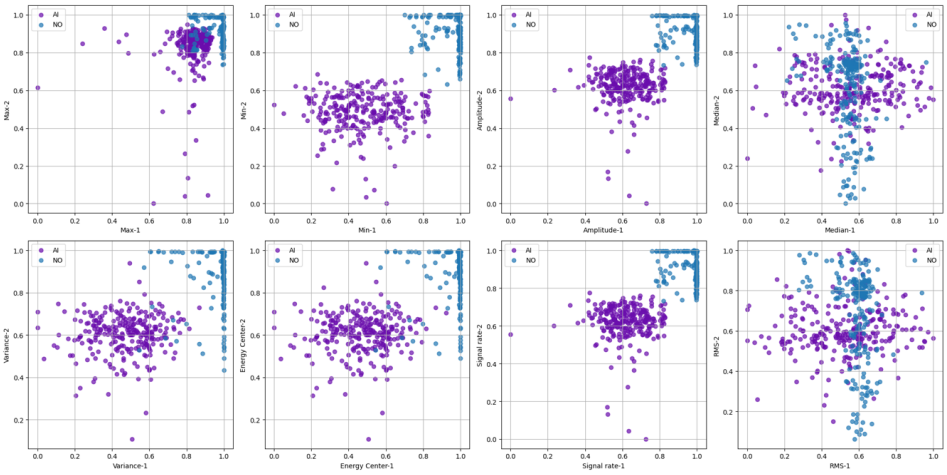


Figure 4.9: Correlation scatter plots between NO and AI failure obtained for the eight normalised features, direction Zs.

The K-means algorithm identifies four clusters corresponding to NO, FC, RI, and AI operations in the multiclass classification and organises the dataset in the response cluster. These clusters allow for further classification validation through evaluation metrics. Figures 4.10 and 4.11 present scatter plots of the features clustered by K-means, each class represented by a distinct colour, enabling prompt identification and evaluation of overlaps of the operational state. In the Ys-direction (Fig. 4.11), the cases show less dispersion in some features. Certain variables in the Zs-direction (Fig. 4.10) separate the normal state and the fault types with greater distributions. In both directions, there is an overlap between fault types, suggesting that these variables alone may not fully distinguish specific failures. This individual feature analysis highlights patterns, but when classification models are applied, features are combined with operational classes with accuracy and quantification of classification metrics. Although some features help differentiate the normal condition from specific fault classes, an effective separation between conditions generally requires a combination of multiple characteristics.

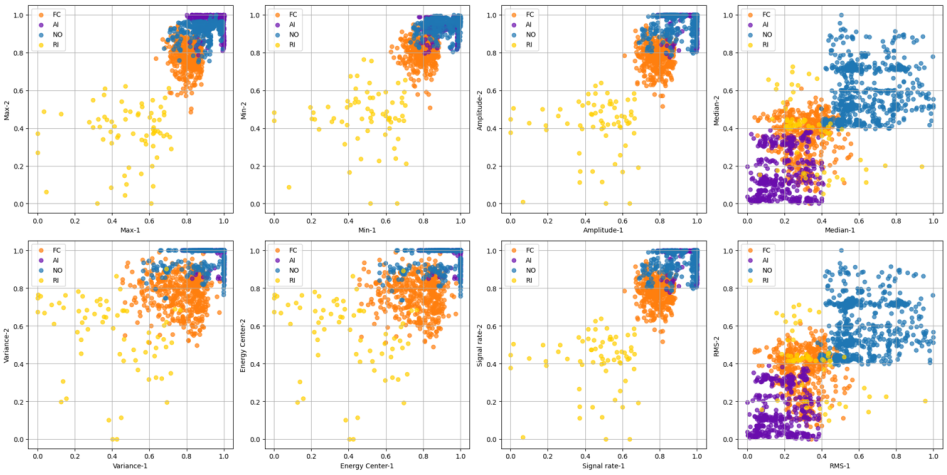


Figure 4.10: Correlation scatter plots between NO, FC, RI, and AI failure for the eight normalised features, direction Ys.

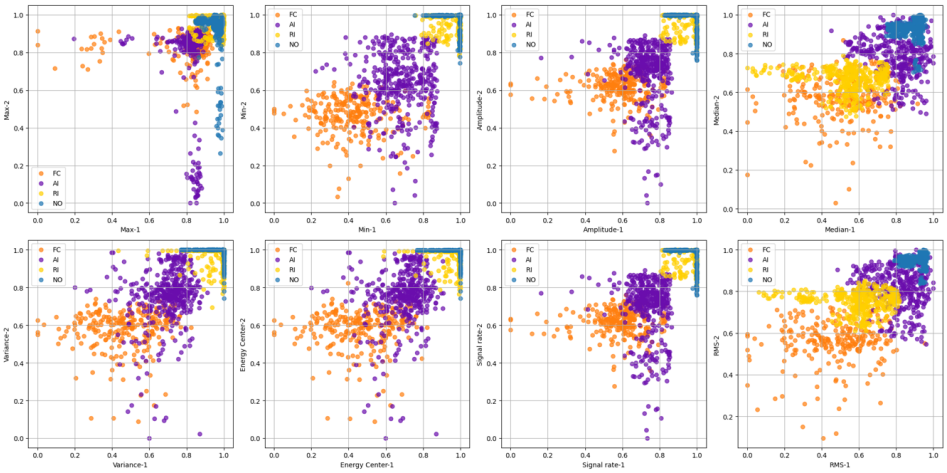


Figure 4.11: Correlation scatter plots between NO, FC, RI, and AI failure for the eight normalised features, direction Zs.

In summary, the results of the dispersion correlation diagrams are analysed considering two distinct measurement directions associated with the turbine's dynamic responses: the Ys-axis direction, which represents the lateral vibration (side-to-side vibration) of the turbine, measured by the sensor, and the Zs-axis direction, which corresponds to the vibration along the X-axis of the turbine, characterising the longitudinal vibration (fore-aft

vibration). These two directions allow a more detailed and segmented investigation of the turbine's dynamic characteristics under different operating conditions. In all cases, the normalization process helps identify the operational conditions, as it was pre-established that unity represents normal operation and values approaching zero indicate fault. The K-means algorithm organizes the clusters, providing straightforward identification, validated by the feature dataset behaviour observed in Fig. 4.3. The FC feature exhibits the greatest dispersion, followed by AI, RI, and NO, as clearly shown in the dispersion diagrams and features observation (Fig. 4.3a- 4.3d).

Operational assessment

The clusters generated by the unsupervised K-means serve as input for various classification machine learning algorithms, including kNN, SVM, DT, RF, Naive Bayes, and XGB. These algorithms perform the final classification based on the initial unsupervised clustering, providing outputs that include confusion matrices and performance metrics. The selection of hyperparameters is based on the investigations in [143, 38], which specified optimal configurations for this assignment. For SVM, a linear kernel was used with a penalty parameter of C=100, a one-versus-one multiclass strategy, and a $1e^{-3}$ tolerance. For kNN, the number of neighbours was set to $k_n = 3$, using the Euclidean metric, uniform weights, and leaf size 30. In both the RF and DT algorithms, the number of trees was fixed at 100, with a maximum depth of 3 and a Gini splitting criterion. The Naive Bayes algorithm employed a Gaussian-NB model, and for XGB, the XGBClassifier model was applied. These configurations achieved high accuracy in fault classification and demonstrated robustness across various experimental scenarios. Accordingly, this study adopts these hyperparameter configurations to optimise operation state accuracy and enhance the performance analysis of the wind turbine.

Table 4.3 presents the performance of machine learning models applied to the Ys- and Zs-axis directions in binary classification tasks (NO-FC, NO-RI, and NO-AI) evaluated through cross-validation and metrics such

as accuracy, precision, recall, and F1-score. Overall, the Ys direction outperforms Zs in most cases, which indicates that the FORE-AFT vibration plane might be more sensitive to the damages.

Among the ML methods used for operation classification, the SVM algorithm achieves excellent metrics, reaching a value of 1 across all evaluated measures (accuracy, precision, recall, and F1-score) in the three binary classification scenarios. The performance of SVM is especially remarkable in NO-RI, where it attains peak scores. Although KNN, NB, DT, RF, and XGB also show strong metrics results in Ys, with less pronounced variations, SVM stands out, offering greater interpretation and precision. This analysis highlights the critical importance of considering the direction of the data when training classification models, as the orientation can significantly influence performance. Confusion matrices are generated for all algorithms, but based on the results, the SVM in the Ys direction is selected for a detailed analysis across the three scenarios. This selection allows for a focused presentation, particularly on the educational aspects of the findings, given the extensive number of graphical results obtained.

Table 4.3: Performance metrics of the ML algorithms SVM, KNN, NB, RF, DT, and XGB for binary classification in directions Ys and Zs.

Performance Metrics	Cases	SVM		KNN		NB		RF		DT		XGB	
		Ys	Zs	Ys	Zs	Ys	Zs	Ys	Zs	Ys	Zs	Ys	Zs
Cross-Validation	**NO-FC**	0.993	0.995	0.989	0.989	0.905	0.969	0.991	0.989	0.991	0.982	0.988	0.985
	NO-RI	1.000	1.000	0.997	0.993	0.914	0.927	0.994	0.994	0.985	0.982	0.985	0.979
	NO-AI	0.985	0.996	0.978	0.978	0.932	0.956	0.982	0.987	0.978	0.971	0.976	0.972
Accuracy	**NO-FC**	0.998	0.988	0.991	0.988	0.924	0.956	0.995	0.986	0.993	0.972	0.998	0.984
	NO-RI	1.000	1.000	0.997	0.988	0.941	0.926	0.994	0.988	1.000	0.975	0.994	0.991
	NO-AI	0.996	0.983	0.983	0.975	0.947	0.955	0.979	0.984	0.971	0.979	0.967	0.975
Precision	**NO-FC**	0.998	0.988	0.991	0.988	0.924	0.956	0.995	0.986	0.993	0.972	0.998	0.984
	NO-RI	1.000	1.000	0.997	0.988	0.941	0.926	0.994	0.988	1.000	0.975	0.994	0.991
	NO-AI	0.996	0.983	0.983	0.975	0.947	0.955	0.979	0.984	0.971	0.979	0.967	0.975
Recall	**NO-FC**	0.998	0.988	0.991	0.988	0.924	0.956	0.995	0.986	0.993	0.972	0.998	0.984
	NO-RI	1.000	1.000	0.997	0.988	0.941	0.926	0.994	0.988	1.000	0.975	0.994	0.991
	NO-AI	0.996	0.983	0.983	0.975	0.947	0.955	0.979	0.984	0.971	0.979	0.967	0.975
F1-Score	**NO-FC**	0.998	0.988	0.991	0.988	0.924	0.956	0.995	0.986	0.993	0.972	0.998	0.984
	NO-RI	1.000	1.000	0.997	0.988	0.941	0.926	0.994	0.988	1.000	0.975	0.994	0.991
	NO-AI	0.996	0.983	0.983	0.975	0.947	0.955	0.979	0.984	0.971	0.979	0.967	0.975

The confusion matrices presented in Figure 4.12 illustrate the binary classification executed by the SVM algorithm. The model demonstrated excellent classification in the NO-FC case, with 268 correct classifications in the NO class (61.47%) and 167 in the FC class (38.30%), recording only a

minimal error. In the NO-RI scenario, the performance was also satisfactory, with 192 correct classifications in the NO class (59.44%) and 131 in the RI class (40.56%), without any false positives. Finally, in the NO-AI case, the model achieved 159 correct classifications in the AI class (65.43%) and 83 in the NO class (34.16%), indicating a slight drop in performance compared to the other scenarios, possibly due to the higher complexity or differences in the distribution of the data for these classes.

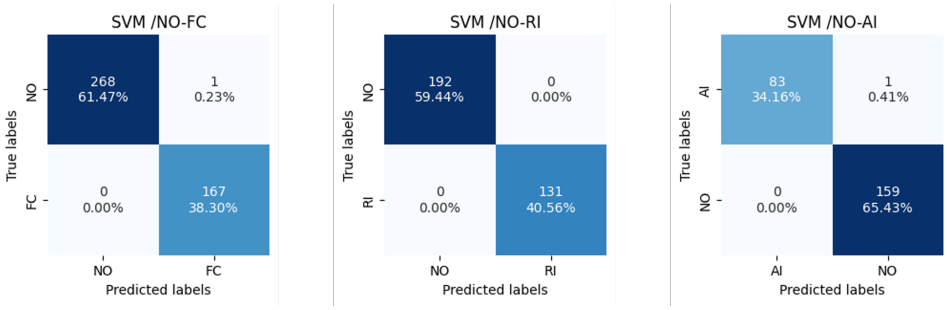


Figure 4.12: Confusion matrix of binary operational classification for NO-FC, NO-RI, and NO-AI cases using the Ys-axis direction data.

The performance metrics of the SVM, KNN, NB, RF, DT, and XGB algorithms for a multiclass classification task are summarised in Table 4.4. The evaluation is based on cross-validation, accuracy, precision, recall, and F1-score, comprehensively assessing each algorithm's effectiveness in the Ys and Zs directions. SVM shows the best-performing model overall, achieving superior results in the Zs direction. KNN, NB, and RF also showed consistent performance, with the Zs direction generally outperforming Ys. In contrast, the DT and XGB models demonstrated a slight advantage in the Zs direction, highlighting the importance of considering data orientation to optimise algorithm performance. Based on SVM's exceptional results, it will be used for further analysis, including the evaluation of confusion matrices in the Zs direction, where it achieved the highest performance.

Table 4.4: Performance metrics of machine learning algorithms (SVM, KNN, NB, RF, DT, XGB) for multiclass classification in directions Ys and Zs.

Performance Metrics	SVM		KNN		NB		RF		DT		XGB	
	Ys	Zs	Ys	Zs	Ys	Zs	Ys	Zs	Ys	Zs	Ys	Zs
Cross-validation	0.986	0.990	0.967	0.975	0.865	0.909	0.972	0.975	0.951	0.964	0.954	0.956
Accuracy	0.987	0.995	0.961	0.961	0.879	0.928	0.979	0.977	0.963	0.955	0.982	0.976
Precision	0.987	0.995	0.961	0.961	0.879	0.928	0.979	0.977	0.963	0.955	0.982	0.976
Recall	0.987	0.995	0.961	0.961	0.879	0.928	0.979	0.977	0.963	0.955	0.982	0.976
F1-Score	0.987	0.995	0.961	0.961	0.879	0.928	0.979	0.977	0.963	0.955	0.982	0.976

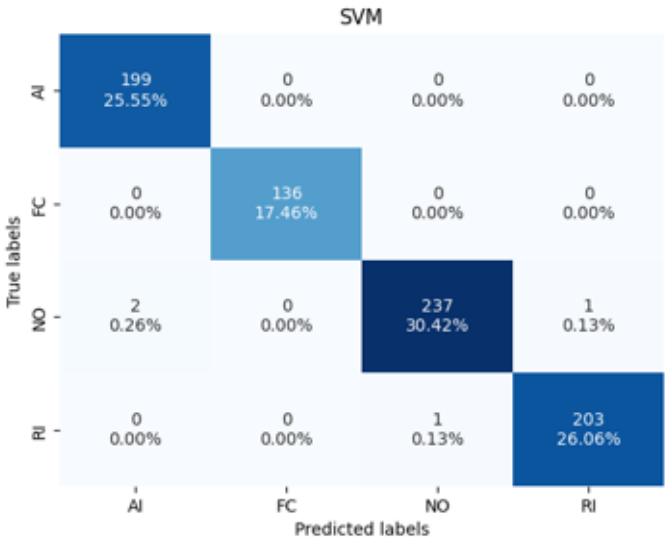


Figure 4.13: Confusion matrix of multiclass SVM operational classification based on Zs-axis direction data.

Figure 4.13 presents the confusion matrix that demonstrates the performance of the SVM model in classifying the AI, FC, NO, and RI classes. The AI class had 199 examples correctly classified as AI (25.55%), with no false positives, indicating that the model accurately identified most of the examples from this class. In the FC class, 136 examples were correctly classified as FC (17.46%), with a small error of 2 FC examples classified as NO (0.26%), suggesting a slight difficulty distinguishing between FC and NO. For the NO class, 237 examples were correctly classified as NO (30.42%), with only one misclassification of NO as RI (0.13%), showing good accuracy. In the RI class, 203 examples were correctly classified as RI (26.06%), with one error of RI being classified as NO (0.13%). The matrix suggests

that the SVM performs well overall, with minimal errors and high accuracy. However, the model showed minor errors between the NO and FC classes and between NO and RI.

Final remarks

The SHM-ML subroutine presented by Al 3 was applied to monitor and assess four operating conditions of the Aventa wind turbine through a structured process involving eight steps, including data acquisition, processing, feature selection, normalisation, data splitting, unsupervised clustering, and subsequent machine learning classification and model evaluation. Eight of fourteen feature extraction methods were sensitive to the different operational failures that the future are input in the unsupervised K-mean for clustering and dataset built for classification and metric evaluation.

In binary classification, SVM emerged as the most robust method, achieving perfect metrics with values equal to 1.0. This reflects its exceptional ability to distinguish between normal conditions and faults across different data orientations, particularly emphasising the Ys direction. Analysis of the confusion matrices further validated the reliability of the SVM model, minimizing classification errors and false positives while providing high precision in structural monitoring. While other algorithms, such as kNN, Naive Bayes, and XGB, also demonstrated commendable performance, SVM stood out for its consistency and stability in results. In multiclass classification, which involved identifying multiple operational states and damage levels, SVM delivered the best performance, excelling in accuracy and its capacity to discriminate between classes. The Zs direction proved to be the most suitable for this task, exhibiting the highest precision and the best capability to differentiate between the various damage levels. Overall, SVM proved to be the best-performing model for binary and multiclass fault detection, demonstrating robustness and effectiveness in monitoring and diagnosing operational conditions of wind turbine components.

4.2 Case II: Detection of loosening torque in bolted structure

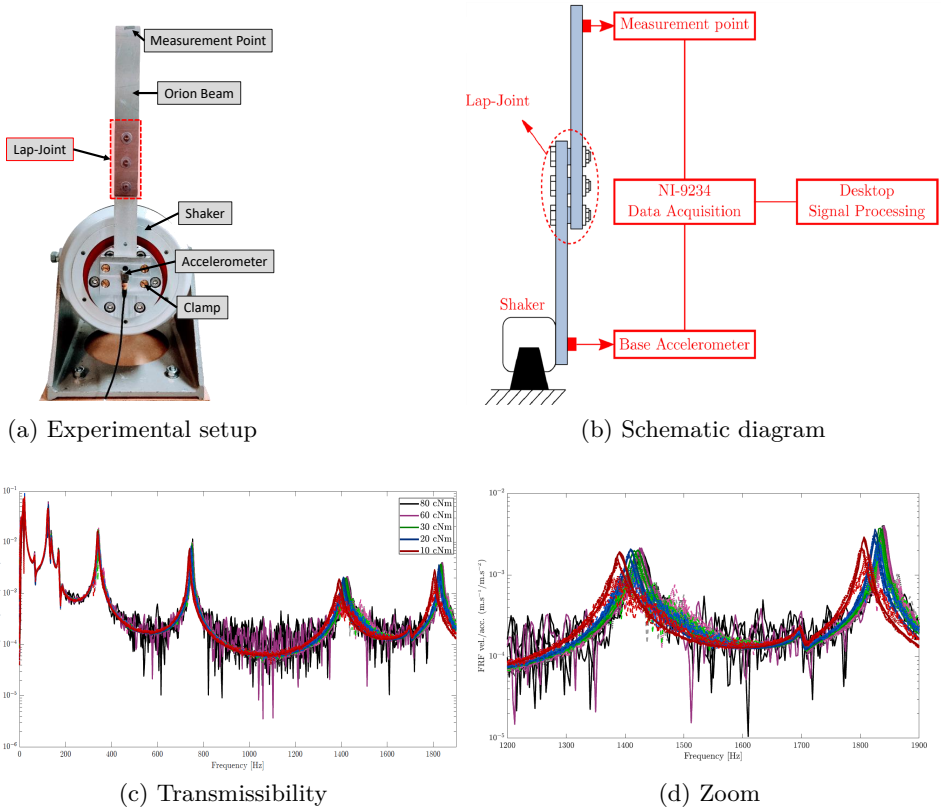
This case study builds upon existing work in bolted joints used for assembly structure analysis, particularly in recognising and detecting loosening torque from a structure's dynamic response. The ML classifiers have been designed to withstand variabilities in raw data, in addition to noise variations, considering the influence of assembling and disassembling the bolted structure during experimental tests, as documented in [159]. To validate the method's applicability, this case presents an implementation of Algorithm 1 from Section 3.1. The present approach simplifies the analysis by eliminating the need to evaluate the most sensitive extraction features, introducing the FRAC concept in obtaining the damage index, which plays a crucial role in our algorithm's methodology. As a result, the main contribution of the ML algorithm architecture for pattern recognition and detection of loosening torque in bolted joints lies in its utilisation of spectral raw signals from experimental tests. The test results show that the algorithms can classify satisfactorily in all three frequency band conditions. These results were published in [38].

4.2.1 Experimental benchmark

Bolt loosening detection from vibration data is challenging due to variability and nonlinear effects from the contact interface in bolted joints. Therefore, this study proposes a data-driven strategy to detect loosening bolts from experimental vibration signals. In the experimental set available in Teloli et al. [275], the authors consider a physical system consisting of two bolted beams (dimension $370 \times 30 \times 2$ mm) in a cantilever and joint lap configuration connected by three bolts with controlled tightening, as shown in Fig. 4.14. The experimental apparatus consists of a load cell (PCB288D01), an electromagnetic Modal Shop shaker (Model K2004E01), a 3D scanning laser, and NI9234 hardware for data acquisition. The excitation spectrum considered was a white noise Gaussian input with the amplitude levels of $1m/s^2, 4m/s^2, 8m/s^2$, and $12m/s^2$ RMS values induced by the shaker at

the clamped end. The tightening torques applied on the beam's bolts were 10 cNm, 20 cNm, 30 cNm, and 80 cNm.

The tightening torques were measured with a Lindstorm MA500-1 torque wrench, and the force acquired with a Futek LTH300 donut-load cell was performed after each experimental run. The experimental data-driven consists of the frequency response obtained by dividing the velocity measured at the beam's free edge by the acceleration measured at the clamp. Three-hundred-sixty response samples were acquired considering all different excitation spectrum amplitude levels and variability in assembling the jointed structure.



(a) Experimental setup

(b) Schematic diagram

(c) Transmissibility

(d) Zoom

Figure 4.14: Physical and schematic drawing of the experimental bolted beams presented in [275, 38].

Figure 4.14 (c) displays four experimental FRFs samples for each torque

and base acceleration level measured over a frequency range from 0 to 1900 Hz, in a total of 20 curves printed in the figure. A different colour line represents the FRF of each torque. The continuous line represents a base acceleration level of 1 m/s^2, the dashed-dot line represents an acceleration level of 4 m/s², the dashed line represents an acceleration level of 8 m/s², and the dotted line represents an acceleration level of 12 m/s². The torque loosening induces clearer changes in higher mode shapes, for instance, on the fifth and sixth modal shapes, which can be more evident as the excitation amplitude levels increase. Both noise and variability are present in the spectral signal across the entire frequency range. When considering the whole FRF signal in the damage index calculation, the estimation of torque loosening can induce a false positive in the prediction and diagnostic because one considered all signal power densities, including the signal in low frequencies, less influenced by the damage. Therefore, the truncated signal ranging from 1200 to 1900 Hz was assumed in the DIs estimation (Fig. 4.14 d). This frequency band was the most affected by the torque loosening.

The structure's vibration signature has been used to detect, locate and quantify damage and anomalies in a structure from changes in its dynamic characteristics [168]. Among the methods that employ the dynamic response, the DI is a metric that correlates a system signal in different states. The reference signal, usually derived from the system considered an undamaged state, correlated to the one provided by the system under the presence of discontinuity or damage [181]. Various DI approaches have been developed to extract signal features in different domains, aiming at identifying structural damage using an indicator that describes the damage as explored in [276, 277, 278, 279]. The DIs are associated with the estimation techniques for damage quantification and reveal important information about the structural health condition. The literature describes a range of DI developed over time. The Frequency Response Assurance Criterion (FRAC) is a damage index representing the correlation between tested frequency responses.

4.2.2 Feature selection and pattern recognition

The ML algorithm is expected to classify the damage state correctly. In this case, DIs close to unity are considered healthy states of the structure for torques of 80 and 60 cNm. In contrast, torque losses are associated with torques of 30, 20, and 10 cNm, indicating structural damage. The classification identifies the structure condition and quantifies it as Healthy (80), Healthy (60), Damage (30), Damage (20), and Damage (10). The DIs are calculated by considering the whole signal spectrum, the signal truncated comprising 5th and 6th mode shapes, and the signal covering only the fifth and sixth modes separately.

From data processing and estimating DIs, two attribute datasets (DI_1 and DI_2), each consisting of 180 samples, were generated based on the selected FRAC DI observations. Having the dataset pre-processed, DI clustering was carried out using the K-means algorithm with a value of K = 5 (for K representing the number of classes) to identify clusters within unlabeled data. Consequently, an object target dataset was created, comprising 180 samples belonging to one of the 5 labelled classes. Further, the samples were randomly divided, with 70% of the data allocated for training and 30% for testing, as defined in Table 4.5. Each classifier was assessed using a 5-fold cross-validation procedure to enhance training accuracy, accomplished by randomly partitioning the training dataset into five distinct subsets.

The dataset grouped in classes by the K-means is shown using a scatter plot to characterise the clustered points, enabling us to observe the correlation in two-dimensional space and recognise a pattern of the torque loosening. Figures 4.15a show the correlation between the two attributes of features (DI_1 and DI_2). The cloud points show an unclear classification of torque loosening, as the points cluster in the range of DIs considered a healthy state of bolted beams. Bolt loosening most affects the dynamic response in higher frequency modes; therefore, by using the whole response signal spectrum, the influence of modes in low frequency intends to dominate the signal power spectrum because they have higher amplitudes. In this case, the torque loosening identification through the DI is compromised.

Therefore, whether the ML algorithm can correctly classify the damage is unclear. All datasets have a considerable correlation, which can induce misleading classification and false diagnosis identification.

