Hindawi
Structural Control and Health Monitoring
Volume 2023, Article ID 6646599, 16 pages
https://doi.org/10.1155/2023/6646599

*Research Article*

# Damage Detection and Localization at the Jacket Support of an Offshore Wind Turbine Using Transformer Models

**Héctor Triviño**,[1] **Cisne Feijóo**,[1] **Hugo Lugmania**,[2] **Yolanda Vidal**,[3,4] **and Christian Tutivén**[1]

[1]*Mechatronics Engineering, Faculty of Mechanical Engineering and Production Science, Escuela Superior Politécnica del Litoral, ESPOL, Campus Gustavo Galindo Km. 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador*
[2]*Mecatrónica, Facultad de Ingenierías, Universidad ECOTEC, Samborondón, EC092303, Ecuador*
[3]*Control, Data, and Artificial Intelligence, CoDAlab, Department of Mathematics, Escola d'Enginyeria de Barcelona Est, EEBE, Universitat Politècnica de Catalunya, UPC, Campus Diagonal-Besós (CDB), Barcelona 08019, Spain*
[4]*Institut de Matemàtiques de la UPC, BarcelonaTech, IMTech, Pau Gargallo 14, Barcelona 08028, Spain*

Correspondence should be addressed to Yolanda Vidal; yolanda.vidal@upc.edu

Received 12 June 2023; Revised 30 November 2023; Accepted 19 December 2023; Published 30 December 2023

Academic Editor: Andrea Del Grosso

Early detection of damage in the support structure (submerged part) of an offshore wind turbine is crucial as it can help to prevent emergency shutdowns and extend the lifespan of the turbine. To this end, a promising proof-of-concept is stated, based on a transformer network, for the detection and localization of damage at the jacket-type support of an offshore wind turbine. To the best of the authors' knowledge, this is the first time transformer-based models have been used for offshore wind turbine structural health monitoring. The proposed strategy employs a transformer-based framework for learning multivariate time series representation. The framework is based on the transformer architecture, which is a neural network architecture that has been shown to be highly effective for natural language processing tasks. A down-scaled laboratory model of an offshore wind turbine that simulates the different regions of operation of the wind turbine is employed to develop and validate the proposed methodology. The vibration signals collected from 8 accelerometers are used to analyze the dynamic behavior of the structure. The results obtained show a significant improvement compared to other approaches previously proposed in the literature. In particular, the stated methodology achieves an accuracy of 99.96% with an average training time of only 6.13 minutes due to the high parallelizability of the transformer network. In fact, as it is computationally highly efficient, it has the potential to be a useful tool for implementation in real-time monitoring systems.

## 1. Introduction

In recent years, the use of renewable energy has become a global necessity due to growing concern about climate change and the depletion of fossil fuels [1]. According to [2], in 2021, investment in renewable energy and fuels increased for the fourth consecutive year, reaching USD 366 billion, achieving for the first time in history that