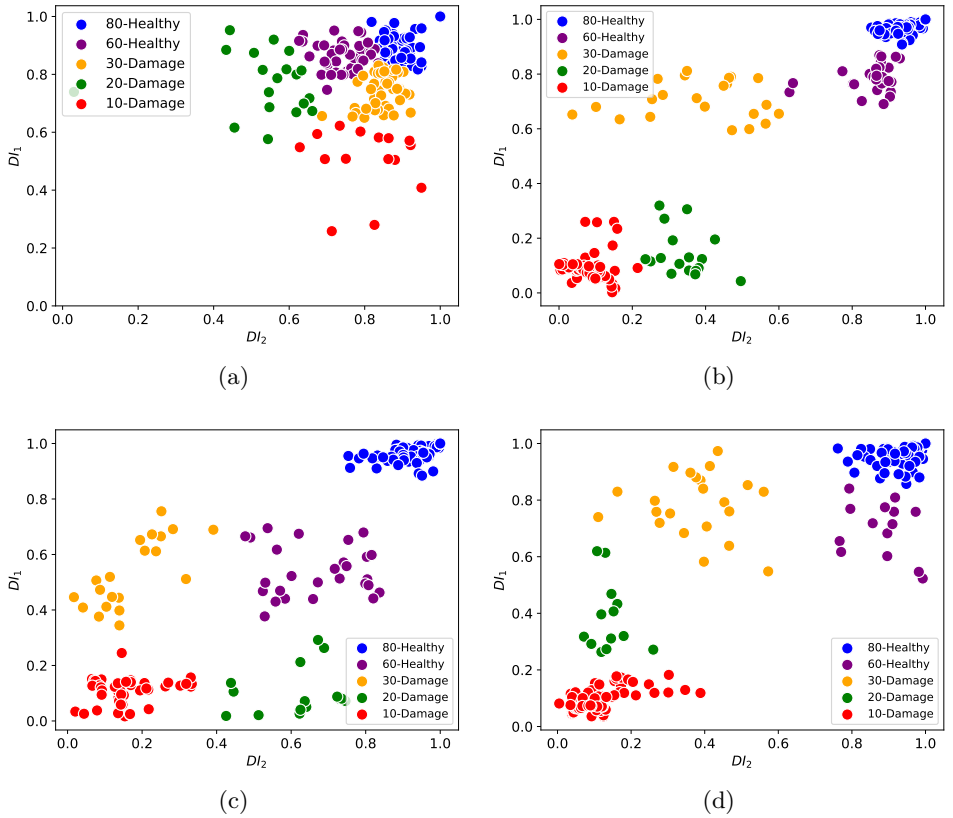


Figure 4.15: Correlation scatter plots of between features DI_1 and DI_2 dataset obtained from five different classes of torques in the frequency ranges of (a) 10 to 1940 Hz, (b) 1250 to 1940 Hz (5th∼6th mode), (c) 1250 to 1550 Hz (5th mode), and (d) 1740 to 1940 Hz (6th mode).

For torques of 80 and 60 cNm, the FRAC DI varies between 1 and 0.9, considered a healthy structure state. In contrast, torque losses are associated with torques of 30, 20, and 10 cNm, with the DIs estimated from 0.8 to 0, indicating torque loosening. Therefore, by following the DI values, the multiclass dataset can be used to identify the structure state and quantify the severity of the damage from a pattern in the DI values.

Assuming only part of the signal is truncated in the frequency range of

the modes' shapes most affected by the torque loosening, the identification turns more consistent. The data correlation is stronger for the sixth mode than the fifth mode shape because of the small change in the frequency response due to the torques from 80 to 30 cNm. Figure 4.15b shows the data set pattern classification for the signal containing information of the fifth and sixth mode shapes and the correlation between the features attributes. In this case, the classification in healthy and damaged states becomes more evident as the torque loosens. The multiclass dataset can recognise the pattern of the toque loosening through the calculated FRAC DIs. Figures 4.15c and 4.15d include information on the mode fifth and sixth, respectively. For each torque level, the DI data points cluster correctly around the range corresponding to the normalised value of DI, indicating that the dataset follows a torque loosening pattern, which can lead to a good classification of the ML algorithm.

Table 4.5: Torque values, labelled classification, train, and test data splitting associated with samples number identified healthy or damaged.

		Classification/Regression					
Torque	**Frequency**	**Train dataset**			**Test dataset**		
(cNm)	**range (Hz)**	**Healthy**	**Damaged**	**Total**	**Healthy**	**Damaged**	**Total**
80,60,30, 20,10	1250∼1550 (5th∼6th modes)	60	66	126	26	28	54
	1740∼1940 (5th mode)	69	57		30	24	
	1740∼1940 (6th mode)	59	67		25	29	
Multiclass label classification (torque): Healthy(80), Healthy(60), Damaged(30), Damaged(20), Damaged(10)							

The approach for assessing torque loosening based on spectrum signal can effectively capture the intricate dynamics and torque loss processes within bolted joints across varying torque levels. As torque drops, DI consistently decreases in most scenarios. Nevertheless, the FRAC method consistently outperforms others, indicating a clear trend of decreasing values with reduced torque. Therefore, adopting DI as a normalisation factor in the vibration dataset becomes the input for unsupervised and supervised machine learning algorithms. Consequently, due to DI's normalisation effect on the data, the range of based motion acceleration excitation is dissociated

from the subsequent analysis and estimation. In our context, using normalised data is one of the advantages of our algorithm, which lies in the DI procedure. However, a limitation of our proposed algorithm in this paper is that it assumes raw data instead of DI. We are currently exploring ideas to address this issue, but they are beyond the scope of this paper.

4.2.3 Machine-Learning techniques for bolt-loosening detection

Bolts have the function of connecting and maintaining stability between two pieces that need to be joined. However, this fixation might only be guaranteed in parts of the structure's lifespan, a problem engineering systems face. It is common for the ends joined by the bolts to loosen over time due to external vibrations, dynamic loading, or thermal variations. Bolted joint loosening is damage-like and modifies the connectivity between the components of the structure. Predicting torque losses is essential and helps engineers create control strategies for torque tightening. Based on actual torque data, it is possible to train a classifier model that predicts whether torque loss occurs for a bolted joint and subsequently identifies whether there is damage to the structure.

Machine learning algorithms can be employed in system monitoring using the dataset related to the loss of torque problem. Machine Learning is an automated process that extracts information from data based on a pattern learned through different algorithms. It utilises the learned patterns to predict future data or perform other types of decision-making. This work predicts the state of the bolted connection with unsupervised-supervised learning from the bolted beams' vibration response data-driven. The ML algorithm quantifies the torque loosening using a regression algorithm under different torque conditions as input. The learning approach is divided into an unsupervised-supervised classification and regression problem, as the data has defined attributes. The ML data input is the DIs, and the target variable will be the information on bolt loosening and indicate the influence of the independent variable. Thus, we intended to obtain a previous classification, determine which unclassified category the data belongs to, and then quantify

the torque loosening.

The classifiers algorithm used are the unsupervised K-means to cluster the data and supervised Naive Bayes, Decision Tree (DT), Random Forest (RF), K-Nearest Neighbours (KNN), Support Vector Machine (SVM), and extreme Gradient Boosting (XGBoost) to detect the torque loosening. General theoretical details on the machine learning techniques used here are presented in section 3.3.3. Each algorithm has hyperparameters that must be configured and tested for optimal performance in application cases. In the case of SVM, a linear kernel function was used, and a grid search was conducted to determine the penalty parameter, assumed as $C = 10$. For KNN, the number of neighbours is set to $k = 3$, and the metric is defined as the Euclidean distance. The function weights are uniform, meaning that all points in each neighbourhood are equally weighted, and the leaf size, which affects query construction and speed, is set to 30. For RF and DT, the number of trees in the forest is 100, and the maximum depth is restricted to 3. The minimum sample split is 2, indicating the minimum number of samples required to split an internal node. The minimum sample leaf represents the training samples on each of the right branches, and the minimum sample leaf values are set to 1. The Max features value is set to 'auto', representing the number of features considered when searching for the best split, and the criterion used is the Gini index. In the case of XGBoost, the objective is assumed as *softmax* for multiclass classification using the softmax objective. The learning rate is set to 0.3, which means that each weight in all trees will be multiplied by this value, and the maximum depth is set to 6. In the Naive Bayes classifier, the Gaussian-NB case was selected. Table 4.6 shows the hyperparameters selected for each ML algorithm implemented in this paper. All algorithms were applied using the open-source Scikit-learn library in Python.

After the machine learning algorithm completes its estimation, it becomes crucial to assess the stability and accuracy of the model. This validation process involves confirming the quantified relationships between variables, which can be accomplished by examining metrics such as accuracy,

Table 4.6: Hyperparameters assumed for each ML algorithms.

Classification Algorithm	**Hyperparameters**
K-Means	Number of clusters = 5; Initialisation = k-means++; Number of times the algorithm is run with different centroid seeds = 20.
SVM	Kernel: Linear; C = 10.
K-NN	Metric: Euclidean distance; Number of neighbours: 3.
Naive Bayes	Gaussian
Ramdon Forest	Number of forest trees = 100; Max_depth = 3; Minimum division = 2; Minimum value of sample sheet = 1; Criterion = Gini Index.
Decision Tree	Criterion = Gini Index; Splitter = 'best'; max_depth = 3.
XGBoost	Objective ='multi:softmax'; max_depth = 6; learning rate = 0.3.

score, precision, and recall [241, 240]. However, it's important to note that these metrics primarily reflect the ML model's performance on the data it was trained on. Therefore, the ML model's cross-validation using a separate dataset is necessary to ensure that it successfully captures the underlying patterns in the data, and a reliable validation set indicates a model with low bias or variance. In the damage assessment, the validation and the cross-validation of the ML algorithms are explored. The evaluation metrics of the ML algorithm are addressed to compare the damage detection capability through their accuracy and the confusion matrix. Accuracy close to 100% is considered a good performance. In all cases presented in this paper, the ML model exhibits excellent performance, as evidenced by a standard deviation of 5% in the cross-validations obtained through five different cluster data.

Initially, an unsupervised clustering algorithm, K-means, was applied to cluster the data, which was divided into five classes. From the samples obtained through the selection of attributes, the algorithms were evaluated by comparing metrics for the classification algorithm in the training and test sets. The idea was to compare the classification metrics, considering the damage index calculated from the FRFs and torques as a characteristic. Therefore, the studies on the dataset investigate the feasibility and accuracy of the six supervising machine learning on the performed classification. The dataset includes five torque identification classes (see Table 4.5), with two

record attributes as input variables or predictors. Loosening torque identification, considered as damage, is performed with 180 samples separated into three different assemblies and divided into training and testing data. The classification method employed here is "one versus one" [280], where one multiclass classification problem turns into ten binary class classification problems. The metrics for multiclass classification are shown in Table 4.7 for the torque estimation using the 5th∼6th mode shapes, Table 4.8 only with the 5th mode, and Table 4.9 with the 6th mode shape. The estimation using the whole frequency band is discarded because of the low accuracy in the DI grouping.

Table 4.7: Comparison between metrics of experimental test ML algorithms for FRF data (5th∼6th mode).

Performance Metrics	**SVM**	**KNN**	**NB**	**RF**	**DT**	**XGB**
Cross-validation	97,8	97.2	94.4	98.9	98.9	98.9
Accuracy	98.1	98.1	85.1	100	100	90.7
Precision	98.1	98.1	85.1	100	100	90.7
Recall	98.1	98.1	85.1	100	100	90.7
F1-Score	98.1	98.1	85.1	100	100	90.7

Table 4.8: Comparison between metrics of experimental test ML algorithms for FRF data (5th mode).

Performance Metrics	**SVM**	**KNN**	**NB**	**RF**	**DT**	**XGB**
Cross-validation	98.3	98.9	98.3	99.4	100	99.4
Accuracy	98.1	98.1	100	100	100	100
Precision	98.1	98.1	100	100	100	100
Recall	98.1	98.1	100	100	100	100
F1-Score	98.1	98.1	100	100	100	100

All torque classification performed with the three signals present cross-validation ranging from 94.4% to 100%, and the accuracy, precision, recall and F1-score range from 88.9% to 100%. Torque estimation using the signal from the 5th mode shape presents the higher metrics, showing the efficiency of the algorithms in detecting the most specific conditions for the health and damage state of the bolted beam. The results consistently demonstrate similar performance, yielding accurate predictions on previously overlooked

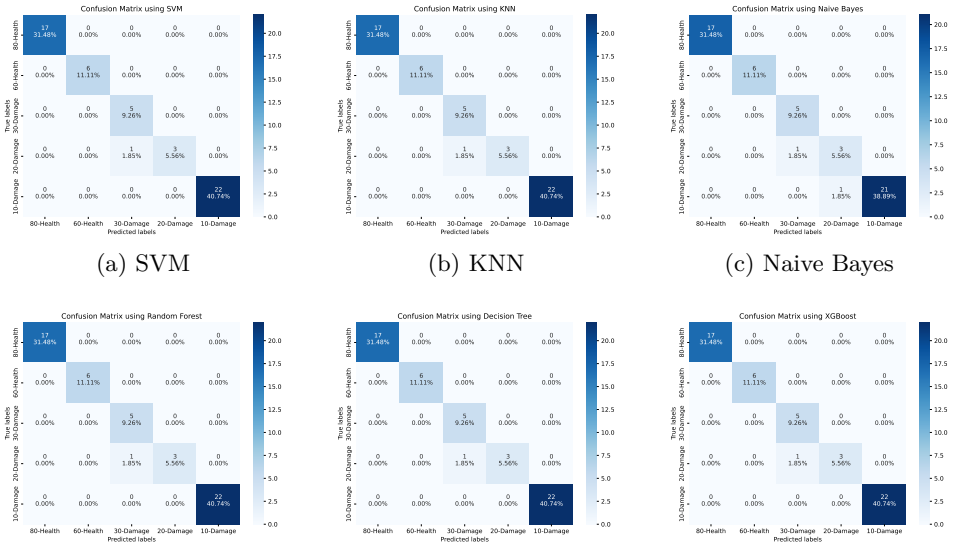
Table 4.9: Comparison between metrics of experimental test ML algorithms for FRF data (6th mode).

Performance Metrics	**SVM**	**KNN**	**NB**	**RF**	**DT**	**XGB**
Cross-validation	96.7	97.2	97.8	97.8	94.4	95
Accuracy	98.1	98.1	92.6	98.1	94.4	88.9
Precision	98.1	98.1	92.6	98.1	94.4	88.9
Recall	98.1	98.1	92.6	98.1	94.4	88.9
F1-Score	98.1	98.1	92.6	98.1	94.4	88.9

datasets. The XGBoost performed K-folds ($K_f = 5$) estimation, where 88%, 95.6%, 94.4%, 97% and 100% were achieved. The cross-validation of 95% given by the XGBoost is estimated by the mean value of the five validation preview tests. It indicates a high level of consistency in the model's performance across different cross-validation iterations, with minimal fluctuations in its learning behaviour. Cross-validation is a valuable tool for evaluating the algorithm's effectiveness by assessing its performance on various data splits. It provides a robust understanding of how the model behaves regarding overfitting. Additionally, the other metrics support the model's strong classification performance.

The confusion matrix [240] is also vastly employed to verify the data classification, which provides the correct configurations of the classified data. It minimises the error in the damage, and the model's successes offer a comparison between actual and predicted values, where the labels are considered "Positive" and "Negative" [241]. In the case of the torque loss problem, the matrix elements are characterised as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The matrix's main diagonal values show how many correct model predictions there are for each class. In this application, the confusion matrix details the ML classifiers' performance in correctly labelling the data and predicting the severity of the loosening torque, which is divided into five classes: damaged or healthy. Therefore, the confusion matrix represents five rows indicating true classes and nine columns representing the models' predictions. Figures 4.16-4.18 show the confusion matrices of the multiclass classification

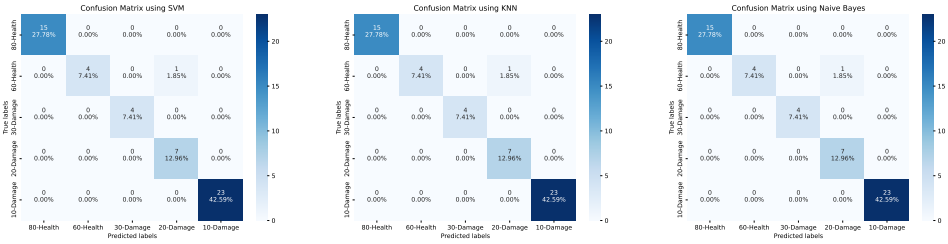
for torque losses for the truncated signal between the 5th and 6th mode shape. The results indicate that a smaller number of samples were misclassified. Most algorithms had only a sample misclassified, representing 1.85% of the total samples used, in this case, the confusion matrix shows the correct and incorrect predictions on each class. For example, by looking at all the values in row four (Figure 4.16a), it can be inferred that, out of four samples, the model predicts that three samples belong to class 20-damage (correct prediction) and one sample belongs to class 30-damage. Overall, the result for a multiclass classification using ML algorithms is satisfactory for this specific task using the dataset for bolt loosening identification. For each class, it is possible to evaluate the model's performance by looking at the confusion matrix in detail.



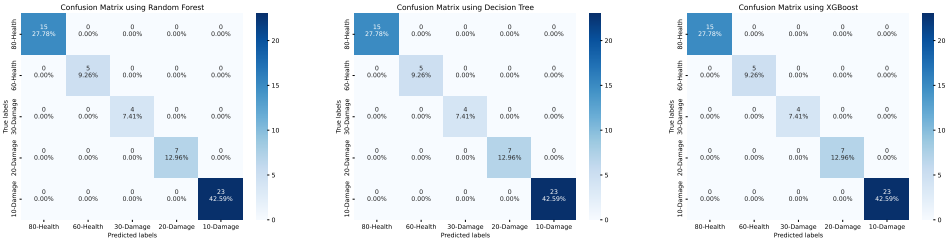
(a) SVM (b) KNN (c) Naive Bayes

(d) Random Forest (e) Decision Tree (f) XGBoost

Figure 4.16: Confusion matrix of Multiclass classification of six ML techniques in the 5th∼6th mode. The values in blue blocks indicate correctly classified points, whereas those in pale blue blocks indicate misclassified points.

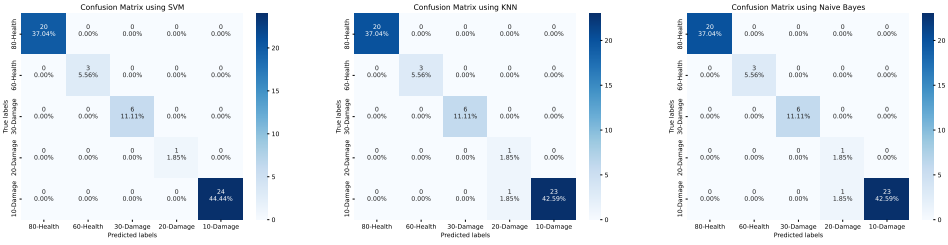


(a) SVM (b) KNN (c) Naive Bayes

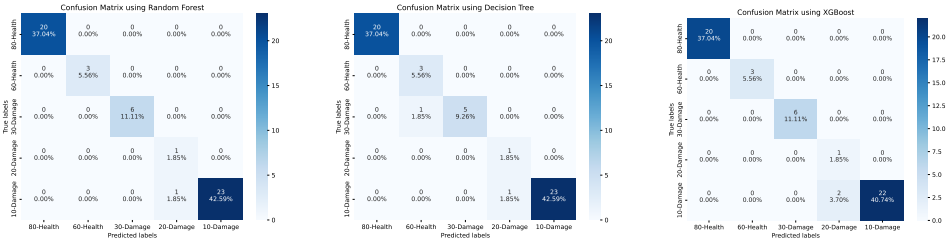


(d) Random Forest (e) Decision Tree (f) XGBoost

Figure 4.17: Multiclass classification confusion matrix of six ML techniques for the 5th mode.



(a) SVM (b) KNN (c) Naive Bayes



(d) Random Forest (e) Decision Tree (f) XGBoost

Figure 4.18: Multiclass classification confusion matrix of six ML techniques in the 6th mode.

Final remarks

The study demonstrated the effectiveness of SHM-ML techniques for detecting bolt loosening torque in bolted joints using vibration-based methods. Supervised classification models achieved high accuracy in identifying damage states. The next section focuses on expanding the dataset, regression techniques, and integrating virtual sensor data augmentation into the SHM-ML to enhance the reliability and applicability of the proposed methodology for continuous structural health monitoring of structure bolted joints.

4.3 Case III: Quantification of loosening torque in bolted structures with virtual sensor integration

Current research on bolt torque loss detection using ML and DL techniques is relatively advanced. However, a significant gap remains in applying ML-based condition assessment methods incorporating data augmentation, uncertainty quantification, and raw vibration spectra acquired from bolted structures for torque loosening monitoring. The uncertainty associated with the assembly system affects the dynamic response and propagates through the monitoring process. Addressing this issue demands a robust technique for torque loosening assessment that also accounts for this epistemic uncertainty propagation, minimizing false indications and reducing false-positive and false-negative detections while incorporating variability errors into the operational evaluation process. This case study proposes a data-driven ML-based condition assessment model to estimate bolt torque loosening. Data augmentation uses a virtual sensor that fuses physical and synthesized data, enhancing dataset quality and improving ML efficiency. To validate the applicability of the method, this case presents an implementation of Algorithm 2 from Section 3.1. The experimental dataset provided in [159] is a physical system consisting of two beams bolted in a cantilever and joint lap configuration, connected by three bolts with controlled tightening torque. The system complexity characterises the experimental data due to variability, nonlinear effects, and uncertainties associated with sub-component

[140, 281, 279]. The main contributions of this study are: *i*) Developing a novel data-driven ML architecture for torque loosening estimation in bolted systems, addressing intrinsic system uncertainties; and *ii*) Proposing a virtual sensor that provides indirect feature measurements and expands the dataset volume. The ML architecture integrates regression algorithms with dedicated data augmentation techniques. The condition assessment of torque loosening detection and quantification ML model is based on seven steps outlined in Figure 4.19. The methodology strategy includes processing the existing data provided from vibration tests, feature extraction, data augmentation strategies through the virtual sensor, feature selection, and regression algorithms for torque loosening and uncertainty quantification. This approach aims to enhance the accuracy and reliability of monitoring and promote proactive maintenance on bolted systems.

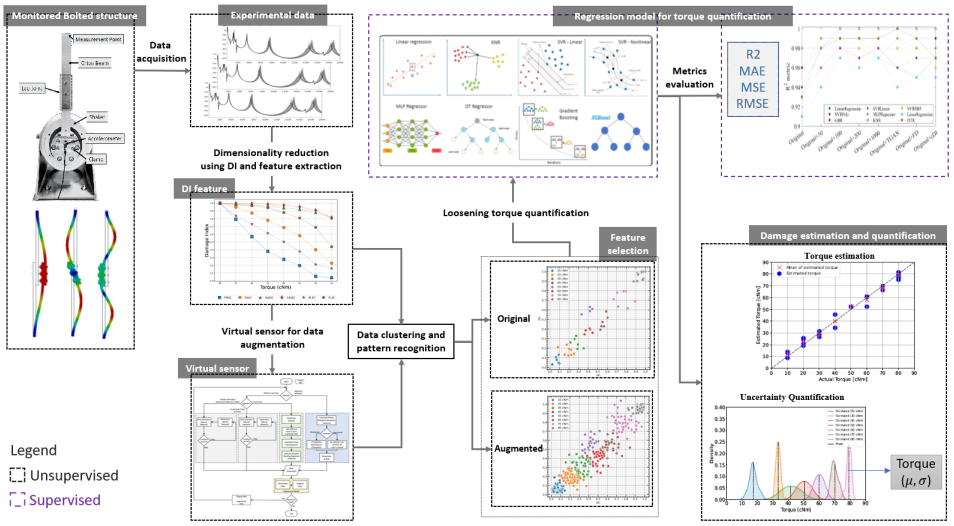


Figure 4.19: Flowchart of a semi-supervised ML model for bolt torque loosening estimation and monitoring (Source: own study).

4.3.1 Feature extraction using damage indices

Feature extraction transforms measured signals into features, also known as attributes, which serve as inputs to a learning algorithm. This process involves changing the original data variables to obtain new forms of nor-

malised data variables [282, 59]. Typically, the reference signal is derived from the system in an assumed health state, compared to the signal provided by the system in the presence of discontinuities or damage [181]. The experimental data consists of the transmissibility responses under different tightening torque values, and the DIs are employed as feature extractors.Figure 4.20 presents the results of the six damage indices calculated and normalised from the vibration signals for different torque levels and frequency bands. The DIs are computed from the entire signal and truncated in a frequency range to reach the 5th to 6th mode shape. The FRAC, FAAC, AIGSC, AIGAC, MDI, and RDI indices show a slightly decreasing trend with decreasing torque for both signal ranges.

Figure 4.20a considers the entire transmissibility signal (0∼2000 Hz) used in the DIs estimation. Except for MDI, all methods maintain a value between 0.95 and 1 for torque levels below 50 cN/m, after which there is a slight decrease, reaching a DI of 0.7. The narrow range of DI variation across torque levels suggests the low sensitivity of these methods to torque, as lower mode shapes are unaffected by loosening, as shown in Fig. 4.14(d). Figure 4.20b presents DIs calculated over a truncated frequency range in the 4th∼6th modes, from 600 to 2000 Hz. The AIGSC and AIGAC indicators exhibit low sensitivity, maintaining constant DI values without significant reduction as torque decreases, indicating their ineffectiveness in damage detection for this analysis. The RDI method shows a DI close to 1 up to 50 cN/m, followed by a gradual decrease with torque, though less sharply than other methods, indicating low sensitivity. In contrast, FRAC, FAAC, and MDI demonstrate greater sensitivity, gradually reducing DI as torque values decrease. Figure 4.20c shows results for the frequency range between the 5th and 6th modes, from 1250 to 2000 Hz. Similar to earlier observations, the AIGSC, AIGAC, and RDI indicators are insensitive to torque levels, while FRAC, FAAC, and MDI gradually reduce as torque decreases. The higher mode shapes in this range are more sensitive to torque loosening, making these indicators more effective for detecting changes at lower torque levels. The correlation between torque loss and the FRAC, FAAC, and MDI

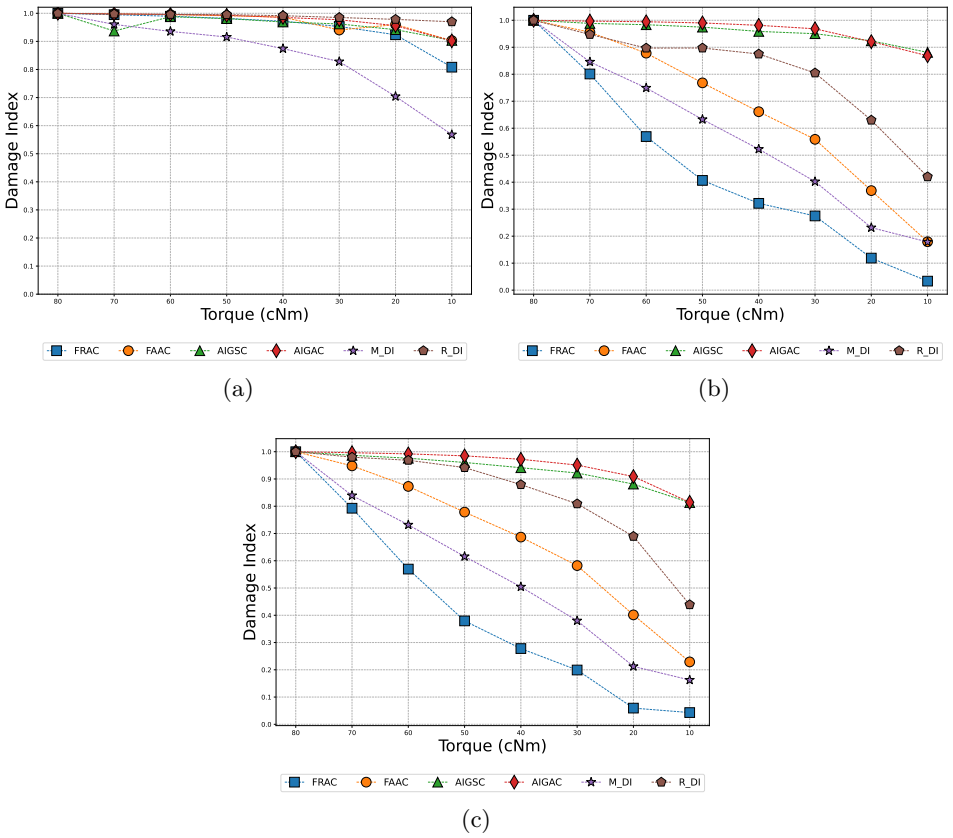


Figure 4.20: Comparison of different damage indexes methods. Results of all damage indexes at different torque levels obtained in the frequency bands: (a) 10∼2000 Hz (All signal spectrum), (b) 600∼2000 Hz (4th∼6th mode), (c) 1250∼2000 Hz (5th∼6th mode).

DIs enhances early damage detection, particularly at lower torque values. Notably, in Figures 4.20b and 4.20c, FRAC shows the highest sensitivity to torque variation, with significant drops in the DI. The difference is more pronounced, with the drop occurring earlier in these graphs. Thus, FRAC is assumed to be this application's main feature extraction technique and is employed for torque loosening detection. In the case of multiple DIs, the FRAC, FAAC, and MDI are used in the data augmentation procedure.

The experimental data is limited to 128 raw samples. Employing the entire transmissibility signal in the DIs calculation may induce a false posi-

tive in the prediction and diagnosis. Therefore, the truncated signal ranging from 1250 to 2000 Hz (Figure 4.20c) was assumed in the DIs estimation. After data processing and feature extraction, two attributes were defined, DI_1 and DI_2, each consisting of 64 FRAC DI samples. The next steps in the toque loosening estimation are data augmentation and clustering. The DI clustering is performed using the unsupervised algorithm K-means with a value of $k = 8$ (where k represents the number of classes) to identify clusters within unlabeled data. Figure 4.21a shows the clustered DI dataset returned by the K-means, comprising 64 samples belonging to one of the 8-labelled classes defining the torque levels.

The limited number of experimental data is a common issue researchers face in various fields [169]. A small amount of data, or data scarcity, is a significant challenge for applying machine learning tools involving experimental data due to its investigative nature [170]. When insufficient statistical analysis or modelling data is available, a common approach is to generate simulated data to fill the gap. In this work, we have adopted four methods to reproduce experimental data, allowing us to generate augment datasets following the system's physical characteristics and enrich the original dataset. Using statistical moments (Section 3.2.4), four additional synthetic datasets were generated with sizes corresponding to 50%, 100%, 200%, and 1000% of the original dataset. Additionally, other datasets were generated using TGAN (Section 3.2.4), Forest Diffusion (Section 3.2.4), and a dataset combining the DIs methods (Section 3.2.4). The idea behind using combined DI techniques is that the accuracy of the models depends directly on the number of features available [283] based on the original data.

Table 4.10: Description of the original dataset and augmented data.

Dataset	**Augmentation method**	**No. of samples**
Original	Experimental	2 x 64 = 128
Orignal + 50%	Statistical (Lognormal)	2 x 96 = 192
Orignal + 100%	Statistical (Lognormal)	2 x 128 = 256
Orignal + 200%	Statistical (Lognormal)	2 x 192 = 384
Orignal + 1000%	Statistical (Lognormal)	2 x 384 = 768
Orignal + TGAN	Tabular GAN	2 x 194 = 388
Orignal + Forest Diffusion	Diffusion and XGBoost	2 x 155 = 310
DI Combined	Increased features	3 x 64 = 192

Comprehensive details regarding the data fused into the original and augmented datasets are provided in Table 4.10. The table lists the number of samples in each dataset after applying various augmentation methods, specifying the number of columns, the data quantity per column, and the total number of data points. These artificial datasets, excluding the Combined DI, were added to the experimental data and grouped into classes using K-means clustering. The scatter plots in Figures 4.21(a-g) illustrate the distribution of the grouped points for the generated and experimental data, showing their correlation in two-dimensional space. The datasets were then randomly split, with 70% allocated for training and 30% for testing.

Figure 4.21a shows the original dataset, estimated using K-means clustering, with 128 samples divided into DI_1 and DI_2 attributes, where the attributes diagonal shows a perfect correlation between them. Data dispersion becomes evident for torque levels between 30 and 60 cNm. This clustering serves as an indicator of the performance and accuracy of regression algorithms. Figures 4.21b and 4.21e display the fused dataset, which includes the original data augmented with 50%, 100%, 200%, and 1000% additional random samples generated via the statistical approach. As the number of samples increases, the data clustering becomes better pronounced, and the correlation between DI and torque levels is clearly defined. Figure 4.21f is the clustering of the original data plus TGAN, and Fig. 4.21g plus Forest Diffusion. In both cases, the algorithms associate the original data with the reproduced one following the identified pattern, where the reproduced data are close to the original, maintaining the pattern of the data. Combining the

DI approach assumed different feature extraction to compose the dataset, relying only on the original data. Figure 4.21h shows the Combined DI cluster dataset comprising the combination of FRAC, FAAC, and MDI damage indices feature extractors.

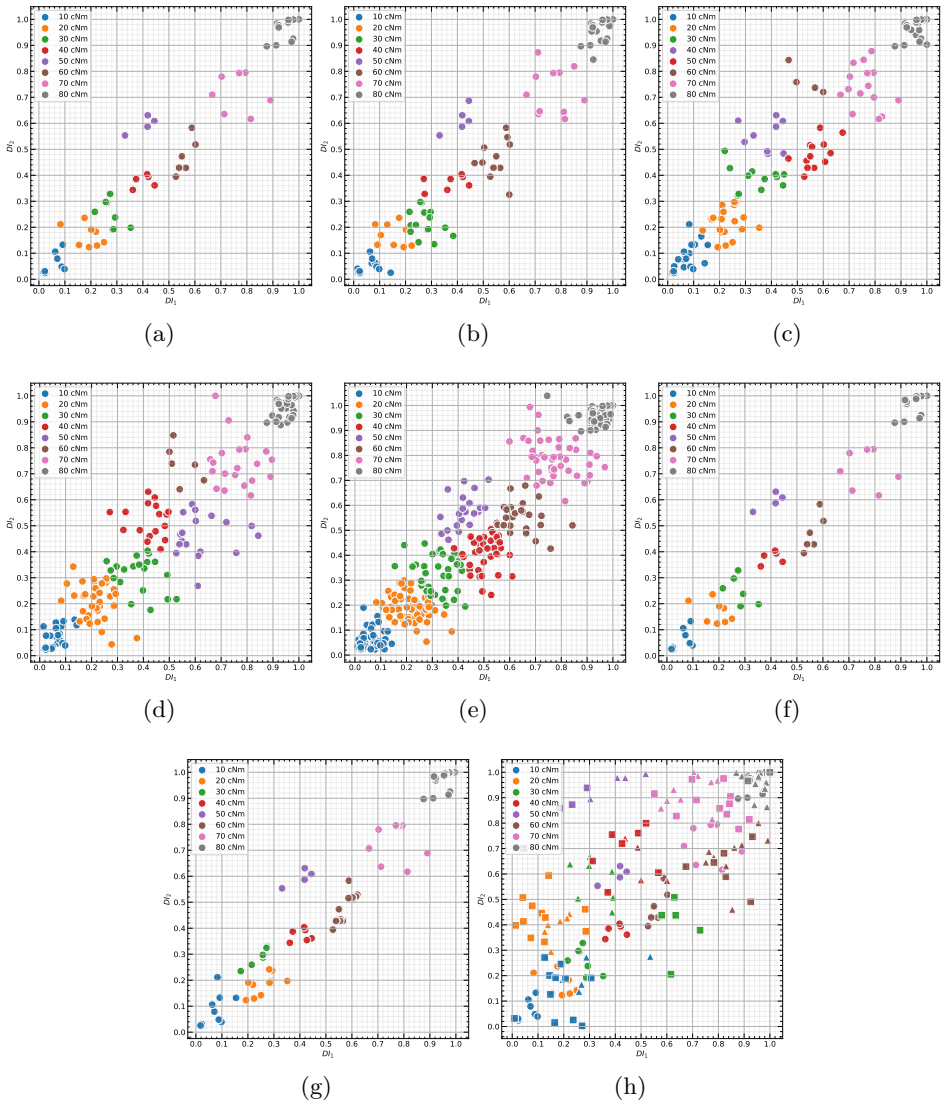


Figure 4.21: Clustered DI dataset returned by the K-means obtained from a) Orignal experimental data and fused data augmented as: b) Experimental data + 50%, c) Experimental data + 100%, d) Experimental data + 200%, e) Experimental data + 1000%, f) Experimental data + TGAN, g) Experimental data + Forest Diffusion, h) Combined DI (" ◯ " FRAC, " ⋀ " FAAC, "□" M_DI).

4.3.2 Regression machine learning for loosening torque estimation

The last step involves torque loosening and uncertainty quantification. Nine machine-learning regression algorithms are employed to estimate torque values. Among them, six are linear as Linear Regression, Lasso Regression, K-Neighbors Regressor (KNR), Decision Tree Regressor (DTR), Gradient Boosting Regressor (GBR), and Support Vector Regression with a linear kernel (SVR-linear). The remaining three are nonlinear techniques such as Support Vector Regression with a Radial Basis Function kernel (SVR-RBF), Support Vector Regression with a Polynomial kernel (SVR-Poly), and Multi-Layer Perceptron Regressor (MLP Regressor). Table 4.11 lists the hyperparameters chosen for each algorithm. All models were implemented using the open-source Scikit-learn library in Python, with simulations performed in Google Colab. The evaluation metrics for these algorithms include the coefficient of determination (R^2), MAE, MSE, and RMSE.

Table 4.11: Hyperparameters assumed for each ML algorithm.

Regression Algorithm	**Hyperparameters**
Linear Regression	fit intercept = True.
Lasso Regression	alpha = 1.0.
KNR	n_neighbors = 2.
SVR-Linear	kernel = 'linear'; gamma = 'auto'; C = 10.
SVR-RBF	kernel = 'rbf'; gamma = 'auto'; C = 10; epsilon = 0.1.
SVR-Poly	kernel = 'poly'; gamma = 'auto'; C = 10; epsilon = 0.1; degree = 3.
MLP Regressor	hidden layer sizes = 50; alpha = 0.001; solver = 'lbfgs'; learning rate = 'adaptive'.
DTR	criterion = 'squared_error'; max_depth = none.
GBR	loss = 'squared_error'; learning_rate = 0.1; n_estimators = 100; criterion = 'friedman_mse'; max_depth = 3.

The evaluation metrics for the regression algorithms are presented in Table 4.12 and Fig. 4.22. The input datasets used for regression include

the pure original dataset, the original dataset combined with 50%, 100%, 200%, and 1000% artificial data generated through statistical methods, the original dataset combined with TGAN-generated artificial data, the original plus artificial data from Forest Diffusion, and the multiples DIs. We focus on FRAC, FAAC, and MDI, given their superior performance over the other damage indices methods. The comparison of the metrics of the original dataset with those from various data augmentation scenarios provides a detailed analysis of the improvements achieved through these techniques.

Table 4.12: Comparison of evaluation metrics for regression models.

Scenario	Regression Algorithm	R^2	MAE	MSE	RMSE	Scenario	Regression Algorithm	R^2	MAE	MSE	RMSE
Original	Linear Regression	0,93	4,98	36,96	6,08	Original + 1000%	Linear Regression	0,97	3,39	17,35	4,16
	SVR-Linear	0,93	4,85	38,97	6,24		SVR-Linear	0,98	3,20	16,76	4,09
	SVR-RBF	0,96	3,56	23,80	4,88		SVR-RBF	0,98	2,38	11,43	3,38
	SVR-Poly	0,94	4,24	31,26	5,59		SVR-Poly	0,98	2,74	13,15	3,63
	MLP Regressor	0,96	3,45	22,75	4,77		MLP Regressor	0,97	3,36	17,31	4,16
	Lasso Regression	0,91	5,87	46,17	6,79		Lasso Regression	0,97	3,80	23,24	4,82
	GBR	0,96	2,95	21,44	4,63		GBR	0,99	1,59	8,08	2,84
	KNR	0,98	2,00	10,00	3,16		KNR	1,00	0,26	1,29	1,14
	DTR	0,93	4,00	40,00	6,32		DTR	0,99	0,78	9,48	3,08
Original + 50%	Linear Regression	0,96	4,53	29,97	5,47	Original + TGAN	Linear Regression	0,96	4,19	25,99	5,10
	SVR-Linear	0,96	4,01	25,99	5,10		SVR-Linear	0,96	3,83	28,82	5,37
	SVR-RBF	0,98	2,59	13,57	3,68		SVR-RBF	0,98	2,08	10,58	3,25
	SVR-Poly	0,98	3,06	16,80	4,10		SVR-Poly	0,96	4,19	26,00	5,10
	MLP Regressor	0,99	2,57	10,05	3,17		MLP Regressor	0,99	2,09	6,60	2,57
	Lasso Regression	0,94	5,46	44,02	6,63		Lasso Regression	0,94	5,04	39,19	6,26
	GBR	0,98	1,62	12,20	3,49		GBR	1,00	0,41	3,31	1,82
	KNR	0,98	1,21	11,21	3,35		KNR	1,00	0,00	0,00	0,00
	DTR	0,98	1,72	17,24	4,15		DTR	1,00	0,34	3,39	1,84
Original +100%	Linear Regression	0,98	2,79	13,64	3,69	Original + Forest Diffusion (FD)	Linear Regression	0,95	4,77	32,84	5,73
	SVR-Linear	0,98	3,05	16,76	4,09		SVR-Linear	0,95	4,41	32,55	5,71
	SVR-RBF	0,99	2,09	7,72	2,78		SVR-RBF	0,98	2,64	15,37	3,92
	SVR-Poly	0,98	2,79	13,83	3,72		SVR-Poly	0,97	2,93	18,24	4,27
	MLP Regressor	0,99	1,96	6,58	2,56		MLP Regressor	0,98	2,59	11,85	3,44
	Lasso Regression	0,97	4,02	24,44	4,94		Lasso Regression	0,95	4,19	32,18	5,67
	GBR	0,99	0,99	5,10	2,26		GBR	0,99	0,52	6,11	2,47
	KNR	1,00	0,38	1,92	1,39		KNR	1,00	0,21	2,13	1,46
	DTR	0,99	0,77	7,69	2,77		DTR	0,99	0,43	8,51	2,92
Original +200%	Linear Regression	0,98	3,40	17,93	4,23	Combined DI (cDI)	Linear Regression	0,97	3,09	15,13	3,89
	SVR-Linear	0,97	3,50	20,16	4,49		SVR-Linear	0,97	3,03	15,72	3,96
	SVR-RBF	0,98	2,72	12,60	3,55		SVR-RBF	0,96	3,67	20,36	4,51
	SVR-Poly	0,97	3,31	18,08	4,25		SVR-Poly	0,97	3,27	18,74	4,33
	MLP Regressor	0,99	2,40	10,77	3,28		MLP Regressor	0,98	2,48	10,40	3,22
	Lasso Regression	0,96	4,56	30,44	5,52		Lasso Regression	0,95	4,35	27,80	5,27
	GBR	0,99	1,52	8,81	2,97		GBR	0,99	1,10	5,84	2,42
	KNR	1,00	0,43	3,02	1,74		KNR	1,00	0,38	1,92	1,39
	DTR	0,99	0,52	5,17	2,27		DTR	0,97	1,54	15,38	3,92

The analysis of metrics presented in Table 4.12 and the plot in Fig 4.22a show that all models performed well, as indicated by the coefficient of determination (R^2). For the original data, the algorithms achieve R^2 values ranging from 0.91 to 0.98, suggesting that the models explain the data variance effectively. As the dataset is augmented through statistical moments (50%-1000%), an improvement is seen in R^2, ranging from 0.94 to 1.00, reflecting enhanced model performance and a stronger ability to handle data variance, particularly for the MLP Regressor, GBR, KNR, and DTR. The

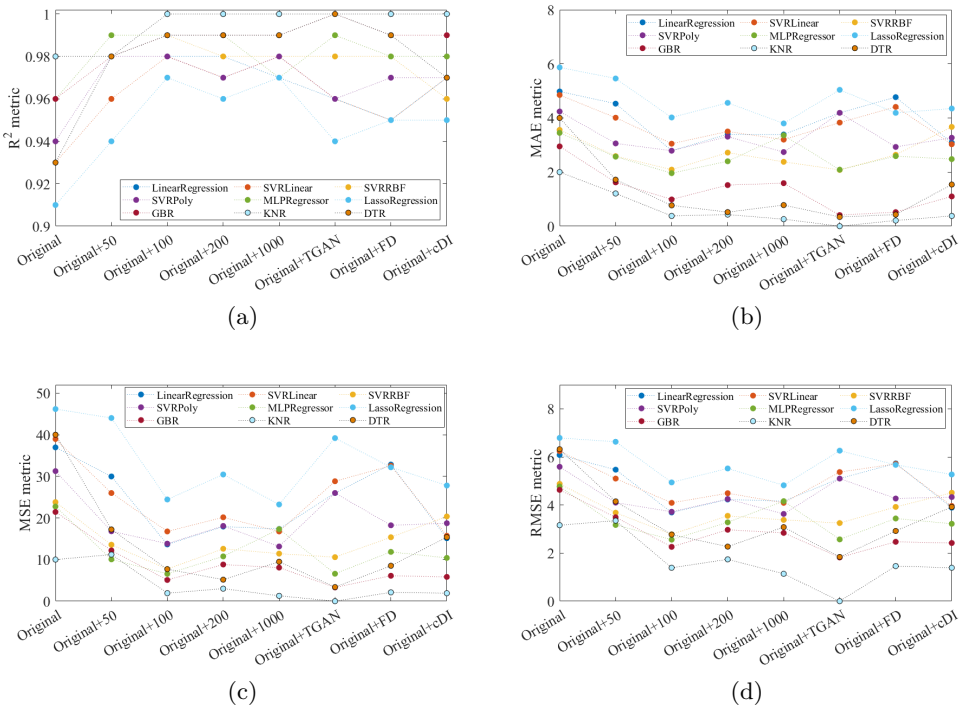


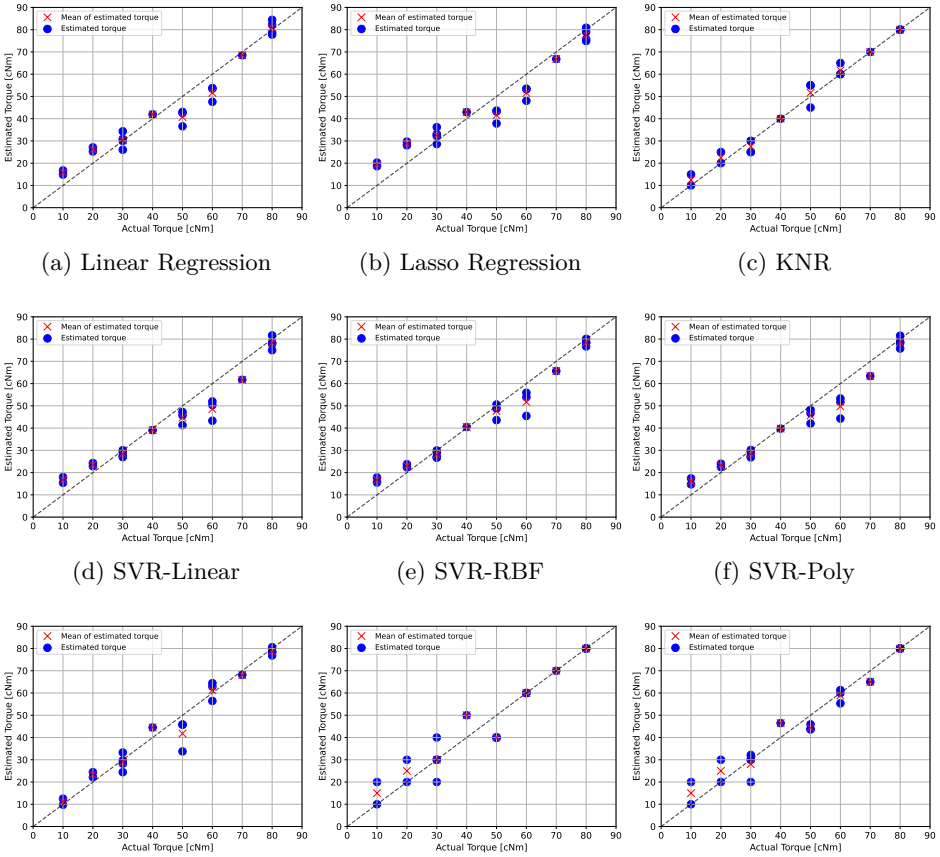
Figure 4.22: Comparison of the performance of the algorithm and the augmented data technique through the metrics a) R^2, b) MAE, c) MSE, and d) RMSE.

data augmentation techniques like TGAN and Forest Diffusion also result in significant improvements, with R^2 values ranging from 0.94 to 1.00, and some models (MLP Regressor, GBR, KNR, DTR) achieving R^2 values close to or equal to 1. The Combined DI method consistently maintains R^2 values between 0.95 and 1.00, demonstrating high performance, with GBR and KNR achieving R^2 values near or equal to 1.

For the original dataset, the algorithms achieve the MAE between 2.00 and 5.87, with KNR standing out for having the lowest MAE among all models, as plotted in Fig. 4.22b. As the data is augmented using statistical moments, the MAE decreases to a range of 0.26 to 5.46, indicating improved model accuracy and reduced error, with GBR, KNR, and DTR showing the lowest MAE values. Augmentation methods like TGAN and Forest Diffusion further reduce the MAE, achieving values between 0.00 and 4.77, with GBR,

KNR, and DTR yielding the most accurate predictions. The Combined DI method significantly improved over the original dataset, with MAE values ranging from 0.38 to 4.35, again highlighting KNR as the best performer.

The MSE shown in Fig 4.22c penalises larger errors more severely. For the original dataset, the MSE values range between 10.00 and 46.17, with KNR exhibiting the lowest value among the algorithms. When the data is augmented using the statistical method, the MSE improves, ranging from 1.29 to 44.02, with KNR again achieving the lowest values. Augmentation methods such as TGAN and Forest Diffusion lead to even greater accuracy, yielding MSE values between 0.00 and 32.84, with KNR using TGAN providing the best result. The Combined DI method also significantly improves, with MSE values ranging from 1.92 to 27.8. In particular, GBR and KNR stand out for having the lowest MSE values, demonstrating the effectiveness of these combined techniques. In Fig. 4.22d, the RMSE provides the magnitude of the error, with lower values indicating better model performance. The algorithms achieve RMSE values between 3.16 and 6.79 for the original dataset, reflecting a good fit. When the dataset is augmented using statistical moments, the RMSE significantly decreases, ranging from 1.14 to 6.63, showing improved performance compared to the original data. The MLP Regressor, GBR, KNR, and DTR algorithms have the lowest RMSE values. Augmentation techniques like TGAN and Forest Diffusion also result in better performance, with RMSE values ranging from 0.00 to 6.26, with KNR showing a perfect fit when using TGAN. The Combined DI method demonstrates RMSE values between 1.39 and 5.27, further improving over the original data. KNR and GBR achieve the lowest RMSE values, reinforcing the enhanced performance provided by combining techniques.

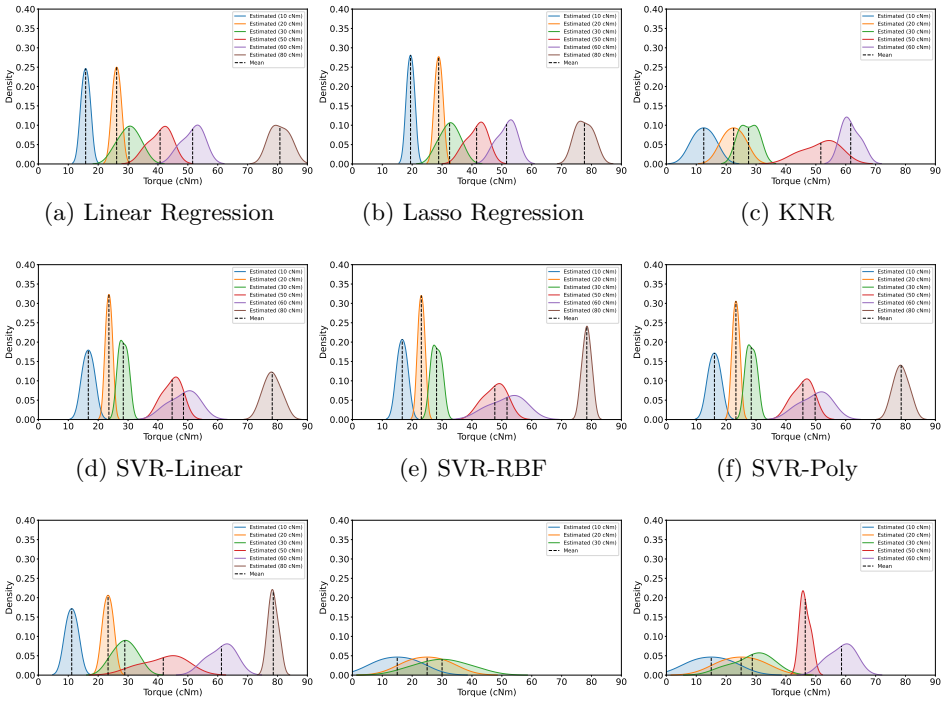


(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.23: Loosening torque estimation versus actual torque predicted from the original experimental dataset.



(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

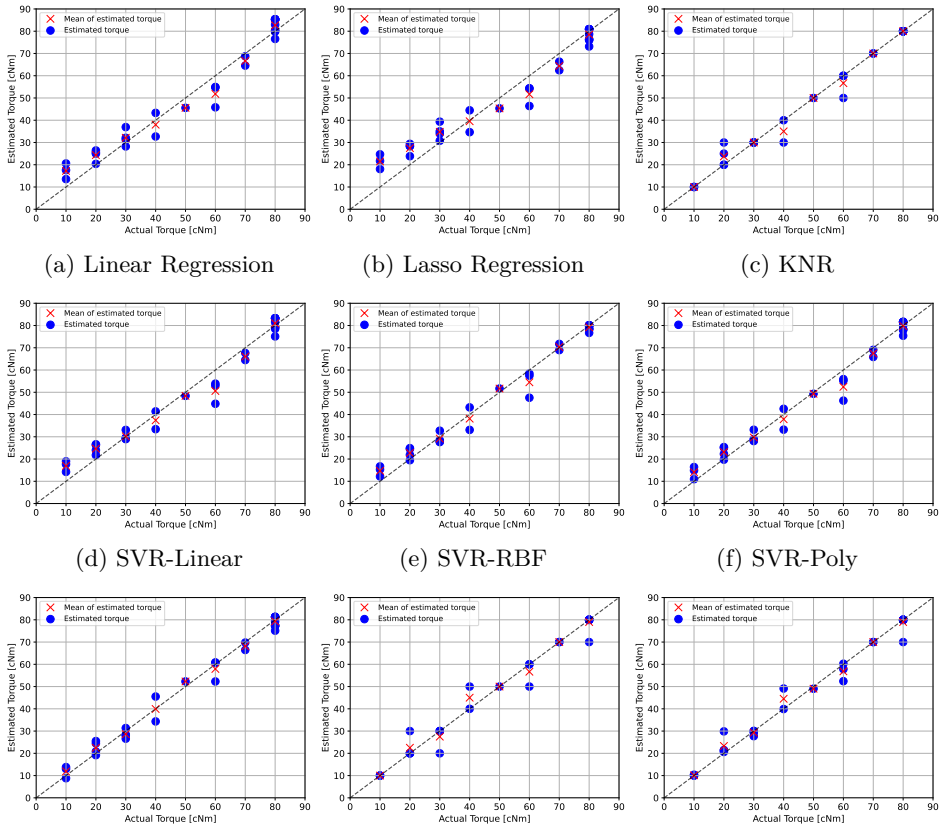
Figure 4.24: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original experimental dataset.

Table 4.13: Means and standard deviation values of the regression models performed from the original data.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Experimental	10	15.81	1.34	16.70	1.30	16.68	1.13	16.07	1.36	11.15	1.92	19.46	1.17	15.00	5.00	12.50	2.50	15.00	5.00
	20	26.18	1.32	23.59	0.72	23.06	0.73	23.26	0.76	23.31	1.60	28.86	1.19	25.00	5.00	22.50	2.50	25.00	5.00
	30	30.31	3.40	28.40	1.28	28.16	1.36	28.38	1.35	28.92	3.66	32.52	3.13	28.86	5.57	27.50	2.50	30.00	7.07
	40	41.96	0.00	39.08	0.00	40.48	0.00	39.77	0.00	44.51	0.00	42.96	0.00	42.75	0.00	40.00	0.00	50.00	0.00
	50	40.72	3.59	44.74	2.51	47.65	2.97	45.67	2.63	41.78	6.96	41.56	3.23	48.47	1.48	51.67	4.71	40.00	0.00
	60	51.61	3.48	48.54	3.77	51.74	4.51	49.75	3.92	61.25	4.25	51.58	3.06	57.63	4.17	61.67	2.36	60.00	0.00
	70	68.51	0.00	61.70	0.00	65.66	0.00	63.38	0.00	68.09	0.00	66.83	0.00	66.14	0.00	70.00	0.00	70.00	0.00
	80	80.81	3.07	78.24	2.36	78.42	1.20	78.55	2.05	78.59	1.53	77.63	2.76	80.00	0.00	80.00	0.00	80.00	0.00

In Figure 4.23(a-i), the plots compare the actual torque with the corresponding estimated values from the nine regression algorithms. The input is the K-means DI clustering extracted from the original dataset. The diagonal dashed-black line represents the perfect correlation between the actual and predicted torques for all similar graphics. At the same time, the red 'X' symbols denote the estimated mean values for each torque condition, and the

blue dots show the individual torque estimates provided by the algorithms. The graphs in Figure 4.24(a-i) illustrate the mean values and probability density functions (PDFs) of the estimated torques at various levels: 10cNm, 20cNm, 30cNm, 40cNm, 50cNm, 60cNm, 70cNm, and 80cNm. The corresponding mean values and STDs are listed in Table 4.13. Overall, the nine methods demonstrated good torque estimation accuracy, with KNR outperforming the others by producing mean torque estimates closely aligned with the correlation line and actual values. However, methods such as KNR, DTR, and GBR made it difficult to capture the PDFs for some torque levels, as their estimates were tightly clustered around the mean value.

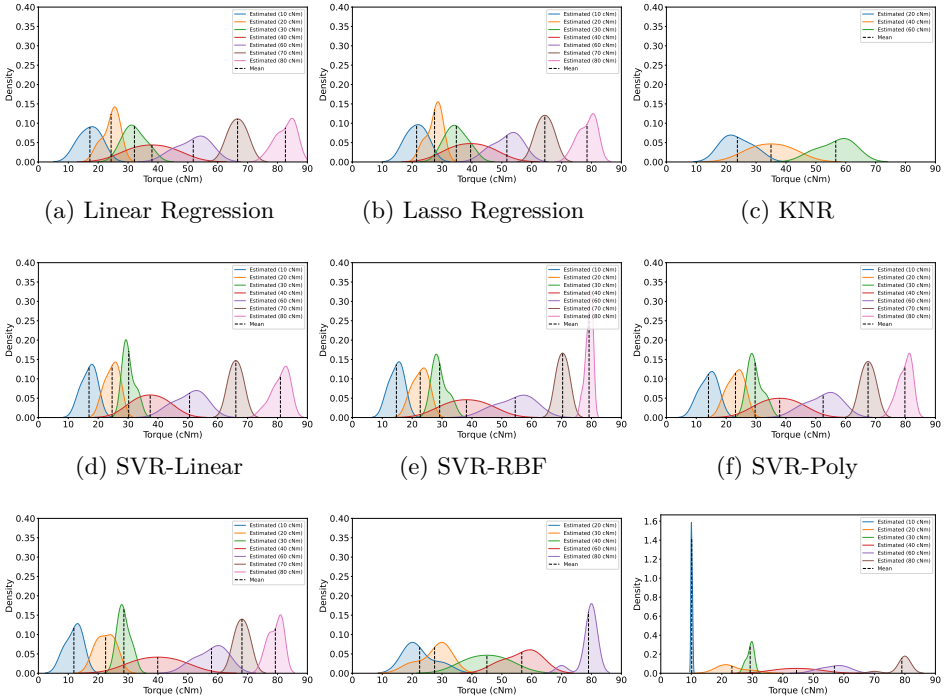


(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.25: Loosening torque estimation versus actual torque predicted from the augmented dataset: Original plus 50% of random samples given by the statistic method.



(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.26: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original plus 50% of random samples given by the statistic method.

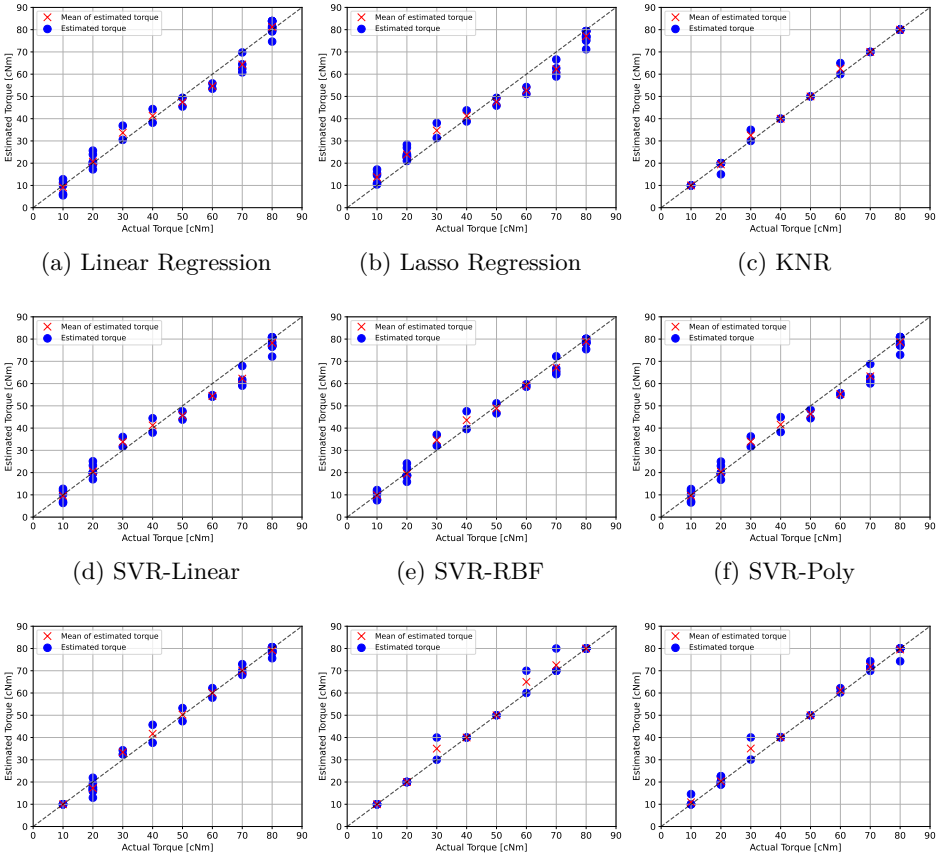
Table 4.14: Means and standard deviation values of the regression models performed from the original plus 50% of random samples given by the statistic method.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Original +50%	10	17.24	3.53	16.95	2.00	14.72	1.89	14.08	2.31	11.88	2.20	21.48	2.71	10.13	0.18	10.00	0.00	10.00	0.00
	20	24.33	2.69	24.67	1.97	22.70	2.20	23.11	2.30	22.49	2.59	27.49	2.15	23.32	3.80	23.75	4.15	22.50	4.33
	30	32.14	3.60	30.21	1.68	29.19	2.07	29.68	2.04	28.60	1.74	34.78	3.08	29.29	0.97	30.00	0.00	27.50	4.33
	40	38.00	7.51	37.41	4.00	38.15	5.06	37.89	4.68	39.92	5.59	39.55	4.91	45.41	5.51	35.00	5.00	45.00	5.00
	50	45.57	0.00	48.39	0.00	51.68	0.00	49.34	0.00	52.30	0.00	45.33	0.00	49.32	0.00	50.00	0.00	50.00	0.00
	60	51.82	5.22	50.58	4.07	54.48	4.92	52.44	4.39	57.92	3.99	51.68	3.74	56.89	2.57	56.67	4.71	56.67	4.71
	70	66.65	2.96	66.06	1.58	70.34	1.40	67.47	1.61	68.11	1.67	64.38	1.93	70.00	0.00	70.00	0.00	70.00	0.00
	80	82.65	3.20	80.98	2.70	79.18	1.11	79.80	2.10	79.29	2.21	78.50	2.72	78.99	3.00	80.00	0.00	79.00	3.00

Figure 4.25(a-i) shows the correlation of the actual versus the predicted torque using nine regression algorithms. The input dataset provided by K-means consists of the original dataset augmented by 50% of its size, with random samples generated based on the original data's first and second statistical moments and a Lognormal distribution. The PDFs in Figs.4.26(a-i) illustrate the estimated torque mean values and their distribution densi-

ties, with the corresponding means and standard deviations displayed in Table4.14. Most regression methods accurately estimated the torque levels, except for the Linear and Lasso regressors, which miscalculated some torque mean. The other techniques accurately predicted the torque values close to the actual values. KNR, however, toiled to capture the PDFs at some torque levels, as its estimates were tightly clustered around the mean value.

Figure 4.27(a-i) shows graphs correlating the actual to estimated torque values with the regression algorithms. The input dataset consists of the original data, augmented by 100% of the original dataset, with random samples calculated using the original data's first and second statistical moments and a Lognormal distribution. The PDFs in Figs. 4.28(a-i) illustrate the mean estimated torque values and their distribution. The corresponding mean values and STD are provided in Table 4.15. Most regression methods accurately estimated the torque levels, except the Linear, Lasso, and SVR Linear regressors, which miscalculated a few of the torque's mean. The remaining methods accurately predicted the torque values, closely aligning with the actual values. The GBR captured a few PDFs only as its estimates were tightly clustered around the mean.

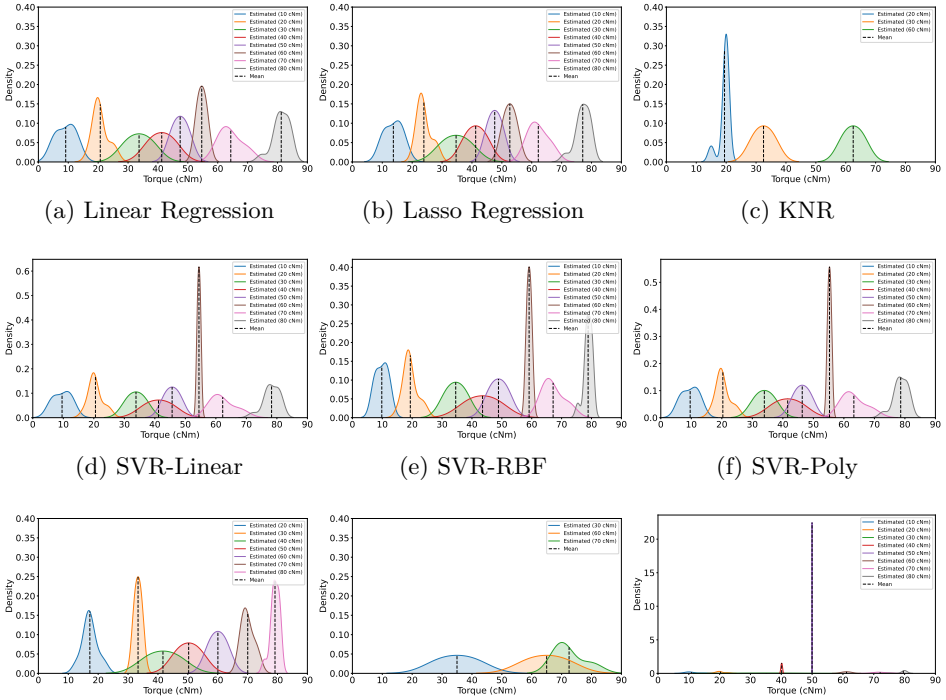


(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.27: Loosening torque estimation versus actual torque predicted from the augmented dataset: Original plus 100% of random samples given by the statistic method.



(a) Linear Regression (b) Lasso Regression (c) KNR

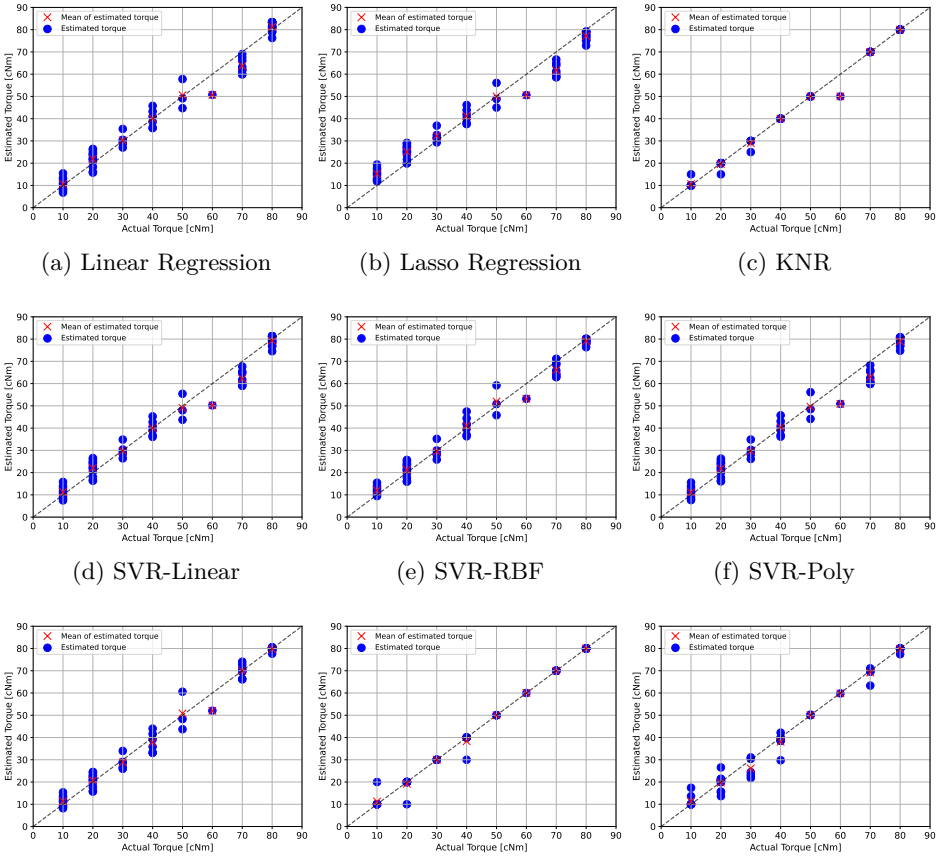
(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.28: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original plus 100% of random samples given by the statistic method.

Table 4.15: Means and standard deviation values of the regression models performed from the original plus 100% of random samples given by the statistic method.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Experimental +100%	10	9.14	3.18	9.59	2.56	9.83	1.85	9.63	2.43	9.99	0.03	13.70	2.61	10.90	1.84	10.00	0.00	10.00	0.00
	20	20.70	2.54	20.54	2.19	19.38	2.24	20.33	2.21	17.22	2.34	23.91	2.19	20.30	1.34	19.44	1.57	20.00	0.00
	30	33.68	4.53	33.83	2.21	34.57	2.47	33.93	22.31	33.34	0.93	34.69	3.38	35.18	5.10	32.50	2.50	35.00	5.00
	40	41.27	4.34	41.21	3.23	43.55	4.00	41.58	3.35	41.68	4.05	41.25	2.50	40.09	0.12	40.00	0.00	40.00	0.00
	50	47.43	2.79	45.66	1.86	48.87	2.26	46.35	1.94	50.28	2.96	47.62	1.75	49.98	0.01	50.00	0.00	50.00	0.00
	60	54.66	1.68	54.46	0.38	59.12	0.58	55.29	0.42	60.06	2.15	52.68	1.55	61.16	0.98	62.50	2.50	65.00	5.00
	70	64.37	3.90	62.24	3.41	67.16	3.09	63.27	3.34	70.04	1.83	62.20	2.86	71.67	1.28	70.00	0.00	72.50	4.33
	80	81.25	2.71	78.26	2.46	78.87	1.29	78.62	2.25	79.17	1.41	77.07	2.31	79.49	1.71	80.00	0.00	80.00	0.00

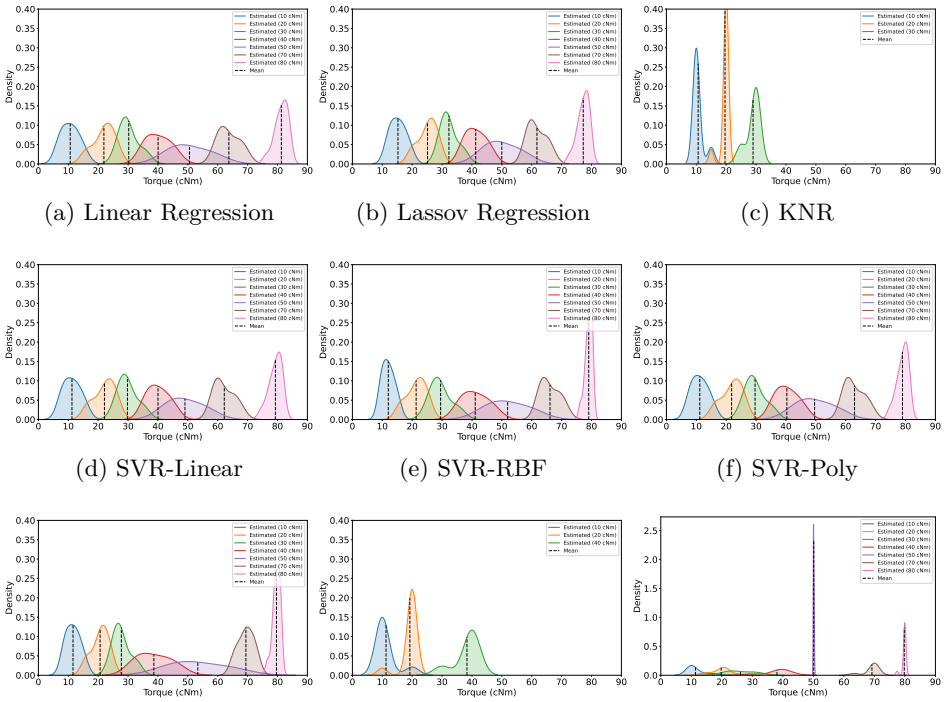


(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.29: Loosening torque estimation versus actual torque predicted from the augmented dataset: Original plus 200% of random samples given by the statistic method.



(a) Linear Regression (b) Lassov Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.30: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original plus 200% of random samples given by the statistic method.

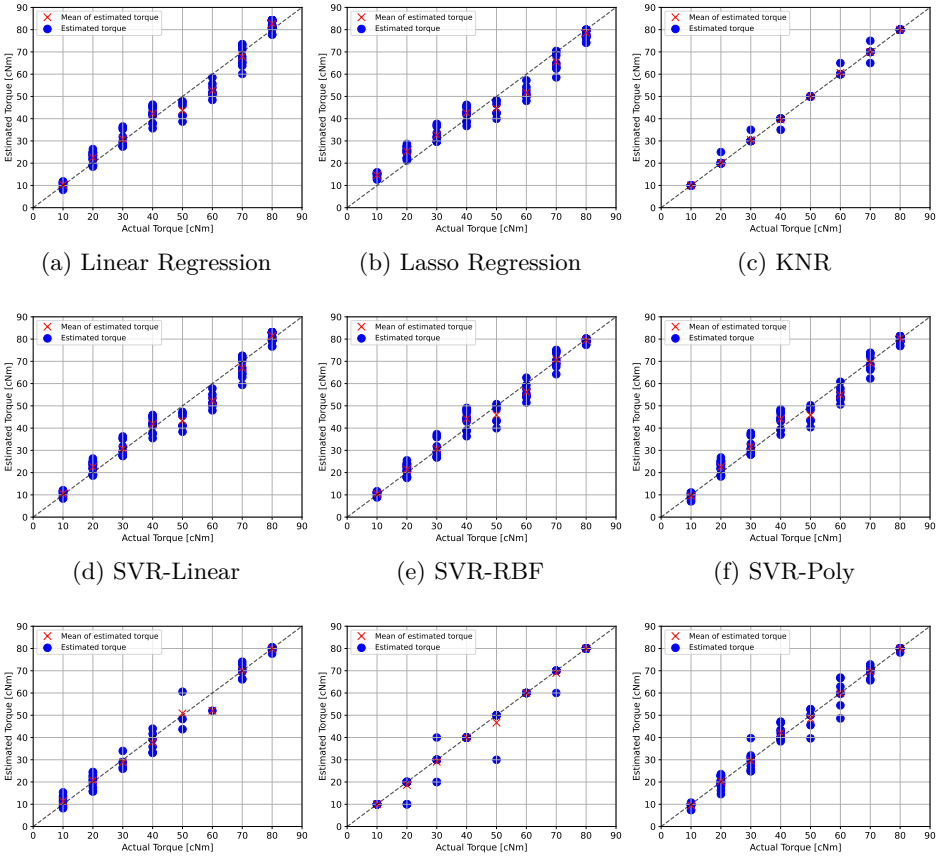
Table 4.16: Means and standard deviation values of the regression models performed from the original plus 200% of random samples given by the statistic method.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Experimental +200%	10	10.65	2.97	11.24	2.70	12.01	1.94	11.17	2.57	11.41	2.31	15.25	2.45	11.37	2.58	10.63	1.65	11.25	3.31
	20	21.92	3.47	22.10	3.28	21.33	3.11	21.82	3.29	20.75	2.79	25.14	2.95	19.72	3.10	19.62	1.33	19.23	2.66
	30	30.23	3.16	29.80	2.84	29.51	3.12	29.66	2.92	28.83	2.83	32.30	2.52	26.12	3.86	29.00	2.00	30.00	0.00
	40	40.05	4.03	40.03	3.25	41.10	3.97	40.30	3.36	37.84	4.11	41.25	3.10	38.05	3.91	40.00	0.00	38.33	3.73
	50	50.55	6.65	49.04	4.83	51.94	5.53	49.58	4.96	50.86	7.13	49.98	4.59	50.12	0.11	50.00	0.00	50.00	0.00
	60	50.72	0.00	50.20	0.00	53.18	0.00	50.87	0.00	52.04	0.00	50.57	0.00	59.27	0.00	50.00	0.00	60.00	0.00
	70	63.72	3.49	62.19	3.09	66.13	2.95	62.94	3.04	69.96	2.60	61.75	2.84	69.20	2.26	70.00	0.00	70.00	0.00
	80	81.29	2.17	79.30	2.03	79.07	1.09	78.99	1.80	79.82	0.78	77.24	1.84	79.79	0.66	80.00	0.00	80.00	0.00

Figure 4.29(a-i) shows the graphs correlating the actual and predicted torque value estimates from nine regression algorithms. The input dataset, provided by K-means, consists of the original data augmented by 200%, using random samples generated based on the original data's first and second statistical moments and a Lognormal distribution. The PDFs in Figs.4.30(a-

i) illustrate the estimated mean torque values and their distribution densities, with the corresponding means and standard deviations shown in Table4.16. KNR and DTR provided torque values that closely matched the correlation line. In contrast, the other regression methods also delivered accurate torque values estimation, therefore with slight deviations, particularly around 60cNm and 70cNm. However, DTR and KNR could not define the PDF, as their predictions were tightly clustered around the mean value.

Figure 4.31(a-i) presents the graphs correlating the actual and estimated torque values by the nine regression algorithms, Figs.4.32(a-i) shows the mean estimated torque values and their distribution densities, and Table4.17 provides the corresponding mean values and standard deviations. The input dataset consists of the original data augmented by 1000%, using random samples calculated based on the first and second statistical moments of the original data and a Lognormal distribution. Most regression methods accurately estimated the torque levels, with KNR and DTR showing tight clustering around the mean value. Overall, all algorithms predicted the torque values with high accuracy.

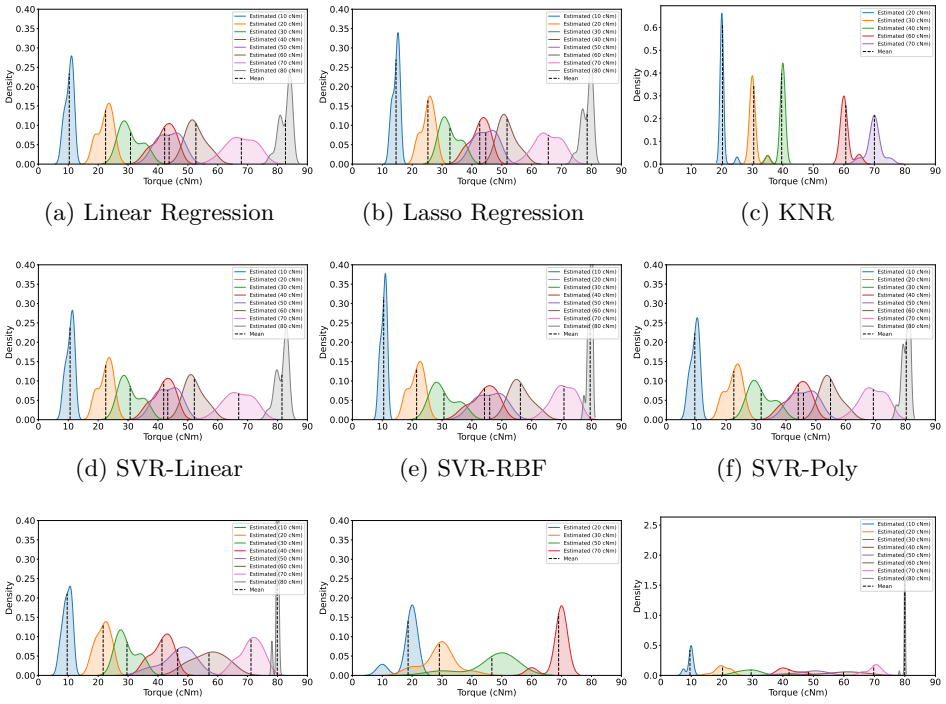


(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.31: Loosening torque estimation versus actual torque predicted from the augmented dataset: Original plus 1000% of random samples given by the statistic method.



(a) Linear Regression (b) Lasso Regression (c) KNR

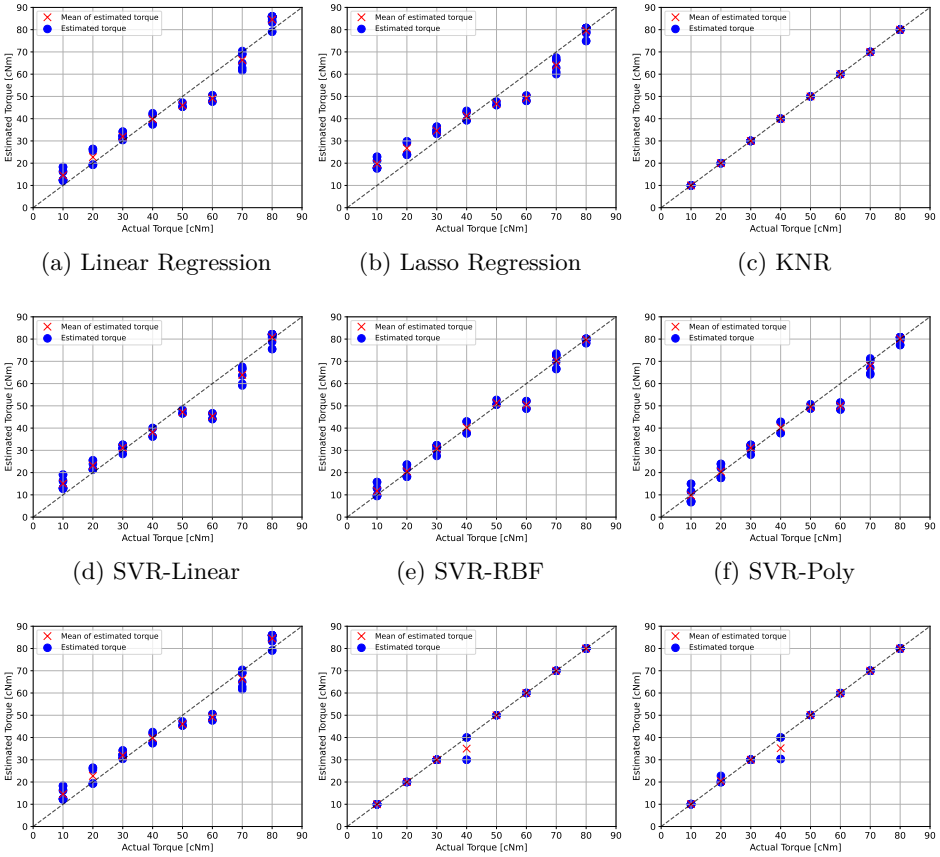
(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.32: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original plus 1000% of random samples given by the statistic method.

Table 4.17: Means and standard deviation values of the regression models performed from the original plus 1000% of random samples given by the statistic method.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Experimental +1000%	10	10.33	1.30	10.61	1.22	10.48	0.91	9.50	1.30	10.59	0.75	14.61	1.06	9.58	1.01	10.00	0.00	10.00	0.00
	20	22.48	2.35	22.54	2.25	21.48	2.34	22.56	2.51	22.37	2.30	25.26	2.07	20.26	2.25	20.23	1.04	18.64	3.43
	30	30.82	3.49	30.70	3.26	30.63	3.87	31.71	3.65	30.72	3.34	32.64	2.96	29.61	4.04	30.45	1.44	29.09	5.14
	40	42.24	3.42	41.91	3.22	44.19	3.93	44.11	3.51	42.19	3.29	42.61	2.91	42.03	3.00	39.62	1.33	40.00	0.00
	50	43.74	3.72	43.28	3.31	45.93	4.03	45.86	3.68	43.66	3.40	44.62	3.13	48.33	4.60	50.00	0.00	46.67	7.45
	60	52.69	3.21	52.17	2.93	56.20	3.31	54.91	3.02	52.67	3.01	51.75	2.72	59.94	5.77	60.63	1.65	60.00	0.00
	70	67.91	4.53	67.03	4.21	70.79	3.39	69.25	3.68	67.92	4.31	65.55	3.80	69.76	2.14	70.00	2.24	69.00	3.00
	80	82.67	2.00	81.50	1.93	79.56	0.74	80.25	1.24	82.73	1.97	78.55	1.73	79.93	0.31	80.00	0.00	80.00	0.00

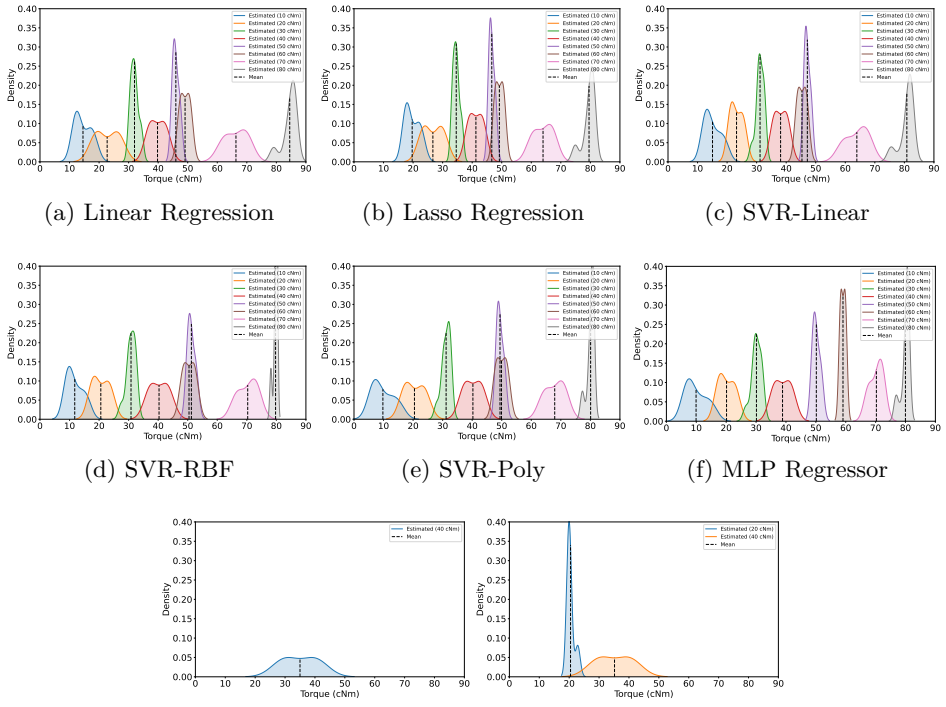


(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.33: Loosening torque estimation versus actual torque predicted based on an original plus augmented samples from TGAN.



(a) Linear Regression (b) Lasso Regression (c) SVR-Linear

(d) SVR-RBF (e) SVR-Poly (f) MLP Regressor

(g) DTR (h) GBR

Figure 4.34: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original plus augmented sample from TGAN.

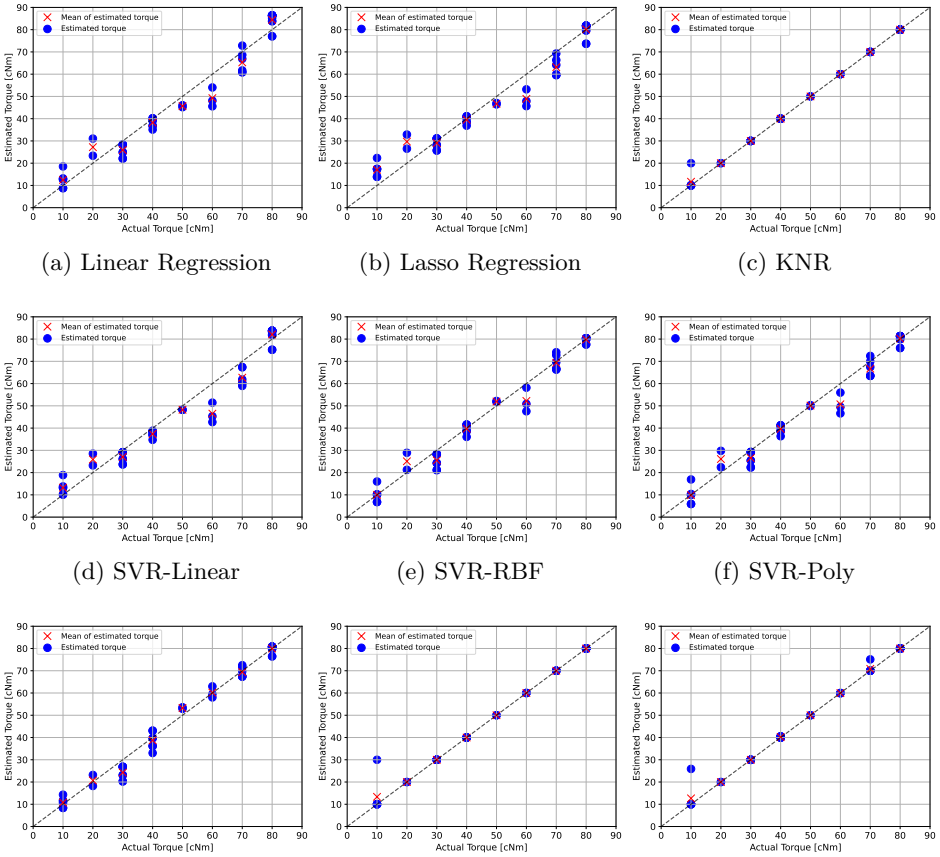
Table 4.18: Means and standard deviation values of the regression models performed from the original plus augmented sample from TGAN.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Experimental +TGAN	10	14.55	2.69	15.07	2.84	11.76	2.67	9.78	3.52	14.57	2.68	19.69	2.30	10.05	0.05	10.00	0.00	10.00	0.00
	20	22.73	3.74	23.20	2.01	20.58	2.74	20.46	3.16	22.73	3.74	26.70	3.16	20.44	1.14	20.00	0.00	20.00	0.00
	30	31.99	1.30	31.16	1.28	30.81	1.51	31.18	1.42	31.99	1.30	34.60	1.10	30.02	0.03	30.00	0.00	30.00	0.00
	40	39.78	2.70	38.11	2.19	40.29	3.07	40.24	2.91	39.78	2.70	41.26	2.30	35.17	5.60	40.00	0.00	35.00	5.77
	50	45.98	1.09	47.16	0.99	51.24	1.26	49.42	1.13	45.98	1.09	46.63	0.93	50.11	0.06	50.00	0.00	50.00	0.00
	60	49.10	1.61	45.34	1.48	50.44	1.95	49.91	1.79	49.10	1.61	49.20	1.38	59.92	0.10	60.00	0.00	60.00	0.00
	70	66.33	3.68	63.91	3.65	70.31	3.12	67.93	3.14	66.33	3.68	63.98	3.15	70.01	0.00	70.00	0.00	70.00	0.00
	80	84.53	2.33	80.84	2.28	79.74	0.65	80.09	1.19	84.53	2.33	79.55	1.99	79.98	0.00	80.00	0.00	80.00	0.00

The graphs of Figure 4.33(a-i) correlate the actual torque values with the nine regression algorithms' predictions, while Figs.4.34(a-h) shows the mean estimated torque values and their distribution densities. Table4.18 provides the corresponding torque mean values and standard deviations. The input dataset consists of the original data augmented with samples from the TGAN approach. Most regression methods accurately estimated

the torque levels, showing tight clustering around the mean value. The KNR presents the perfect correlation for all torque values, followed by DTR and GBR, misleading only 40cNm torque.

Another generative network employed to augment data is the Forest Diffusion. Figure 4.35(a-i) shows the correlation of the actual torque values with the estimated by the regression algorithms, Figs.4.36(a-i) exhibits the mean estimated torque values and their distribution densities, and Table 4.19 provides the corresponding torque's mean and STD values. The input dataset consists of the original data augmented with samples provided by the Forest Diffusion approach. Most regression methods accurately estimated the torque levels showing tight clustering around the mean value. The KNR, DTR, and GBR precisely predicted the mean torque values with a tight data cluster. The other method estimated the mean torque with a certain dispersion, indicating the variability in the prediction.

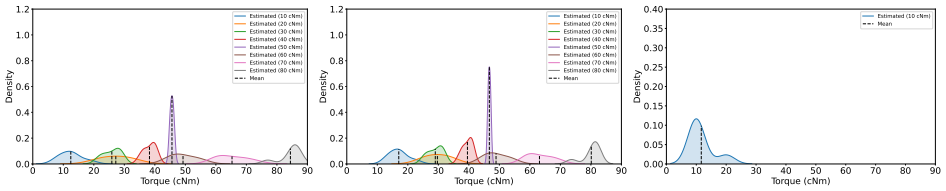


(a) Linear Regression (b) Lasso Regression (c) KNR

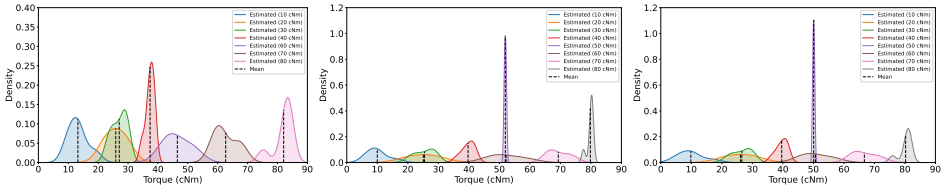
(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

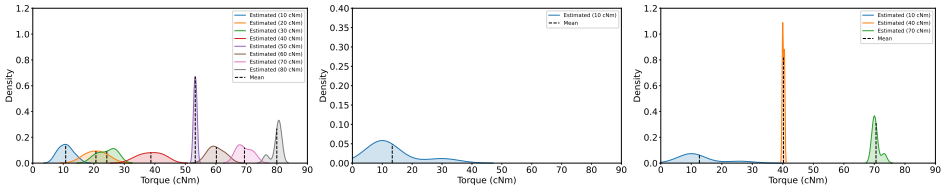
Figure 4.35: Loosening torque estimation versus actual torque predicted based on an original plus augmented samples from Forest Diffusion.



(a) Linear Regression (b) Lasso Regression (c) KNR



(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

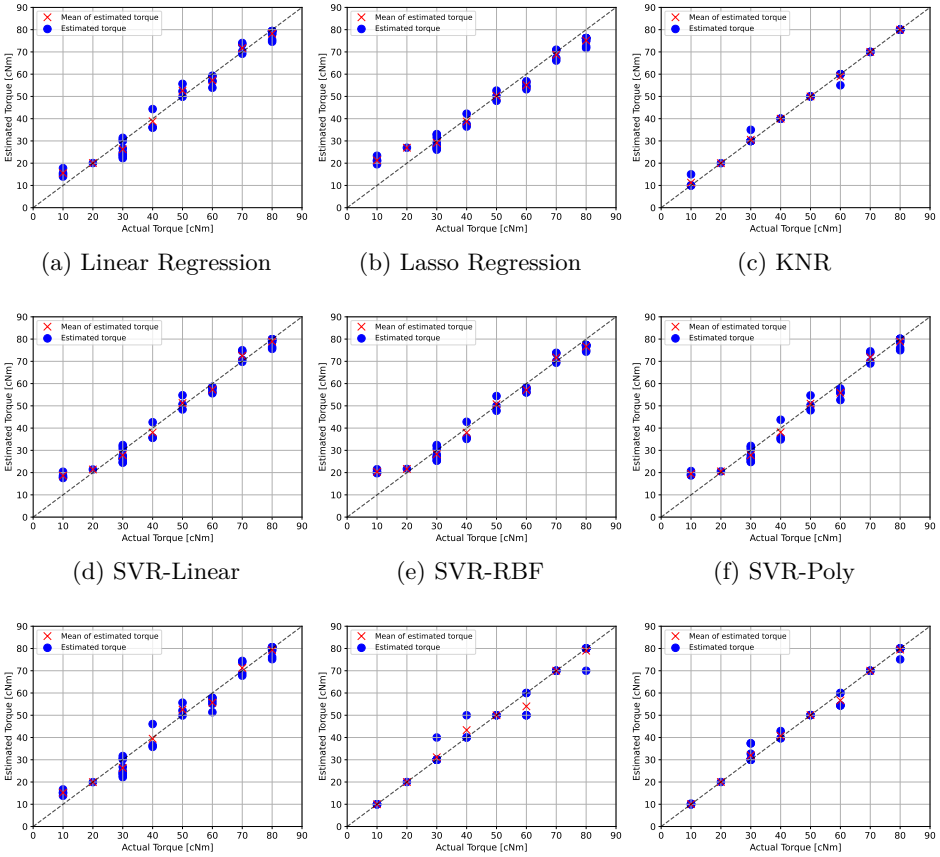


(g) MLP Regressor (h) DTR (i) GBR

Figure 4.36: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original plus augmented sample from Forest Diffusion.

Table 4.19: Means and standard deviation values of the regression models performed from the original plus augmented sample from Forest Diffusion.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Experimental + Forest Diffusion	10	12.46	3.60	13.22	3.21	9.99	3.37	9.85	4.04	10.78	2.26	17.04	3.11	12.66	6.48	11.67	4.08	13.33	8.16
	20	27.21	5.45	25.88	3.75	25.08	5.42	26.10	5.21	20.70	3.51	29.69	4.50	19.99	0.00	20.00	0.00	20.00	0.00
	30	25.91	2.77	27.04	2.45	25.49	3.19	26.57	3.08	24.27	2.94	29.01	2.39	29.98	0.03	30.00	0.00	30.00	0.00
	40	38.24	2.00	37.37	1.41	39.74	2.09	39.63	1.87	38.71	3.76	39.54	1.65	40.21	0.32	40.00	0.00	40.00	0.00
	50	45.61	0.62	48.23	0.02	51.95	0.34	50.11	0.30	53.25	0.49	46.72	0.44	50.00	0.04	50.00	0.00	50.00	0.00
	60	49.24	4.34	46.49	4.48	52.20	5.43	50.70	4.78	60.17	2.54	48.91	3.87	59.95	0.09	60.00	0.00	60.00	0.00
	70	65.30	4.97	62.66	3.76	69.32	3.44	66.71	3.87	69.36	2.32	63.14	4.12	70.60	1.47	70.00	0.00	70.00	0.00
	80	84.43	3.42	82.07	3.13	79.82	1.02	80.24	1.95	79.97	1.59	80.12	2.97	79.99	0.00	80.00	0.00	80.00	0.00

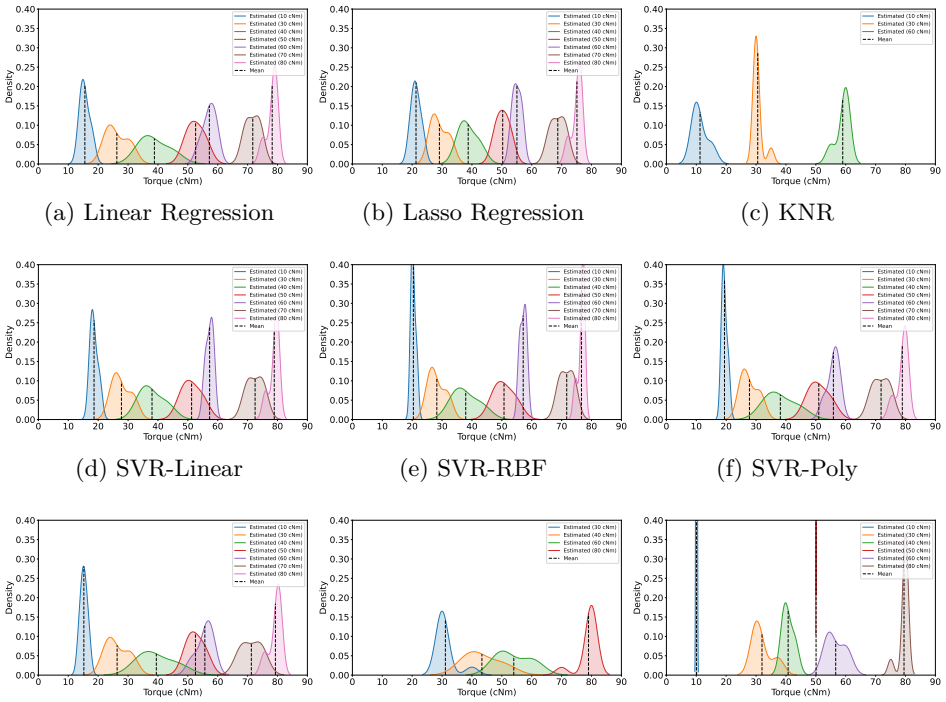


(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.37: Loosening torque estimation versus actual torque predicted based on an original plus augmented samples from Combined DI.



(a) Linear Regression (b) Lasso Regression (c) KNR

(d) SVR-Linear (e) SVR-RBF (f) SVR-Poly

(g) MLP Regressor (h) DTR (i) GBR

Figure 4.38: Probability density function of the estimated torque predicted with clustering data from the regression algorithms based on the original plus augmented sample from Combined DI.

Table 4.20: Means and standard deviation values of the regression models performed from the original plus augmented sample from Combined DI.

Scenario	Torque	Linear Regr.		SVR-Linear		SVR-RBF		SVR-Poly		MLP Regr.		Lasso Regr.		GBR		KNR		DTR	
		Mean	STD	Medias	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Experimental + Combined DI	10	15.55	1.62	18.62	1.27	20.40	0.79	19.44	0.90	13.53	4.99	21.28	1.58	10.05	0.15	11.25	2.50	10.00	0.00
	20	20.06	0.00	21.40	0.00	21.65	0.00	20.55	0.00	19.10	0.00	26.96	0.00	20.00	0.00	20.00	0.00	20.00	0.00
	30	26.25	3.50	27.79	2.98	28.25	2.70	27.78	2.76	30.38	3.17	29.11	2.73	31.88	2.84	30.56	1.67	31.11	3.33
	40	38.81	4.77	38.02	4.01	7.92	4.27	38.12	4.90	39.27	3.44	38.74	3.08	40.73	1.87	40.00	0.00	43.33	5.77
	50	52.61	2.92	51.26	3.24	50.73	3.37	50.98	3.41	50.63	2.34	50.29	2.30	50.07	0.02	50.00	0.00	50.00	0.00
	60	57.20	2.18	57.29	1.23	57.15	1.05	55.87	2.01	56.03	0.83	55.05	1.48	57.05	2.69	59.00	2.24	54.00	5.48
	70	71.73	2.37	72.51	2.65	71.71	2.31	71.83	2.76	71.20	3.43	68.72	2.42	70.00	0.00	70.00	0.00	70.00	0.00
	80	78.22	1.78	78.91	1.61	76.64	1.10	78.93	1.91	79.17	1.87	75.13	1.68	79.29	2.23	80.00	0.00	79.00	3.16

The graphs of Figure 4.33(a-i) correlate the actual torque values with the estimated ones by the nine regression algorithms, Figs.4.34(a-h) show the mean estimated torque values and their distribution densities, and Table4.18 provides the corresponding mean values and standard deviations. The input dataset consists of the original data augmented with samples provided by the Combined DI approach. In the other approaches, the dataset

has its tabular rows increased by the new artificial data generated by the other augmentation techniques. In the case of the Combined DI, the tabular dataset increases in columns, and more information based strictly on the experimental data is employed. Most regression methods accurately estimated the torque levels, showing tight clustering around the mean value with a small standard deviation. The KNR presents the perfect correlation for all torque values.

In general, the algorithms' performance improved with data augmentation. The GBR, KNR, MLP regressor and DTR machine learning algorithms consistently outperformed others across all metrics, demonstrating the best results for torque loosening estimation. This was evident through their lower dispersion, while other methods could estimate torque but with a higher dispersion, as indicated by the PDFs and calculated STDs of the torque values. The four data augmentation strategies, Statistical, TGAN, Forest Diffusion, and Combined DI, effectively increased the dataset's size and quality, ultimately enhancing the performance of the regression algorithms. This suggests that employing diverse data augmentation techniques can significantly maximise the efficiency of regression models. Among these, the statistical approach was the fastest and most intrusive in increasing the data size. However, minimal sampling is required to ensure effective uncertainty quantification and information of the original dataset.

TGAN and Forest Diffusion offer advantages over the statistical approach in data augmentation by preserving the data patterns, as they generate data that aligns closely with the original characteristics. These methods operate autonomously based on GANs, leaving little user control over the process since the neural networks handle reproduction. The combined DI method, which relies solely on data-driven techniques, also proved highly effective in improving ML regression estimations. Our findings indicate that the proposed regression algorithms, enriched with data augmentation techniques, can accurately estimate torque levels using raw vibration spectra as input. Additionally, the models provided valuable uncertainty information, allowing us to quantify the variability in the torque estimations.

Final remarks

The study proposed a data-driven regression machine learning framework for estimating bolt torque loosening using raw vibration spectra from experimental tests, enhanced with data augmentation techniques. The approach effectively captured torque variations while addressing aleatory uncertainty from assembly variability by employing nine supervised regression models and leveraging damage indices such as FRAC, FAAC, and MDI for feature extraction within selected frequency bands. Integrating a virtual sensor for data augmentation improved the robustness of torque estimation, mitigating challenges related to limited datasets in structural health monitoring. Results demonstrated high accuracy, with MLP Regressor, KNR, GBR, and DTR outperforming other models. The findings highlight the effectiveness of combining data augmentation with ML regression techniques, offering a reliable, experimentally driven methodology for bolt integrity assessment without requiring numerical models.

4.4 Chapter final remarks

This chapter presented the methodology and machine learning-based framework developed for detecting bolt loosening, estimating torque in bolted joints, and monitoring the operational conditions of the Aventa wind turbine. The proposed approach leveraged supervised regression and classification models, feature extraction techniques, and data augmentation to enhance model robustness and reliability in structural health monitoring.

For torque estimation, supervised regression models—particularly MLP Regressor, KNR, GBR, and DTR, achieved high accuracy, benefiting from data augmentation techniques that improved dataset quality and addressed the challenges of limited experimental data. Feature extraction based on damage indices such as FRAC, FAAC, and MDI proved effective in tracking torque variations. Additionally, unsupervised learning was applied to explore inherent data patterns, supporting the classification and regression models.

In wind turbine monitoring, the SHM-ML subroutine effectively identified fault conditions, with SVM emerging as the most robust classification algorithm. It achieved perfect binary classification performance and distinguished multiple operational states, particularly along the Zs direction. These findings reinforce the potential of machine learning for real-time structural health monitoring, providing a data-driven approach to diagnosing faults without relying on numerical models.

Despite the promising results, challenges remain, particularly in handling nonlinearities and discontinuities in regression modelling. Future work will expand the dataset, explore nonlinear regression techniques, and integrate physics-informed learning algorithms to enhance model interpretability and generalization. The proposed framework lays a strong foundation for intelligent monitoring systems in bolted structures and wind turbine components, contributing to advancements in predictive maintenance and structural integrity assessment.

5 Conclusion

The studies proposed in this thesis were to develop a methodology for monitoring systems and structures based on their dynamic response using SHM and ML techniques. This final chapter aims to recapitulate the main findings presented throughout this research, bringing together the information discussed in the previous chapters, and consolidating the conclusions reached. In addition, possible directions for future research are suggested.

In this thesis, a methodology for monitoring structures based on their dynamic response was developed using SHM and ML techniques. The focus was on recognizing and detecting damage in bolted structures, taking into account the influence of assembly and disassembly of the structure during experimental tests. To deal with variability in raw spectral data and noise, ML classifiers were designed to give robustness to the methodology. In addition, an approach was proposed to simplify the analysis by eliminating the need to evaluate the most sensitive features for extraction, thus introducing the concept of FRAC in obtaining the damage index, which plays a crucial role in the methodology of our algorithm. Despite the limitation of the data set, which can impact the results, the proposed methodology has shown potential for applications in structure monitoring. The limitation was subsequently incorporated into the algorithm developed.

Two approaches have been proposed for quantifying damage to structures. The first was the development of a virtual sensor in SHM-ML for data augmentation through the fusion of physical and synthetic data in order to increase the reliability and applicability of the proposed methodology. The second consists of developing a methodology involving ML algorithms for quantifying damage associated with process uncertainty. The methodology was applied to a bolted joint to estimate the loosening of the bolt torque using raw vibration spectra and the uncertainty associated with the assembly in the experimental tests. The integration of a virtual sensor for data augmentation improved the robustness of torque estimation, mitigating the challenges related to limited data sets in SHM. The approach effectively cap-

tured torque variations by addressing the random uncertainty of assembly variability, employing nine supervised regression models, and using damage indices for feature extraction within selected frequency bands. The findings highlight the effectiveness of combining data augmentation with ML regression techniques, offering a reliable, experimentally driven methodology for assessing bolt integrity without requiring numerical models.

A performance evaluation of the data-driven SHM-ML methodology was proposed with experimental data tested on an operational *in-situ* wind turbine. The methodology used the unsupervised k-means clustering technique to classify and group data into homogeneous clusters, allowing the identification of patterns without predefined labels. Two types of classification were applied for fault detection: binary and multiclass. The results of binary classification achieved perfect metrics, reflecting its ability to distinguish between normal conditions and faults in different data orientations. In multiclass classification, the approach was able to identify multiple operating states and different levels of damage with high precision, excelling in discriminating between classes. The results indicate that the proposed methodology has great potential to improve the structural monitoring of wind turbines in operation.

The obtained results contribute to the development of methods based on the analysis of measurement data for monitoring the technical condition of wind turbines, increasing their reliability and safety of use. The proposed methods are particularly effective in assessing the technical condition of structures and in the future will become a basic expert tool in identifying changes in construction objects such as wind turbines.

5.1 Products derived from this research

1. Scientific product - Journal

    - Coelho, J.S.; Machado, M.R.; Sousa, A.A.S.R. PyMLDA: A Python open-source code for Machine Learning Damage Assess-

ment, Software Impacts, Volume 19, 2024, ISSN 2665-9638. https://doi.org/10.1016/j.simpa.2024.100628.

- Coelho, J.S.; Machado, M.R.; Dutkiewicz, M.; Teloli, R.O. Data-driven machine learning for pattern recognition and detection of loosening torque in bolted joints. J. Braz. Soc. Mech. Sci. Eng. 46, 75 (2024). https://doi.org/10.1007/s40430-023-04628-6.
- Sousa A.A.S.R.; Coelho, J.S.; Machado M.R.; Dutkiewicz M. Multiclass Supervised Machine Learning Algorithms Applied to Damage and Assessment Using Beam Dynamic Response. J. Vib. Eng. Technol. 11, 2709–2731 (2023). https://doi.org/10.1007/s42417-023-01072-7.
- Coelho, J. S.; Machado M. R.; Dutkiewicz M. Integrating Virtual Sensor Data Augmentation into Machine Learning for Damage Quantification of Bolted Structures under Assembly Uncertainty,2025, Under review.

2. Technological product - Patent request, Registration of Computer Programs and Open code.

    - Machado, M.R.; Coelho, J.S.; de Sousa, A.A.S.R. Structural Integrity Monitoring Based on Machine Learning Techniques. 2024, Brasil. Patente: Privilégio de Inovação. Número do registro: BR1020240152867, título: ”Método para o Monitoramento de Integridade Estrutural Baseado em Técnicas de Aprendizado de Máquina”, Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial. Depósito: 25/07/2024.
    - Machado, M.R.; Coelho, J.S.; Sousa, A.A.S.R. PyMLDA - Machine Learning for Damage Assessment. 2024. Patente: Programa de Computador. Número do registro: BR512024001008-4, data de registro: 17/01/2024, título: ”PyMLDA - Machine Learning for Damage Assessment”, Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial.

- M.R. Machado , J.S. Coelho , A.A. S. R. Sousa (2024). PyMLDA - Machine Learning for Damage Assessment. Open code: https://github.com/mromarcela/PyMLDA.

5.2 Suggestions for further work

The study carried out in this thesis opens up new questions to be explored, and possible improvements can be implemented. The following are some aspects that could be addressed in future research.

- In the study, only ML models were used for classification, in order to distinguish between normal operation and turbine failure. As a suggestion, we propose the application of regression-based ML models, an approach that shows promise for improving the analysis and monitoring of turbine performance.
- Include in the methodology the optimization of resource selection and the exploration of clustering techniques to increase the accuracy of the models.
- Integrating multiphysics data and Machine Learning for monitoring wind turbines. In this context, the integration of multiphysics data with ML techniques appears as a promising solution to improve the understanding and performance of these complex systems.
- Incorporate DL techniques into the methodology, such as neural networks and Variational Autoencoders, to improve data modeling and model accuracy.

6 Bibliography

[1] M. Elgendi, M. AlMallahi, A. Abdelkhalig, and M. Y. Selim, "A review of wind turbines in complex terrain," *International Journal of Thermofluids*, vol. 17, p. 100289, 2023. (p. 11)

[2] GWEC, "Global wind report," tech. rep., Global Wind Energy Council, 2024. (p. 11, 12)

[3] M. Machado and M. Dutkiewicz, "Wind turbine vibration management: An integrated analysis of existing solutions, products, and open-source developments," *Energy Reports*, vol. 11, pp. 3756–3791, 2024. (p. 11, 37, 38)

[4] A. of 20 May 2016, "On investments in wind farms, ustawa z dnia 20 maja 2016 r. o inwestycjach w zakresie elektrowni wiatrowych, dz.u.2024.317," (p. 12)

[5] A. of 7 July 1994, "Construction law, ustawa z dnia 7 lipca 1994 r. prawo budowlane, dz.u.2024.725," (p. 12)

[6] A. of 27 March 2003, "On spatial planning and development, ustawa z dnia 27 marca 2003 r. o planowaniu i zagospodarowaniu przestrzennym, dz.u.2024.1130," (p. 12)

[7] F. P. G. de Sá, D. N. Brandão, E. Ogasawara, R. d. C. Coutinho, and R. F. Toso, "Wind turbine fault detection: A semi-supervised learning approach with automatic evolutionary feature selection," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 323–328, July 2020. (p. 12)

[8] Z. Liu and L. Zhang, "A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings," *Measurement*, vol. 149, p. 107002, 2020. (p. 13)

[9] Y. Amirat, M. Benbouzid, E. Al-Ahmar, B. Bensaker, and S. Turri, "A brief status on condition monitoring and fault diagnosis in wind energy conversion systems," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 9, pp. 2629–2636, 2009. (p. 14)

[10] R. McKenna, P. Ostman v.d. Leye, and W. Fichtner, "Key challenges and prospects for large wind turbines," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 1212–1221, 2016. (p. 14)

[11] G. Helbing and M. Ritter, "Deep learning for fault detection in wind turbines," *Renewable and Sustainable Energy Reviews*, vol. 98, pp. 189–198, 2018. (p. 14)

[12] Z. Gao and X. Liu, "An overview on fault diagnosis, prognosis and resilient control for wind turbine systems," *Processes*, vol. 9, no. 2, 2021. (p. 14, 16)

[13] J. Liang, K. Zhang, A. Al-Durra, S. Muyeen, and D. Zhou, "A state-of-the-art review on wind power converter fault diagnosis," *Energy Reports*, vol. 8, pp. 5341–5369, 2022. (p. 14)

[14] "This Photo Of Two Engineers Hugging Atop A Burning Turbine Before They Died Is Heart-Breaking!m." `http://www.archipelagofiles.com/2014/03/this-photo-of-two-engineers-hugging.html`. [Accessed on November 05, 2024]. (p. 15)

[15] T. Marques, "Hélice de torre eólica com mais de 20 metros desaba no litoral do Piauí." `https://agenciasertao.com/2020/02/03/helice-de-torre-eolica-com-mais-de-20-metros-desaba-no-litoral-do-piaui/`. [Accessed on November 05, 2024]. (p. 15)

[16] M. Zimmermann, "Feuerwehr-Einsatz in der Nacht: Brand zerstört Windrad bei Losheim – erste Hinweise auf Unglücksursache (mit Bildergalerie)." `https://www.saarbruecker-zeitung.de/blaulicht/losheim-brand-zerstoert-windrad-erste-hinweise-auf-ursache_aid-82064363`. [Accessed on November 05, 2024]. (p. 15)

[17] "Texas Tornado Takes Down Wind Turbines]." `https://1079ishot.com/texas-tornado-wind-turbines/`. [Accessed on November 05, 2024]. (p. 15)

[18] S. Asian, G. Ertek, C. Haksoz, S. Pakter, and S. Ulun, "Wind Turbine Accidents: A Data Mining Study," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1567–1578, 2017. (p. 14)

[19] D. Thomas, "Unveiling wind turbine failures causes, detection, and prevention for enhanced reliability," *Journal of Failure Analysis and Prevention*, vol. 24, p. 2051–2053, Aug. 2024. (p. 14, 16)

[20] S. Singh, M. Kharub, J. Singh, J. Singh, and V. Jangid, "Brief survey on mechanical failure and preventive mechanism of turbine blades," *Materials Today: Proceedings*, vol. 38, pp. 2515–2524, 2021. International Conference Exposition on Mechanical, Material and Manufacturing Technology (ICE3MT). (p. 14)

[21] H. Liu, Y. Wang, T. Zeng, H. Wang, S.-C. Chan, and L. Ran, "Wind turbine generator failure analysis and fault diagnosis: A review," *IET Renewable Power Generation*, vol. 18, no. 15, pp. 3127–3148, 2024. (p. 15)

[22] K. Morozovska, F. Bragone, A. X. Svensson, D. A. Shukla, and E. Hellstenius, "Trade-offs of wind power production: A study on the environmental implications of raw materials mining in the life cycle of wind turbines," *Journal of Cleaner Production*, vol. 460, p. 142578, 2024. (p. 15)

[23] W. Y. Liu, H. Gu, Q. W. Gao, and Y. Zhang, "A review on wind turbines gearbox fault diagnosis methods," *Journal of Vibroengineering*, vol. 23, pp. 26–43, jan 2021. (p. 16)

[24] P. Zhang and D. Lu, "A survey of condition monitoring and fault diagnosis toward integrated o&m for wind turbines," *Energies*, vol. 12, no. 14, 2019. (p. 16)

[25] E. Chatzi, I. Abdallah, M. Hofsäß, O. Bischoff, S. Barber, and Y. Marykovskiy, "Aventa av-7 eth zurich research wind turbine scada and high-frequency structural health monitoring (shm) data," aug 2023. (p. 17, 94)

[26] J.-S. Chou and W.-T. Tu, "Failure analysis and risk management of a collapsed large wind turbine tower," *Engineering Failure Analysis*, vol. 18, no. 1, pp. 295–313, 2011. (p. 17, 18)

[27] A. Mehmanparast, S. Lotfian, and S. P. Vipin, "A review of challenges and opportunities associated with bolted flange connections in the offshore wind industry," *Metals*, vol. 10, no. 6, 2020. (p. 18)

[28] T. C. Nguyen, T. C. Huynh, J. H. Yi, and J. T. Kim, "Hybrid bolt-loosening detection in wind turbine tower structures by vibration and impedance responses," *Wind and Structures, An International Journal*, vol. 24, no. 4, pp. 385–403, 2017. (p. 18, 35, 40)

[29] I. Shakeri, H. K. Danielsen, A. Tribhou, S. Fæster, O. V. Mishin, and M. A. Eder, "Effect of manufacturing defects on fatigue life of high strength steel bolts for wind turbines," *Engineering Failure Analysis*, vol. 141, p. 106630, 2022. (p. 18)

[30] D. Liu and X. Shang, "Failure investigation of the wind turbine blade root bolt," *Journal of Failure Analysis and Prevention*, vol. 13, no. 3, pp. 333–339, 2013. (p. 18)

[31] G. Gremza and J. Zamorowski, "Damage Analysis of the Blade to the Rotor Hub Connection in the Wind Turbine," *Journal of Civil Engineering, Environment and Architecture*, vol. 65, pp. 55–67, 2018. (p. 18)

[32] Y. Deng, Y. Zhao, H. Ju, T.-H. Yi, and A. Li, "Abnormal data detection for structural health monitoring: State-of-the-art review," *Developments in the Built Environment*, vol. 17, p. 100337, 2024. (p. 18)

[33] D. J. Inman, C. R. Farrar, V. L. Junior, and V. S. Junior, *Damage Prognosis: For Aerospace, Civil and Mechanical Systems*. 2005. (p. 18)

[34] H. Habibi, I. Howard, and S. Simani, "Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review," *Renewable Energy*, vol. 135, pp. 877–896, 2019. (p. 19)

[35] D. Mariano-Hernández, L. Hernández-Callejo, A. Zorita-Lamadrid, O. Duque-Pérez, and F. Santos García, "A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect diagnosis," *Journal of Building Engineering*, vol. 33, p. 101692, 2021. (p. 19)

[36] A. Abid, M. T. Khan, and J. Iqbal, "A review on fault detection and diagnosis techniques: basics and beyond," *Artificial Intelligence Review*, vol. 54, p. 3639–3664, Nov. 2020. (p. 19)

[37] R. Kumar, M. Ismail, W. Zhao, M. Noori, A. R. Yadav, S. Chen, V. Singh, W. A. Altabey, A. I. H. Silik, G. Kumar, J. Kumar, and A. Balodi, "Damage detection of wind turbine system based on signal processing approach: a critical review," *Clean Technologies and Environmental Policy*, vol. 23, p. 561–580, Jan. 2021. (p. 19)

[38] J. S. Coelho, M. R. Machado, M. Dutkiewicz, and R. O. Teloli, "Data-driven machine learning for pattern recognition and detection of loosening torque in bolted joints," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, Jan. 2024. (p. 19, 53, 82, 108, 113, 114)

[39] Y. Du, S. Zhou, X. Jing, Y. Peng, H. Wu, and N. Kwok, "Damage detection techniques for wind turbine blades: A review," *Mechanical Systems and Signal Processing*, vol. 141, p. 106445, 2020. (p. 19)

[40] J. C. PERAFÁN-LÓPEZ and J. SIERRA-PÉREZ, "An unsupervised pattern recognition methodology based on factor analysis and a genetic-dbscan algorithm to infer operational conditions from strain measurements in structural applications," *Chinese Journal of Aeronautics*, vol. 34, no. 2, pp. 165–181, 2021. (p. 24, 33)

[41] D. Reagan, A. Sabato, and C. Niezrecki, "Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges," *Structural Health Monitoring*, vol. 17, no. 5, pp. 1056–1072, 2018. (p. 24)

[42] Y. Liu, "Health monitoring data driven prediction of the civil aircraft structural condition," in *International Conference on Statistics, Data Science, and Computational Intelligence (CSDSCI 2022)* (G. N. Beligiannis, ed.), vol. 12510, p. 125100V, International Society for Optics and Photonics, SPIE, 2023. (p. 24)

[43] V. Giurgiutiu, "Coupling of PWAS Transducers to the Monitored Structure," in *Structural Health Monitoring with Piezoelectric Wafer Active Sensors*, pp. 395–443, Elsevier, 2014. (p. 24, 29, 74)

[44] A. Rytter, *Vibrational Based Inspection of Civil Engineering Structures*. PhD thesis, Aalborg Universitet, 1993. (p. 24)

[45] C. R. Farrar and N. A. Lieven, "Damage prognosis: The future of structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 623–632, 2007. (p. 25)

[46] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, "A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications,"

Mechanical Systems and Signal Processing, vol. 147, p. 107077, jan 2021. (p. 25, 26)

[47] W. Fan and P. Qiao, "Vibration-based damage identification methods: A review and comparative study," *Structural Health Monitoring*, vol. 10, no. 1, pp. 83–111, 2011. (p. 26)

[48] "On information value and decision analyses," *Structural Safety*, vol. 113, p. 102481, 2025. The Joint Committee of Structural Safety: past, present and a perspective on the future. (p. 27)

[49] "Value of information from vibration-based structural health monitoring extracted via bayesian model updating," *Mechanical Systems and Signal Processing*, vol. 166, p. 108465, 2022. (p. 27)

[50] Y. Bao and H. Li, "Machine learning paradigm for structural health monitoring," *Structural Health Monitoring*, vol. 20, pp. 1353 – 1372, 2020. (p. 27)

[51] F.-G. Yuan, S. A. Zargar, Q. Chen, and S. Wang, "Machine learning for structural health monitoring: challenges and opportunities," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020* (D. Zonta, H. Sohn, and H. Huang, eds.), vol. 1137903, p. 2, SPIE, apr 2020. (p. 27, 28, 73)

[52] M. Flah, I. Nunez, W. B. Chaabene, and M. Nehdi, "Machine learning algorithms in civil structural health monitoring: A systematic review," *Archives of Computational Methods in Engineering*, vol. 28, pp. 2621 – 2643, 2020. (p. 27, 28)

[53] A. Malekloo, E. Ozer, M. AlHamaydeh, and M. Girolami, "Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights," *Structural Health Monitoring*, vol. 21, no. 4, pp. 1906–1955, 2022. (p. 27, 76, 77)

[54] K. Smarsly, K. Dragos, and J. Wiggenbrock, "Machine learning techniques for structural health monitoring," *European Workshop on Structural Health Monitoring (EWSHM 2016)*, 2016. (p. 27, 28)

[55] C. R. Farrar and K. Worden, *Structural Health Monitoring: A Machine Learning Perspective*. Wiley, nov 2012. (p. 27, 30, 54, 73, 74)

[56] Y. Bao, Z. Chen, S. Wei, Y. Xu, Z. Tang, and H. Li, "The state of the art of data science and engineering in structural health monitoring," *Engineering*, 2019. (p. 28)

[57] M. Azimi, A. Eslamlou, and G. Pekcan, "Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review," *Sensors (Basel, Switzerland)*, vol. 20, 2020. (p. 28)

[58] X. Tan, X. Sun, W. zhong Chen, B. Du, J. Ye, and L. Sun, "Investigation on the data augmentation using machine learning algorithms in structural health monitoring information," *Structural Health Monitoring*, vol. 20, pp. 2054 – 2068, 2021. (p. 28)

[59] Christopher M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2nd ed., 2006. (p. 29, 53, 128)

[60] A. R. Webb and K. D. Copsey, *Statistical Pattern Recognition*. 3rd ed. ed., 2011. (p. 29)

[61] E. Figueiredo and J. Brownjohn, "Three decades of statistical pattern recognition paradigm for shm of bridges," *Structural Health Monitoring*, vol. 21, no. 6, pp. 3018–3054, 2022. (p. 30, 56)

[62] K. Worden and G. Manson, "The application of machine learning to structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 515–537, 2007. (p. 30)

[63] Z. Zhang and C. Sun, "Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating," *Structural Health Monitoring*, vol. 20, no. 4, pp. 1675–1688, 2021. (p. 30)

[64] L. A. Bull, P. Gardner, T. J. Rogers, E. J. Cross, N. Dervilis, and K. Worden, "Probabilistic inference for structural health monitoring: New modes of learning from data," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 7, no. 1, p. 03120003, 2021. (p. 30, 74)

[65] D. Sen and S. Nagarajaiah, "Data-driven approach to structural health monitoring using statistical learning algorithms," *Intelligent Systems, Control and Automation: Science and Engineering*, vol. 92, pp. 295–305, 2018. (p. 31)

[66] J. Zhang, T. Sato, S. Iai, and T. Hutchinson, "A pattern recognition technique for structural identification using observed vibration signals: Linear case studies," *Engineering Structures*, vol. 30, no. 5, pp. 1439–1446, 2008. (p. 31)

[67] M. Gul and F. Necati Catbas, "Statistical pattern recognition for Structural Health Monitoring using time series modeling: Theory and experimental verifications," *Mechanical Systems and Signal Processing*, vol. 23, no. 7, pp. 2192–2204, 2009. (p. 31)

[68] A. Entezami, H. Sarmadi, B. Behkamal, and S. Mariani, "Big data analytics and structural health monitoring: A statistical pattern recognition-based approach," *Sensors (Switzerland)*, vol. 20, no. 8, 2020. (p. 31)

[69] A. Herrera-Iriarte, J. Alvarez-Montoya, and J. Sierra-Pérez, "Study of sensitivity for strain-based structural health monitoring," *IOP Conference Series: Materials Science and Engineering*, vol. 836, no. 1, 2020. (p. 31)

[70] I. Trendafilova and W. Heylen, "Categorisation and pattern recognition methods for damage localisation from vibration measurements," *Mechanical Systems and Signal Processing*, vol. 17, no. 4, pp. 825–836, 2003. (p. 32)

[71] L. Qiao, A. Esmaeily, and H. G. Melhem, "Structural damage detection using signal pattern-recognition," *Key Engineering Materials*, vol. 400-402, pp. 465–470, 2009. (p. 32)

[72] D. Shan, C. Fu, and Q. Li, "Experimental investigation of damage identification for continuous railway bridges," *Journal of Modern Transportation*, vol. 20, no. 1, pp. 1–9, 2012. (p. 33)

[73] S. Goswami and P. Bhattacharya, "Pattern Recognition for damage detection in aerospace vehicle structures," *Proceedings - 2012 3rd International Conference on Emerging Applications of Information Technology, EAIT 2012*, no. Figure 2, pp. 178–182, 2012. (p. 33)

[74] L. Li, M. Morgantini, and R. Betti, "Structural damage assessment through a new generalized autoencoder with features in the quefrency domain," *Mechanical Systems and Signal Processing*, vol. 184, p. 109713, 2023. (p. 34)

[75] C. C. Ciang, J.-R. Lee, and H.-J. Bang, "Structural health monitoring for a wind turbine system: a review of damage detection methods," *Measurement Science and Technology*, vol. 19, p. 122001, oct 2008. (p. 35)

[76] W. Kim, J.-H. Yi, J.-T. Kim, and J.-H. Park, "Vibration-based structural health assessment of a wind turbine tower using a wind turbine model," *Procedia Engineering*, vol. 188, pp. 333–339, 2017. Structural Health Monitoring - From Sensing to Diagnosis and Prognosis. (p. 35)

[77] W. Weijtjens, T. Verbelen, E. Capello, and C. Devriendt, "Vibration based structural health monitoring of the substructures of five offshore wind turbines," *Procedia Engineering*, vol. 199, pp. 2294–2299, 2017. X International Conference on Structural Dynamics, EURODYN 2017. (p. 35)

[78] I. Antoniadou, N. Dervilis, R. J. Barthorpe, G. Manson, and K. Worden, "Advanced tools for damage detection in wind turbines," *Key Engineering Materials*, vol. 569-570, pp. 547–554, 2013. (p. 35)

[79] N. Dervilis, M. Choi, I. Antoniadou, K. Farinholt, S. Taylor, R. J. Barthorpe, G. Park, C. R. Farrar, and K. Worden, "Machine learning applications for a wind turbine blade under continuous fatigue loading," *Key Engineering Materials*, vol. 588, p. 166–174, Oct. 2013. (p. 35)

[80] N. Dervilis, M. Choi, S. Taylor, R. Barthorpe, G. Park, C. Farrar, and K. Worden, "On damage diagnosis for a wind turbine blade using pattern recognition," *Journal of Sound and Vibration*, vol. 333, no. 6, pp. 1833–1850, 2014. (p. 35)

[81] I. Antoniadou, N. Dervilis, E. Papatheou, A. E. Maguire, and K. Worden, "Aspects of structural health and condition monitoring of offshore wind turbines," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2035, 2015. (p. 36)

[82] A. Joshuva. and V. Sugumaran., "A data driven approach for condition monitoring of wind turbine blade using vibration signals through best-first tree algorithm and functional trees algorithm: A comparative study," *ISA Transactions*, vol. 67, pp. 160–172, 2017. (p. 36)

[83] S. Tsiapoki, M. W. Häckell, T. Grießmann, and R. Rolfes, "Damage and ice detection on wind turbine rotor blades using a three-tier modular structural

health monitoring framework," *Structural Health Monitoring*, vol. 17, no. 5, pp. 1289–1312, 2018. (p. 36)

[84] A. Joshuva, R. Vishnuvardhan, G. Deenadayalan, R. Sathishkumar, and S. Sivakumar, "Implementation of rule based classifiers for wind turbine blade fault diagnosis using vibration signals," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2 Special Issue 11, pp. 320–331, 2019. (p. 36)

[85] K. Chandrasekhar, N. Stevanovic, E. J. Cross, N. Dervilis, and K. Worden, "Damage detection in operational wind turbine blades using a new approach based on machine learning," *Renewable Energy*, vol. 168, pp. 1249–1264, 2021. (p. 36)

[86] M. Khazaee, P. Derian, and A. Mouraud, "A comprehensive study on structural health monitoring (shm) of wind turbine blades by instrumenting tower using machine learning methods," *Renewable Energy*, vol. 199, pp. 1568–1579, 2022. (p. 37)

[87] J. Arockia Dhanraj, R. S. Alkhawaldeh, P. Van De, V. Sugumaran, N. Ali, N. Lakshmaiya, P. K. Chaurasiya, P. S., K. Velmurugan, M. S. Chowdhury, S. Channumsin, S. Sreesawet, and H. Fayaz, "Appraising machine learning classifiers for discriminating rotor condition in 50w–12v operational wind turbine for maximizing wind energy production through feature extraction and selection process," *Frontiers in Energy Research*, vol. 10, 2022. (p. 37)

[88] Y. Vidal, G. Aquino, F. Pozo, and J. E. M. Gutiérrez-Arias, "Structural health monitoring for jacket-type offshore wind turbines: Experimental proof of concept," *Sensors*, vol. 20, no. 7, 2020. (p. 37)

[89] J. X. Leon-Medina, M. Anaya, N. Parés, D. A. Tibaduiza, and F. Pozo, "Structural damage classification in a jacket-type wind-turbine foundation using principal component analysis and extreme gradient boosting," *Sensors*, vol. 21, no. 8, 2021. (p. 37)

[90] L. Ren and B. Yong, "Wind turbines fault classification treatment method," *Symmetry*, vol. 14, no. 4, 2022. (p. 37)

[91] E. Hoxha, Y. Vidal, and F. Pozo, "Damage diagnosis for offshore wind turbine foundations based on the fractal dimension," *Applied Sciences*, vol. 10, no. 19, 2020. (p. 38)

[92] C. U. Nguyen, T. C. Huynh, and J. T. Kim, "Vibration-based damage detection in wind turbine towers using artificial neural networks," *Structural Monitoring and Maintenance*, vol. 5, no. 4, pp. 507–519, 2018. (p. 38)

[93] P. S. Zarrin, C. Martin, P. Langendoerfer, C. Wenger, and M. DIaz, "Vibration Analysis of a Wind Turbine Gearbox for Off-cloud Health Monitoring through Neuromorphic-computing," *IECON Proceedings (Industrial Electronics Conference)*, vol. 2021-October, 2021. (p. 38)

[94] H. M. Praveen, G. Sabareesh, V. Inturi, and A. Jaikanth, "Component level signal segmentation method for multi-component fault detection in a wind turbine gearbox," *Measurement*, vol. 195, p. 111180, 2022. (p. 38)

[95] M. Elforjani, "Diagnosis and prognosis of real world wind turbine gears," *Renewable Energy*, vol. 147, pp. 1676–1693, 2020. (p. 39)

[96] Q. Gao, X. Wu, J. Guo, H. Zhou, and W. Ruan, "Machine-Learning-Based Intelligent Mechanical Fault Detection and Diagnosis of Wind Turbines," *Mathematical Problems in Engineering*, vol. 2021, no. 1, 2021. (p. 39)

[97] J. Vives, "Vibration analysis for fault detection in wind turbines using machine learning techniques," *Advances in Computational Intelligence*, vol. 2, no. 1, pp. 1–12, 2022. (p. 39)

[98] J. Vives, E. Roses Albert, E. Quiles, J. Palací, and T. Fuster, "Vibration Analysis for Fault Detection of Wind Turbines by Combining Machine-Learning Techniques and 3D Scanning Laser," *Genetics Research*, vol. 2022, 2022. (p. 39)

[99] A. Amin, A. Bibo, M. Panyam, and P. Tallapragada, "Vibration based fault diagnostics in a wind turbine planetary gearbox using machine learning," *Wind Engineering*, vol. 47, no. 1, pp. 175–189, 2023. (p. 39)

[100] A. Meyer, "Vibration fault diagnosis in wind turbines based on automated feature learning," *Energies*, vol. 15, no. 4, 2022. (p. 39)

[101] D. Rommel, D. D. Maio, and T. Tinga, "Calculating loads and life-time reduction of wind turbine gearbox and generator bearings due to shaft misalignment," *Wind Engineering*, vol. 45, pp. 547 – 568, 2020. (p. 40)

[102] Z. Xue, Z. Zhou, H. Lai, S. Shao, J. Rao, Y. Li, L. He, and Z. Jia, "Investigations on rigid–flexible coupling multibody dynamics of 5mw wind turbine," *Energy Science & Engineering*, 2024. (p. 40)

[103] X. Pang, D. Zhu, X. Zuo, D. Wang, W. Hao, M. Qiu, and D. Liu, "Analysis of rigid-flexible coupled collision force in a variable load offshore wind turbine main three-row cylindrical roller bearing," *Lubricants*, 2024. (p. 40)

[104] S. Son, H.-W. Lee, J.-Y. Han, Y. Kim, and J. Kang, "Development of high speed coupling for 2mw class wind turbine," *Journal of the Korean Society of Marine Engineering*, vol. 38, pp. 262–268, 2014. (p. 40)

[105] D. Zheng, H. Guo, S. Zhang, and Y. Liu, "Study on fatigue performance of double cover plate through-core bolted joint of rectangular concrete-filled steel tube bundle wind turbine towers," *Journal of Constructional Steel Research*, vol. 203, p. 107830, 2023. (p. 40)

[106] S.-Z. Li, H. Li, X.-H. Zhou, Y.-H. Wang, X.-H. Li, D. Gan, and R.-H. Zhu, "Damage detection of flange bolts in wind turbine towers using dynamic strain responses," *Journal of Civil Structural Health Monitoring*, vol. 13, pp. 67–81, jan 2023. (p. 40)

[107] Y. Yue, J. Tian, Y. Bai, K. Jia, J. He, D. Luo, and T. Chen, "Applicability Analysis of Inspection and Monitoring Technologies in Wind Turbine Towers," *Shock and Vibration*, vol. 2021, 2021. (p. 40)

[108] B. Badrkhani Ajaei and S. Soyoz, "Effects of preload deficiency on fatigue demands of wind turbine tower bolts," *Journal of Constructional Steel Research*, vol. 166, p. 105933, 2020. (p. 40)

[109] Y. Li, W. Sun, R. Jiang, and Y. Han, "Signal-segments cross-coherence method for nonlinear structural damage detection using free-vibration signals," *Advances in Structural Engineering*, vol. 23, no. 6, pp. 1041–1054, 2020. (p. 40)

[110] H. Xianlong and S. Tianli, "A New Identification Method for Bolt Looseness in Wind Turbine Towers," *Shock and Vibration*, vol. 2019, pp. 1–10, jul 2019. (p. 40)

[111] P. Schaumann, R. Eichstädt, and A. Stang, "Advanced performance assessment methods for high-strength bolts in ring-flange connections," *Stahlbau*, vol. 87, no. 5, pp. 446–455, 2018. (p. 40)

[112] T.-C. Nguyen, T.-C. Huynh, and J.-T. Kim, "Numerical evaluation for vibration-based damage detection in wind turbine tower structure," *Wind and Structures*, vol. 21, pp. 657–675, 12 2015. (p. 40)

[113] E. Zugasti, A. G. González, J. Anduaga, M. A. Arregui, and F. Martínez, "Nullspace and autoregressive damage detection: a comparative study," *Smart Materials and Structures*, vol. 21, p. 085010, jul 2012. (p. 40)

[114] A. Rincón-Casado, J. Juliá-Lerma, D. García-Vallejo, and J. Domínguez, "Experimental estimation of the residual fatigue life of in-service wind turbine bolts," *Engineering Failure Analysis*, vol. 141, p. 106658, 2022. (p. 40)

[115] A. S. Verma, Z. Jiang, Z. Ren, Z. Gao, and N. P. Vedvik, "Effects of Wind-Wave Misalignment on a Wind Turbine Blade Mating Process: Impact Velocities, Blade Root Damages and Structural SafetyAssessment," *Journal of Marine Science and Application*, vol. 19, no. 2, pp. 218–233, 2020. (p. 40)

[116] J. Kang, H. Liu, and D. Fu, "Fatigue life and strength analysis of a main shaft-to-hub bolted connection in a wind turbine," *Energies*, vol. 12, no. 1, 2019. (p. 40)

[117] A. S. Verma, Z. Jiang, N. P. Vedvik, Z. Gao, and Z. Ren, "Impact assessment of a wind turbine blade root during an offshore mating process," *Engineering Structures*, vol. 180, pp. 205–222, 2019. (p. 40)

[118] C.-H. Loh, K. J. Loh, Y.-S. Yang, W.-Y. Hsiung, and Y.-T. Huang, "Vibration-based system identification of wind turbine system," *Structural Control and Health Monitoring*, vol. 24, no. 3, p. e1876, 2017. e1876 STC-14-0158.R3. (p. 40)

[119] C. Deters, H.-K. Lam, E. L. Secco, H. A. Würdemann, L. D. Seneviratne, and K. Althoefer, "Accurate bolt tightening using model-free fuzzy control for wind turbine hub bearing assembly," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 1–12, 2015. (p. 40)

[120] H. Huang, Y. Wang, and Q. Pang, "Analysis and prediction of wind turbine bolts based on GPR method," *Journal of Mechanical Science and Technology*, feb 2023. (p. 40)

[121] R. W. Messler, "Preface," in *Joining of Materials and Structures* (R. W. Messler, ed.), pp. xxi–xxiv, Burlington: Butterworth-Heinemann, 2004. (p. 41)

[122] P. R. Childs, "16 - fastening and power screws," in *Mechanical Design Engineering Handbook (Second Edition)* (P. R. Childs, ed.), pp. 773–832, Butterworth-Heinemann, second edition ed., 2019. (p. 41)

[123] J. Mackerle, "Finite element analysis of fastening and joining: A bibliography (1990–2002)," *International Journal of Pressure Vessels and Piping*, vol. 80, no. 4, pp. 253–271, 2003. (p. 41)

[124] M. R. W. Brake, *The Mechanics of Jointed Structures*. 2018. (p. 41)

[125] R. A. Ibrahim and C. L. Pettit, "Uncertainties and dynamic problems of bolted joints and other fasteners," *Journal of Sound and Vibration*, vol. 279, no. 3-5, pp. 857–936, 2005. (p. 42)

[126] R. Miao, R. Shen, S. Zhang, and S. Xue, "A review of bolt tightening force measurement and loosening detection," *Sensors (Switzerland)*, vol. 20, no. 11, 2020. (p. 42, 43)

[127] O. Eraliev, K.-H. Lee, and C.-H. Lee, "Vibration-Based Loosening Detection of a Multi-Bolt Structure Using Machine Learning Algorithms," *Sensors*, vol. 22, p. 1210, feb 2022. (p. 43, 44, 46)

[128] I. Zaman, A. Khalid, B. Manshoor, S. Araby, and M. I. Ghazali, "The effects of bolted joints on dynamic response of structures," *IOP Conference Series: Materials Science and Engineering*, vol. 50, no. 1, 2013. (p. 43)

[129] J. Huang, J. Liu, H. Gong, and X. Deng, "A comprehensive review of loosening detection methods for threaded fasteners," *Mechanical Systems and Signal Processing*, vol. 168, no. December 2021, 2022. (p. 43, 44)

[130] H. Gong, X. Ding, J. Liu, and H. Feng, "Review of research on loosening of threaded fasteners," *Friction*, vol. 10, pp. 335–359, mar 2022. (p. 43, 45)

[131] J. H. Bickford and M. Oliver, *Introduction to the Design and Behavior of Bolted Joints: Non-Gasketed Joints*. 2022. (p. 43)

[132] G. H. Junker, "New criteria for self-loosening of fasteners under vibration," *SAE Transactions*, vol. 78, pp. 314–335, 1969. (p. 43)

[133] N. Pai and D. Hess, "Three-dimensional finite element analysis of threaded fastener loosening due to dynamic shear load," *Engineering Failure Analysis*, vol. 9, no. 4, pp. 383–402, 2002. (p. 43)

[134] N. PAI and D. HESS, "Experimental study of loosening of threaded fasteners due to dynamic shear loads," *Journal of Sound and Vibration*, vol. 253, no. 3, pp. 585–602, 2002. (p. 43)

[135] Y. Chen, Q. Gao, and Z. Guan, "Self-Loosening Failure Analysis of Bolt Joints under Vibration considering the Tightening Process," *Shock and Vibration*, vol. 2017, pp. 1–15, 2017. (p. 43, 46)

[136] S. M. Y. Nikravesh and M. Goudarzi, "A review paper on looseness detection methods in bolted structures," *Latin American Journal of Solids and Structures*, vol. 14, no. 12, pp. 2153–2176, 2017. (p. 43)

[137] M. R. M. Maciej Dutkiewicz, "Measurements in situ and spectral analysis of wind flow effects on overhead transmission lines," *Sound & Vibration*, vol. 53, no. 4, pp. 161–175, 2019. (p. 44)

[138] M. R. Machado and J. M. C. Dos Santos, "Reliability analysis of damaged beam spectral element with parameter uncertainties," *Shock and Vibration*, vol. 2015, no. 1, p. 574846, 2015. (p. 44)

[139] M. R. Machado, S. Adhikari, and J. M. C. Dos Santos, "Spectral element-based method for a one-dimensional damaged structure with distributed random properties," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 40, Aug. 2018. (p. 44)

[140] M. Machado and J. Dos Santos, "Effect and identification of parametric distributed uncertainties in longitudinal wave propagation," *Applied Mathematical Modelling*, vol. 98, pp. 498–517, 2021. (p. 44, 127)

[141] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. 2012. (p. 44, 73)

[142] Y. Selvaraj and C. Selvaraj, "Proactive maintenance of small wind turbines using IoT and machine learning models," *International Journal of Green Energy*, vol. 19, pp. 463–475, apr 2022. (p. 44)

[143] A. A. S. R. de Sousa, J. da Silva Coelho, M. R. Machado, and M. Dutkiewicz, "Multiclass supervised machine learning algorithms applied to damage and assessment using beam dynamic response," *Journal of Vibration Engineering & Technologies*, vol. 11, p. 2709–2731, July 2023. (p. 44, 50, 53, 59, 108)

[144] Y. Zhou, S. Wang, M. Zhou, H. Chen, C. Yuan, and Q. Kong, "Percussion-based bolt looseness identification using vibration-guided sound reconstruction," *Structural Control and Health Monitoring*, vol. 29, no. 2, pp. 2–5, 2022. (p. 44)

[145] F. Wang and G. Song, "1D-TICapsNet: An audio signal processing algorithm for bolt early looseness detection," *Structural Health Monitoring*, 2020. (p. 45)

[146] F. Wang and G. Song, "Bolt-looseness detection by a new percussion-based method using multifractal analysis and gradient boosting decision tree," *Structural Health Monitoring*, vol. 19, no. 6, pp. 2023–2032, 2020. (p. 45)

[147] D. Q. Tran, J. W. Kim, K. D. Tola, W. Kim, and S. Park, "Artificial intelligence-based bolt loosening diagnosis using deep learning algorithms for laser ultrasonic wave propagation data," *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–25, 2020. (p. 45)

[148] Y. Zhang, X. Zhao, X. Sun, W. Su, and Z. Xue, "Bolt loosening detection based on audio classification," *Advances in Structural Engineering*, vol. 22, no. 13, pp. 2882–2891, 2019. (p. 45)

[149] Q. Kong, J. Zhu, S. C. M. Ho, and G. Song, "Tapping and listening: A new approach to bolt looseness monitoring," *Smart Materials and Structures*, vol. 27, no. 7, 2018. (p. 45)

[150] X. Zhao, Y. Zhang, and N. Wang, "Bolt loosening angle detection technology using deep learning," *Structural Control and Health Monitoring*, vol. 26, p. e2292, jan 2019. (p. 45)

[151] Y. Zhang, X. Sun, K. J. Loh, W. Su, Z. Xue, and X. Zhao, "Autonomous bolt loosening detection using deep learning," *Structural Health Monitoring*, vol. 19, no. 1, pp. 105–122, 2020. (p. 45)

[152] Y. Yu, Y. Liu, J. Chen, D. Jiang, Z. Zhuang, and X. Wu, "Detection Method for Bolted Connection Looseness at Small Angles of Timber Structures based on Deep Learning," *Sensors*, vol. 21, p. 3106, apr 2021. (p. 45)

[153] H. C. Pham, Q. B. Ta, J. T. Kim, D. D. Ho, X. L. Tran, and T. C. Huynh, "Bolt-loosening monitoring framework using an image-based deep learning and graphical model," *Sensors (Switzerland)*, vol. 20, no. 12, pp. 1–19, 2020. (p. 45)

[154] L. Ramana, W. Choi, and Y. J. Cha, "Fully automated vision-based loosened bolt detection using the Viola–Jones algorithm," *Structural Health Monitoring*, vol. 18, no. 2, pp. 422–434, 2019. (p. 45)

[155] Y. J. Cha, K. You, and W. Choi, "Vision-based detection of loosened bolts using the Hough transform and support vector machines," *Automation in Construction*, vol. 71, no. Part 2, pp. 181–188, 2016. (p. 45)

[156] P. Razi, R. A. Esmaeel, and F. Taheri, "Improvement of a vibration-based damage detection approach for health monitoring of bolted flange joints in pipelines," *Structural Health Monitoring*, vol. 12, no. 3, pp. 207–224, 2013. (p. 46)

[157] D. Ziaja and P. Nazarko, "SHM system for anomaly detection of bolted joints in engineering structures," *Structures*, vol. 33, pp. 3877–3884, oct 2021. (p. 46)

[158] L. P. Miguel, R. d. O. Teloli, S. da Silva, and G. Chevallier, "Probabilistic machine learning for detection of tightening torque in bolted joints," *Structural Health Monitoring*, vol. 0, no. 0, pp. 1–16, 2022. (p. 46, 74)

[159] R. d. O. Teloli, P. Butaud, G. Chevallier, and S. da Silva, "Good practices for designing and experimental testing of dynamically excited jointed structures: The Orion beam," *Mechanical Systems and Signal Processing*, vol. 163, no. June 2021, p. 108172, 2022. (p. 46, 113, 126)

[160] Z. Zhuang, Y. Yu, Y. Liu, J. Chen, and Z. Wang, "Ultrasonic signal transmission performance in bolted connections of wood structures under different preloads," *Forests*, vol. 12, no. 6, 2021. (p. 46)

[161] F. Wang, "A novel autonomous strategy for multi-bolt looseness detection using smart glove and Siamese double-path CapsNet," *Structural Health Monitoring*, vol. 0, no. 0, pp. 1–11, 2021. (p. 46)

[162] F. Wang, Z. Chen, and G. Song, "Monitoring of multi-bolt connection looseness using entropy-based active sensing and genetic algorithm-based least square support vector machine," *Mechanical Systems and Signal Processing*, vol. 136, p. 106507, 2020. (p. 46)

[163] L. Zhou, S. X. Chen, Y. Q. Ni, and A. W. H. Choy, "EMI-GCN: A hybrid model for real-time monitoring of multiple bolt looseness using electromechanical impedance and graph convolutional networks," *Smart Materials and Structures*, vol. 30, no. 3, 2021. (p. 46)

[164] G. Mariniello, T. Pastore, C. Menna, P. Festa, and D. Asprone, "Structural damage detection and localization using decision tree ensemble and vibration data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 9, pp. 1129–1149, 2021. (p. 46)

[165] J. da Silva Coelho, M. R. Machado, and A. A. S. de Sousa, "Pymlda: A python open-source code for machine learning damage assessment," *Software Impacts*, vol. 19, p. 100628, 2024. (p. 48, 49)

[166] M. MACHADO, A. A. S. R. SOUSA, and J. S. COELHO, "Structural integrity monitoring based on machine learning techniques," INPI BR Patent BR10202401528, July 2024. (p. 48)

[167] W. Keith, a. M. G. Farrar Charles R, and P. Gyuhae, "The fundamental axioms of structural health monitoring," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2007. (p. 50)

[168] J. J. Sinou, "A review of damage detection and health monitoring of mechanical systems from changes in the measurement of linear and non-linear vibrations," *Mechanical Vibrations: Measurement, Effects and Control*, no. June, pp. 643–702, 2009. (p. 53, 59, 115)

[169] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, Y. Duan, A. Abdullah, L. Farhan, Y. Lu, A. Gupta, F. Albu, A. Abbosh, and Y. Gu, "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," *Journal of Big Data*, vol. 10, 2023. (p. 56, 130)

[170] F. Wang, A. Harker, M. Edirisinghe, and M. Parhizkar, "Tackling data scarcity challenge through active learning in materials processing with electrospray," *Advanced Intelligent Systems*, vol. n/a, no. n/a, p. 2300798, 2024. (p. 56, 130)

[171] G. Tsialiamanis, N. Dervilis, D. Wagg, and K. Worden, "Towards a population-informed approach to the definition of data-driven models for structural dynamics," *Mechanical Systems and Signal Processing*, vol. 200, p. 110581, 2023. (p. 56)

[172] I. H. Sarker, "Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective," *SN Computer Science*, vol. 2, jul 2021. (p. 56)

[173] A. Santos, E. Figueiredo, M. Silva, C. Sales, and J. Costa, "Machine learning algorithms for damage detection: Kernel-based approaches," *Journal of Sound and Vibration*, vol. 363, pp. 584–599, 2016. (p. 57)

[174] M. Azimi, A. D. Eslamlou, and G. Pekcan, "Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review," *Sensors*, vol. 20, no. 10, 2020. (p. 57)

[175] S. L. Brunton, J. Nathan Kutz, K. Manohar, A. Y. Aravkin, K. Morgansen, J. Klemisch, N. Goebel, J. Buttrick, J. Poskin, A. W. Blom-Schieber, T. Hogan, and D. McDonald, "Data-driven aerospace engineering: Reframing the industry with machine learning," *AIAA Journal*, vol. 59, no. 8, pp. 2820–2847, 2021. (p. 57)

[176] D. Samadian, I. B. Muhit, and N. Dawood, "Application of data-driven surrogate models in structural engineering: A literature review," *Archives of Computational Methods in Engineering*, jul 2024. (p. 57)

[177] A. Moallemi, A. Burrello, D. Brunelli, and L. Benini, "Model-based vs. data-driven approaches for anomaly detection in structural health monitoring: a case study," in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, 2021. (p. 57)

[178] G. Tsialiamanis, *A data-driven approach to modelling structures*. PhD thesis, University of Sheffield, August 2022. (p. 58, 59)

[179] H. V. Dang, H. Tran-Ngoc, T. V. Nguyen, T. Bui-Tien, G. De Roeck, and H. X. Nguyen, "Data-driven structural health monitoring using feature fusion and hybrid deep learning," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 4, pp. 2087–2103, 2021. (p. 59)

[180] H. Joseph, G. Quaranta, B. Carboni, and W. Lacarbonara, "Deep learning architectures for data-driven damage detection in nonlinear dynamic systems under random vibrations," *Nonlinear Dynamics*, vol. 112, p. 20611–20636, Sept. 2024. (p. 59)

[181] L. S. Barreto, M. R. Machado, J. C. Santos, B. B. d. Moura, and L. Khalij, "Damage indices evaluation for one-dimensional guided wave-based structural health monitoring," *Latin American Journal of Solids and Structures*, vol. 18, no. Lat. Am. j. solids struct., 2021 18(2), 2021. (p. 59, 115, 128)

[182] Z. Saeed, C. M. Firrone, and T. M. Berruti, "Joint identification through hybrid models improved by correlations," *Journal of Sound and Vibration*, vol. 494, p. 115889, 2021. (p. 60)

[183] C. Zang, H. Grafe, and M. Imregun, "Frequency-domain criteria for correlating and updating dynamic finite element models," *Mechanical Systems and Signal Processing*, vol. 15, no. 1, pp. 139–155, 2001. (p. 60, 61)

[184] H. Policarpo, M. Neves, and N. Maia, "A simple method for the determination of the complex modulus of resilient materials using a longitudinally vibrating three-layer specimen," *Journal of Sound and Vibration*, vol. 332, no. 2, pp. 246–263, 2013. (p. 60)

[185] C. Zang and M. Imregun, "Structural damage detection and localization using FRF-based model updating approach," *Key Engineering Materials*, vol. 245-246, pp. 191–200, 2003. (p. 61)

[186] C. Zang, M. I. Friswell, and M. Imregun, "Structural Health Monitoring and Damage Assessment Using Measured FRFs from Multiple Sensors, Part I: The Indicator of Correlation Criteria," *Key Engineering Materials*, vol. 245-246, pp. 131–140, 2003. (p. 61)

[187] T. Monnier, "Lamb waves-based impact damage monitoring of a stiffened aircraft panel using piezoelectric transducers," *Journal of Intelligent Material Systems and Structures*, vol. 17, no. 5, pp. 411–421, 2006. (p. 61)

[188] S. Banerjee, F. Ricci, E. Monaco, L. Lecce, and A. Mal, "Autonomous impact damage monitoring in a stiffened composite panel," *Journal of Intelligent Material Systems and Structures*, vol. 18, no. 6, pp. 623–633, 2007. (p. 61)

[189] S. Banerjee, F. Ricci, E. Monaco, and A. Mal, "A wave propagation and vibration-based approach for damage identification in structural components," *Journal of Sound and Vibration*, vol. 322, no. 1-2, pp. 167–183, 2009. (p. 61)

[190] J. Chen, B. Xu, and X. Zhang, "A vibration feature extraction method based on time-domain dimensional parameters and mahalanobis distance," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 2498178, 2021. (p. 62)

[191] W. Caesarendra and T. Tjahjowidodo, "A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing," *Machines*, vol. 5, no. 4, 2017. (p. 62)

[192] I. Trendafilova, "Vibration-based damage detection in structures using time series analysis," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 220, p. 261–272, Mar. 2006. (p. 62)

[193] C. Zhang, A. A. Mousavi, S. F. Masri, G. Gholipour, K. Yan, and X. Li, "Vibration feature extraction using signal processing techniques for structural health monitoring: A review," *Mechanical Systems and Signal Processing*, vol. 177, p. 109175, 2022. (p. 62)

[194] K. Alameh, N. Cité, G. Hoblos, and G. Barakat, "Feature extraction for vibration-based fault detection in permanent magnet synchronous motors," in *2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, pp. 163–168, 2015. (p. 62)

[195] M. Huang and Z. Liu, "Research on mechanical fault prediction method based on multifeature fusion of vibration sensing data," *Sensors*, vol. 20, p. 6, dec 2019. (p. 62)

[196] Knittel, Dominique, Makich, Hamid, and Nouari, Mohammed, "Milling diagnosis using artificial intelligence approaches," *Mechanics Industry*, vol. 20, no. 8, p. 809, 2019. (p. 62)

[197] A. Jolicoeur-Martineau, K. Fatras, and T. Kachman, "Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees," 2024. Preprint available at arXiv:2309.09968. (p. 64, 65, 70, 71)

[198] D. Martin, N. Kühl, and G. Satzger, "Virtual sensors," *Business Information Systems Engineering*, vol. 63, pp. 315–323, 2021. (p. 64)

[199] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," 2018. Preprint available at arXiv:1811.11264. (p. 65, 69)

[200] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Lognormal Distribution*, ch. 29, pp. 131–134. John Wiley & Sons, Ltd, 2010. (p. 67)

[201] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014. (p. 68)

[202] S. Kaddoura, *A Primer on Generative Adversarial Networks*. Springer International Publishing, 2023. (p. 68)

[203] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, "Improving and generalizing flow-based generative models with minibatch optimal transport," 2024. Preprint available at arXiv:2302.00482. (p. 70)

[204] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, ACM, Aug. 2016. (p. 70, 80)

[205] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge University Press, 2020. (p. 72)

[206] Ryan, Cooper, and Tauer, *Machine Learning The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012. (p. 73)

[207] Z. Jan, F. Ahamed, W. Mayer, N. Patel, G. Grossmann, M. Stumptner, and A. Kuusk, "Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities," *Expert Systems with Applications*, vol. 216, p. 119456, 2023. (p. 73)

[208] S. R. Vadyala, S. N. Betgeri, J. C. Matthews, and E. Matthews, "A review of physics-based machine learning in civil engineering," *Results in Engineering*, vol. 13, p. 100316, 2022. (p. 73)

[209] P. Cawley, "Structural health monitoring: Closing the gap between research and industrial deployment," *Structural Health Monitoring*, vol. 17, no. 5, pp. 1225–1244, 2018. (p. 73)

[210] M. Machado, S. Adhikari, and J. D. Santos, "A spectral approach for damage quantification in stochastic dynamic systems," *Mechanical Systems and Signal Processing*, vol. 88, pp. 253–273, 2017. (p. 73)

[211] D. Sarkar, R. Bali, and T. Sharma, *Practical Machine Learning with Python*. Berkeley, CA: Apress, 2018. (p. 73)

[212] M. Martinez-Luengo, A. Kolios, and L. Wang, "Structural health monitoring of offshore wind turbines: A review through the Statistical Pattern Recognition Paradigm," *Renewable and Sustainable Energy Reviews*, vol. 64, pp. 91–105, 2016. (p. 73)

[213] A. Abhaya and B. K. Patra, "An efficient method for autoencoder based outlier detection," *Expert Systems with Applications*, vol. 213, p. 118904, 2023. (p. 74)

[214] N. R. Prasad, S. Almanza-Garcia, and T. T. Lu, "Anomaly detection: A Survey," *Computers, Materials and Continua*, vol. 14, no. 1, pp. 1–22, 2009. (p. 74)

[215] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys*, vol. 54, no. 2, 2021. (p. 74)

[216] V. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review*, vol. 22, pp. 85–126, oct 2004. (p. 74)

[217] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering*. Cambridge University Press, jan 2019. (p. 74, 76)

[218] E. M. Coraça, J. V. Ferreira, and E. G. Nóbrega, "An unsupervised structural health monitoring framework based on variational autoencoders and hidden markov models," *Reliability Engineering & System Safety*, vol. 231, p. 109025, 2023. (p. 74)

[219] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine Learning Algorithms and Applications*. CRC Press, 2017. (p. 74, 75)

[220] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104743, Apr. 2022. (p. 74)

[221] J. Cutler and M. Dickenson, *Introduction to Machine Learning with Python*, p. 129–142. Springer International Publishing, 2020. (p. 75, 76, 77)

[222] M. Bramer, *Principles of Data Mining*, vol. 30 of *Undergraduate Topics in Computer Science*. London: Springer London, 2020. (p. 75)

[223] P. Gogas and T. Papadimitriou, "Machine Learning in Economics and Finance," *Computational Economics*, vol. 57, no. 1, pp. 1–4, 2021. (p. 75)

[224] J. Shi, "Examine Bitcoin price predictability with machine learning algorithms," in *Third International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI 2022)* (K. Subramanian, ed.), vol. 12509, p. 125092L, International Society for Optics and Photonics, SPIE, 2023. (p. 75)

[225] J. Chen, Y. Gao, J. Shan, K. Peng, C. Wang, and H. Jiang, "Manipulating supply chain demand forecasting with targeted poisoning attacks," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1803–1813, 2023. (p. 75)

[226] Y. Wudil, A. Imam, M. Gondal, U. Ahmad, and M. A. Al-Osta, "Application of machine learning regressors in estimating the thermoelectric performance of bi2te3-based materials," *Sensors and Actuators A: Physical*, vol. 351, p. 114193, 2023. (p. 75)

[227] G. Li, M. Luo, J. Huang, and W. Li, "Early-age concrete strength monitoring using smart aggregate based on electromechanical impedance and machine learning," *Mechanical Systems and Signal Processing*, vol. 186, p. 109865, 2023. (p. 75)

[228] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Computer Science*, vol. 2, no. 3, pp. 1–21, 2021. (p. 76)

[229] P.-N. Tan, M. Steinbach, V. Kumar, and A. Karpatne, *Introduction to Data Mining*. New York: Pearson, 2nd ed ed., 2018. (p. 76)

[230] A. C. Faul, *A Concise Introduction to Machine Learning*. CRC Press, 2019. (p. 76)

[231] B. Kurian and R. Liyanapathirana, "Machine learning techniques for structural health monitoring," in *Proceedings of the 13th International Conference on Damage Assessment of Structures* (M. A. Wahab, ed.), (Singapore), pp. 3–24, Springer Singapore, 2020. (p. 76)

[232] Q. Zhou, Y. Ning, Q. Zhou, L. Luo, and J. Lei, "Structural damage detection method based on random forests and data fusion," *Structural Health Monitoring*, vol. 12, p. 48–58, Nov. 2012. (p. 77)

[233] D. A. Otchere, T. O. Arbi Ganat, R. Gholami, and S. Ridha, "Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ann and svm models," *Journal of Petroleum Science and Engineering*, vol. 200, p. 108182, 2021. (p. 78)

[234] P. A. Flach and N. Lachiche, "Naive bayesian classification of structured data," *Machine Learning*, vol. 57, p. 233–269, Dec. 2004. (p. 80)

[235] S. Russell and P. Norvig, *Artificial intelligence*. Prentice Hall series in artificial intelligence, Upper Saddle River, NJ: Pearson, 2 ed., July 2003. (p. 80)

[236] M. Chen, Q. Liu, S. Chen, Y. Liu, C.-H. Zhang, and R. Liu, "Xgboost-based algorithm interpretation and application on post-fault transient stability status prediction of power system," *IEEE Access*, vol. 7, p. 13149–13158, 2019. (p. 80)

[237] P. Flach, "Performance evaluation in machine learning: The good, the bad, the ugly, and the way forward," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, p. 9808–9814, July 2019. (p. 81)

[238] J. Lever, M. Krzywinski, and N. Altman, "Classification evaluation," *Nature Methods*, vol. 13, p. 603–604, July 2016. (p. 81)

[239] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2nd ed., 2019. (p. 81, 88)

[240] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass confusion matrix reduction method and its application on net promoter score classification problem," *Technologies*, vol. 9, no. 4, 2021. (p. 81, 121, 123)

[241] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, p. 168–192, July 2020. (p. 81, 121, 123)

[242] V. R. Joseph and A. Vakayil, "Split: An optimal method for data splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022. (p. 82)

[243] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Data Mining: An ASA Data Science Journal*, vol. 15, no. 4, pp. 531–538, 2022. (p. 82)

[244] T. H. Stephen Bates and R. Tibshirani, "Cross-validation: What does it estimate and how well does it do it?," *Journal of the American Statistical Association*, vol. 119, no. 546, pp. 1434–1445, 2024. (p. 82)

[245] I. Tougui, A. Jilbab, and J. E. Mhamdi, "Impact of the choice of cross-validation techniques on the results of machine learning-based diagnostic applications," *Healthcare Informatics Research*, vol. 27, p. 189–199, July 2021. (p. 82)

[246] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, p. 022022, feb 2019. (p. 83)

[247] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*, p. 532–538. Springer US, 2009. (p. 83)

[248] D. Shakya, V. Deshpande, M. J. S. Safari, and M. Agarwal, "Performance evaluation of machine learning algorithms for the prediction of particle froude number (frn) using hyper-parameter optimizations techniques," *Expert Systems with Applications*, vol. 256, p. 124960, 2024. (p. 83)

[249] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011. (p. 84, 85)

[250] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. (p. 84)

[251] H. Liang, W. Li, Y. Li, and Y. Li, "Machine learning-based multi-objective optimization and physical-geometrical competitive mechanisms for 3d woven thermal protection composites," *International Journal of Heat and Mass Transfer*, vol. 232, p. 125920, 2024. (p. 84)

[252] L. Tang, P. Si, X. Ling, S. Tian, and W. Zhang, "Machine learning-based predictors for maximum pile bending moment of the soil-pile-superstructure system in liquefiable soils," *Ocean Engineering*, vol. 309, p. 118360, 2024. (p. 84)

[253] A. Tharwat, "Parameter investigation of support vector machine classifier with kernel functions," *Knowl. Inf. Syst.*, vol. 61, p. 1269–1302, dec 2019. (p. 85)

[254] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification And Regression Trees (1st ed.)*. Chapman and Hall/CRC, Oct. 1984. (p. 86)

[255] A. B. Degefa, G. Jeon, S. Choi, J. Bak, S. Park, H. Yoon, and S. Park, "Data-driven insights into controlling the reactivity of supplementary cementitious materials in hydrated cement," *International Journal of Concrete Structures and Materials*, vol. 18, June 2024. (p. 86)

[256] H. Mete Ayhan and S. Kır, "Ml-driven approaches to enhance inventory planning: Inoculant weight application in casting processes," *Computers & Industrial Engineering*, vol. 193, p. 110280, 2024. (p. 86, 87)

[257] I. I. Idris, M. Mousa, and M. Hassan, "Modeling retroreflectivity degradation of pavement markings across the us with advanced machine learning algorithms," *Journal of Infrastructure Preservation and Resilience*, vol. 5, Feb. 2024. (p. 86)

[258] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. (p. 86)

[259] H. V. Ho, "Determination of the surface roller length of hydraulic jumps in horizontal rectangular channels using the machine learning method," *Stochastic Environmental Research and Risk Assessment*, vol. 38, p. 2539–2562, Mar. 2024. (p. 86)

[260] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. (p. 87)

[261] M. Nandal, V. Grover, D. Sahu, and M. Dogra, "Employee attrition: Analysis of data driven models," *EAI Endorsed Transactions on Internet of Things*, vol. 10, Jan. 2024. (p. 87)

[262] M. T. Elshazli, D. Hussein, G. Bhat, A. Abdel-Rahim, and A. Ibrahim, "Advancing infrastructure resilience: machine learning-based prediction of bridges' rating factors under autonomous truck platoons," *Journal of Infrastructure Preservation and Resilience*, vol. 5, Apr. 2024. (p. 87)

[263] V. Q. Tran, H.-V. T. Mai, T.-A. Nguyen, and H.-B. Ly, "Assessment of different machine learning techniques in predicting the compressive strength of self-compacting concrete," *Frontiers of Structural and Civil Engineering*, vol. 16, p. 928–945, July 2022. (p. 88)

[264] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, July 2021. (p. 88)

[265] T. Siddique, M. S. Mahmud, A. M. Keesee, C. M. Ngwira, and H. Connor, "A survey of uncertainty quantification in machine learning for space weather prediction," *Geosciences*, vol. 12, no. 1, 2022. (p. 88, 89, 90)

[266] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021. (p. 89, 90)

[267] R. Izbicki, *Machine Learning Beyond Point Predictions: Uncertainty Quantification*. 1st ed., 2025. (p. 89, 90)

[268] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods," *Machine Learning*, vol. 110, p. 457–506, Mar. 2021. (p. 89)

[269] T. Blasco, J. S. Sánchez, and V. García, "A survey on uncertainty quantification in deep learning for financial time series prediction," *Neurocomputing*, vol. 576, p. 127339, 2024. (p. 89)

[270] W. He, Z. Jiang, T. Xiao, Z. Xu, and Y. Li, "A survey on uncertainty quantification methods for deep learning," 2025. (p. 89)

[271] R. Natras, B. Soja, and M. Schmidt, "Uncertainty quantification for machine learning-based ionosphere and space weather forecasting: Ensemble, bayesian neural network, and quantile gradient boosting," *Space Weather*, vol. 21, no. 10, p. e2023SW003483, 2023. e2023SW003483 2023SW003483. (p. 89, 90)

[272] V. Nemani, L. Biggio, X. Huan, Z. Hu, O. Fink, A. Tran, Y. Wang, X. Zhang, and C. Hu, "Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial," *Mechanical Systems and Signal Processing*, vol. 205, p. 110796, 2023. (p. 89, 90)

[273] Z. Mian, X. Deng, X. Dong, Y. Tian, T. Cao, K. Chen, and T. A. Jaber, "A literature review of fault diagnosis based on ensemble learning," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107357, 2024. (p. 91)

[274] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. Hoboken, NJ: Wiley-Blackwell, 4 ed., Nov. 2010. (p. 91)

[275] R. de Oliveira Teloli, P. Butaud, G. Chevallier, and S. da Silva, "Dataset of experimental measurements for the orion beam structure," *Data in Brief*, vol. 39, p. 107627, 2021. (p. 113, 114)

[276] C. Zang, M. I. Friswell, and M. Imregun, "Structural health monitoring and damage assessment using measured frfs from multiple sensors, part i: The indicator of correlation criteria," in *Damage Assessment of Structures V*, vol. 245 of *Key Engineering Materials*, pp. 131–140, Trans Tech Publications Ltd, 7 2003. (p. 115)

[277] M. Dallali, L. Khalij, E. Conforto, A. Dashti, C. Gautrelet, M. Rodrigues Machado, and E. Souza de Cursi, "Effect of geometric size deviation induced by machining on the vibration fatigue behavior of ti-6al-4v," *Fatigue & Fracture of Engineering Materials & Structures*, vol. 45, no. 6, pp. 1784–1795, 2022. (p. 115)

[278] B. B. de Moura, M. R. Machado, T. Mukhopadhyay, and S. Dey, "Dynamic and wave propagation analysis of periodic smart beams coupled with resonant shunt circuits: passive property modulation," *The European Physical Journal Special Topics*, vol. 231, p. 1415–1431, Mar. 2022. (p. 115)

[279] B. de Moura, M. Machado, S. Dey, and T. Mukhopadhyay, "Manipulating flexural waves to enhance the broadband vibration mitigation through

inducing programmed disorder on smart rainbow metamaterials," *Applied Mathematical Modelling*, vol. 125, pp. 650–671, 2024. (p. 115, 127)

[280] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, p. 1761–1776, Aug. 2011. (p. 122)

[281] M.R.Machado, B. Moura, S. Dey, and T. Mukhopadhyay, "Bandgap manipulation of single and multi-frequency smart metastructures with random impedance disorder," *Smart Materials and Structures*, vol. 31, p. 105020, 2022. (p. 127)

[282] T. Buckley, B. Ghosh, and V. Pakrashi, "A feature extraction & selection benchmark for structural health monitoring," *Structural Health Monitoring*, vol. 22, no. 3, pp. 2082–2127, 2023. (p. 128)

[283] R. E. Nogales and M. E. Benalcázar, "Analysis and evaluation of feature selection and feature extraction methods," *International Journal of Computational Intelligence Systems*, vol. 16, p. 153, Sep 2023. (p. 130)

ABSTRACT

STRUCTURAL HEALTH MONITORING OF WIND TURBINES COMPONENTS USING MACHINE LEARNING

M.Sc. Jefferson da Silva Coelho

Keywords: Unsupervised-supervised machine learning, Damage index, Damage detection, Uncertainty quantification.

Wind turbines are complex electromechanical systems that require continuous monitoring to ensure operational efficiency, minimise costs, and prevent critical failures. ML has shown great promise in SHM by automating defect detection through data-driven methods. Vibration-based ML techniques are particularly effective for monitoring turbine components such as blades, towers, and gearboxes. However, challenges persist in adapting SHM methods to complex environmental conditions and ensuring reliable monitoring and failure detection. The objective of this work is to propose a data-driven SHM-ML methodology designed for pattern recognition, damage detection, and quantification in wind turbine components. Three case studies were proposed to validate the SHM-ML approach, comprising supervised regression and classification models, feature extraction techniques and data augmentation to improve the robustness and reliability of the models. The first study monitored and evaluated three failure events during the operation of a real wind turbine using the acceleration time spectrum as raw monitoring data. The second case focused on the detection and classification of torque loosening in bolted joints based on frequency domain spectral signals from experimental tests, combining supervised and unsupervised techniques with a damage index derived from the dynamic response. The third case integrates regression algorithms with data augmentation techniques to enhance an accurate estimate of torque loosening using raw vibration spectra in bolted structures. The results demonstrated high accuracy in the

estimation and classification of damage, validating the effectiveness of the SHM-ML methodology developed. These findings contribute to the advancement of data-driven approaches to wind turbine SHM, increasing reliability and operational safety.

STRESZCZENIE

MONITOROWANIE STANU TECHNICZNEGO TURBIN WIATROWYCH Z WYKORZYSTANIEM UCZENIA MASZYNOWEGO

M.Sc. Jefferson da Silva Coelho

Słowa kluczowe: Uczenie maszynowe nadzorowane i nienadzorowane, wskaźnik uszkodzenia, wykrywanie uszkodzeń, ocena niepewności.

Turbiny wiatrowe to złożone systemy konstrukcyjno - budowlane, które wymagają ciągłego monitorowania w celu zapewnienia efektywności użytkowej, minimalizacji kosztów oraz zapobiegania awariom. Są to konstrukcje wieżowe o znacznej wysokości i smukłości, znajdujące się pod działaniem złożonych obciążeń aerodynamicznych, wynikających z działania wiatru. Obciążenia działające na tego typu konstrukcje dodatkowo zmieniają się w sytuacjach oblodzenia i zmian temperatury. Uczenie maszynowe (z ang. Machine Learning - ML) wykazuje duży potencjał w monitorowaniu stanu technicznego konstrukcji (z ang. Structural Health Monitoring – SHM), w szczególności wieżowych, poprzez wykrywanie uszkodzeń przy pomocy metod wykorzystujących dane pomiarowe. Techniki ML oparte na analizie drgań są szczególnie skuteczne w monitorowaniu elementów turbin wiatrowych, takich jak łopaty, wieża i przekładnia. Nieustanny rozwój technologii turbin wiatrowych, zmienne warunki środowiskowe, w których turbiny się znajdują sprawiają, iż nadal istnieje potrzeba rozwijania metod związanych z monitorowaniem konstrukcji w celu niezawodnej i bezawaryjnej pracy turbin. Celem niniejszej pracy jest opracowanie metodyki uczenia maszynowego opartego na pomierzonych parametrach pracy i odpowiedzi turbin wiatrowych na działające obciążenia. W proponowanych rozwiązaniach algorytmicznych stosowane są rozpoznawanie wzorców, rozpoznawanie uszkodzeń oraz ocena uszkodzeń elementów turbin wiatrowych. Zastoso-

wano modele nadzorowanej regresji i klasyfikacji, w których przewidywano wartości liczbowe na podstawie oznaczonych danych wyjściowych, a także wykorzystano metody przekształcania surowych danych w zestaw istotnych informacji wykorzystywanych w algorytmie uczenia maszynowego. W celu poprawy wydajności i niezawodności modelu zastosowano metodę rozszerzania danych (data augmentation). W pracy analizowano trzy przypadki związane z zastosowaniem algorytmów monitorowania stanu technicznego konstrukcji turbin wiatrowych. W pierwszej analizie przeprowadzono klasyfikację uszkodzeń turbiny wiatrowej podczas jej pracy. Analiza obejmowała monitorowanie i ocenę trzech zdarzeń awaryjnych podczas pracy turbiny wiatrowej Aventa, wykorzystując jako surowe dane monitorujące widmo czasowe. W drugim przypadku zastosowany algorytm pozwalał na wykrywanie zmian momentu dokręcenia połączenia śrubowego. Połączenia śrubowe są bardzo istotne w konstrukcji turbin wiatrowych, gdyż segmenty stalowej wieży turbiny wiatrowej łączone są przy pomocy kołnierzowego połączenia śrubowego. W celu określenia zmian momentu dokręcenia zastosowano sygnały spektralne w dziedzinie częstotliwości. Dane pochodziły z badań laboratoryjnych. W tym przypadku łączono techniki nadzorowane i nienadzorowane. Na podstawie odpowiedzi dynamicznej określono wskaźniki uszkodzenia. Trzeci przypadek obejmuje ocenę zmiany momentu dokręcenia łączników. Zastosowano algorytm regresyjny w połączeniu z metodą augmentacji danych w celu dokładniejszego oszacowania luzowania momentu dokręcenia na podstawie widma drgań. Uzyskane wyniki wykazały wysoką dokładność w ocenie i klasyfikacji uszkodzeń, potwierdzając skuteczność opracowanej metodyki monitorowania konstrukcji z zastosowaniem uczenia maszynowego. Uzyskane rezultaty przyczyniają się do rozwoju metod opartych na analizie danych pomiarowych do monitorowania stanu technicznego turbin wiatrowych, zwiększając ich niezawodność i bezpieczeństwo użytkowania. Proponowane metody są szczególnie efektywne w ocenie stanu technicznego konstrukcji i w przyszłości staną się podstawowym narzędziem eksperckim w identyfikacji zmian konstrukcji budowlanych.

RESUMO

MONITORAMENTO DA INTEGRIDADE ESTRUTURAL DE COMPONENTES DE TURBINAS EÓLICAS UTILIZANDO APRENDIZADO DE MÁQUINA

M.Sc. Jefferson da Silva Coelho

Palavras-chave: Aprendizado de máquina supervisionado e não supervisionado, índice de danos, detecção de danos, quantificação de incertezas.

As turbinas eólicas são sistemas eletromecânicos complexos que exigem monitoramento contínuo para garantir a eficiência operacional, minimizar os custos e evitar falhas críticas. O ML tem se mostrado muito promissor no SHM, automatizando a detecção de defeitos por meio de métodos orientados por dados. As técnicas de ML baseadas em vibração são particularmente eficazes para monitorar componentes de turbinas, como lâminas, torres e caixas de engrenagens. No entanto, ainda há desafios para adaptar os métodos SHM a condições ambientais complexas e garantir o monitoramento confiável e a detecção de falhas. O objetivo deste trabalho é propor uma metodologia de algoritmo de ML orientada por dados projetada para reconhecimento de padrões, detecção e quantificação de danos em componentes de turbinas eólicas. Três estudos de caso foram propostos para validar a abordagem SHM-ML. A abordagem proposta aproveitou os modelos de regressão e classificação supervisionados, as técnicas de extração de recursos e o aumento de dados para melhorar a robustez e a confiabilidade dos modelos. O primeiro estudo monitorou e avaliou três eventos de falha durante a operação de turbina eólica usando o espectro de tempo de aceleração como dados brutos de monitoramento. O segundo caso concentrou-se na detecção e classificação do afrouxamento de torque em juntas aparafusadas com base em sinais espectrais de domínio de frequência de testes experimentais, combinando técnicas supervisionadas e não supervisionadas com

um índice de danos derivado da resposta dinâmica. O terceiro caso integra algoritmos de regressão com técnicas de aumento de dados para melhorar a estimativa precisa do afrouxamento do torque usando espectros de vibração brutos em estruturas aparafusadas. Os resultados demonstraram alta precisão na estimativa e classificação de danos, validando a eficácia da metodologia SHM-ML desenvolvida. Essas descobertas contribuem para o avanço das abordagens orientadas por dados para o SHM de turbinas eólicas, aumentando a confiabilidade e a segurança operacional.

Publications

1. Peer-reviewed journal publications

    - **Coelho, J.S.**; Machado, M.R.; Sousa, A.A.S.R. PyMLDA: A Python open-source code for Machine Learning Damage Assessment, Software Impacts, Volume 19, 2024, ISSN 2665-9638. https://doi.org/10.1016/j.simpa.2024.100628.
    - **Coelho, J.S.**; Machado, M.R.; Dutkiewicz, M.; Teloli, R.O. Data-driven machine learning for pattern recognition and detection of loosening torque in bolted joints. J. Braz. Soc. Mech. Sci. Eng. 46, 75 (2024). https://doi.org/10.1007/s40430-023-04628-6.
    - Sousa A. A. S. R.; **Coelho, J. S.**; Machado M. R.; Dutkiewicz M. Multiclass Supervised Machine Learning Algorithms Applied to Damage and Assessment Using Beam Dynamic Response. J. Vib. Eng. Technol. 11, 2709–2731 (2023). https://doi.org/10.1007/s42417-023-01072-7.
    - **Coelho, J. S.**; Machado M. R.; Dutkiewicz M. Integrating Virtual Sensor Data Augmentation into Machine Learning for Damage Quantification of Bolted Structures under Assembly Uncertainty,2025, Under review.

2. Patent request, Registration of Computer Programs

    - Machado, M.R. ; **Coelho, J.S.** ; Sousa, A.A.S.R. Structural Integrity Monitoring Based on Machine Learning Techniques. 2024, Brasil. Patente: Privilégio de Inovação. Número do registro: BR1020240152867, título: "Método para o Monitoramento de Integridade Estrutural Baseado em Técnicas de Aprendizado de Máquina" , Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial. Depósito: 25/07/2024
    - Machado, M.R. ; **Coelho, J.S.** ; Sousa, A.A.S.R. PyMLDA - Machine Learning for Damage Assessment. 2024. Patente: Programa

de Computador. Número do registro: BR512024001008-4, data de registro: 17/01/2024, título: ”PyMLDA - Machine Learning for Damage Assessment” , Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial.

3. Publications in conference proceedings, Congress and Symposium:

- **Coelho, J.S.**; Machado, M.R.; Dutkiewicz, M. Estimation of loosening torque in bolted joints from experimental data and regression models. In: XIVth International Conference on Recent Advances in Structural Dynamics, 2024, Southhampton. Journal of Physics: Conference Series, 2024. DOI 10.1088/1742-6596/2909/1/012016.
- Sousa, A.A.S.R. ; Machado, M.R.; Teloli, R.O.; **Coelho, J. S.**. Structural Monitoring of Composite CFRP Plate Under Temperature Variation Using the Machine Learning - PYMLDA Code. In: 7 th Brazilian Conference on Composite Materials, 2024, Brasília. 7th Brazilian Conference on Composite Materials Proceedings, 2024.
- Dutkiewicz, M. ; Machado, M. R. ; **Coelho, J. S.**. Spectral Method and Machine Learning Approach to Wind Turbine Damage Detection. In: XLIV Ibero-Latin American Congress on Computational Methods in Engineering, 2023, Porto-Portugal. XLIV Ibero-Latin American Congress on Computational Methods in Engineering, 2023. v. 5. https://publicacoes.softaliza.com.br /cilamce2023/article/view/4992.
- Sousa, A. A. S. R. ; **Coelho, J. S.** ; Machado, M. R. ; Dutkiewicz, M. A Hybrid Learning Model for Assessment Beam Damage Detection. In: XLIV Ibero-Latin American Congress on Computational Methods in Engineering, 2023, Porto-Portugal. XLIV Ibero-Latin American Congress on Computational Methods in Engineering, 2023. v. 5. https://publicacoes.softaliza.com.br /cilamce2023/article/view/5076.

- **Coelho J. S.**, Sousa A. A S. R., Machado M. R., Dutkiewicz M. Bolt loosening Detection Based in Data-Driven of Bolted Beam Connections by Support Vector Machine Method In: XIX International Symposium on Dynamic Problems of Mechanics, 2023, Pirenópolis-GO. Proceedings of the XIX International Symposium on Dynamic Problems of Mechanics. ABCM, 2023, DOI://10.26678/ABCM.DINAME2023.DIN2023-0111.
- Sousa A. A. S. R., **Coelho J. S.**, Machado M. R., Dutkiewicz M. Assessment of Damage in a Reinforced Beam by K-Nearest Neighbors Algorithm In: XIX International Symposium on Dynamic Problems of Mechanics, 2023, Pirenópolis-GO. Proceedings of the XIX International Symposium on Dynamic Problems of Mechanics. ABCM, 2023, DOI://10.26678/ABCM.DINAME 2023.DIN2023-0081.
- **Coelho, J.S**, de Sousa, A.A.S.R., Machado, M.R., Dutkiewicz, M. (2023). Study of Machine Learning Techniques for Damage Identification in a Beam. In: Dimitrovová, Z., Biswas, P., Silva, T., Gonçalves, R. (eds) Recent Trends in Wave Mechanics and Vibrations. WMVC 2022. Mechanisms and Machine Science, vol 125. Springer, Cham. https://doi.org/10.1007/978-3-031-15758-5_84
- **Coelho J. S.**, Sousa A. A. S. R., Machado M. R., Jorge A. B. Spectral Model and Vibration Analysis of Beams Connected by Bolts-Joint. In: 26th International Congress of Mechanical Engineering, 2021, Virtual. Proceedings of the 26th International Congress of Mechanical Engineering., 2021.
- Sousa, A. A. S. R., **Coelho J. S.**, Machado M. R., Jorge, A. B. Vibration Characterization of Beams Riveted Lap Joint Using Spectral Element Method. In: 26th International Congress of Mechanical Engineering, 2021, Virtual. Proceedings of the 26th International Congress of Mechanical Engineering., 2021.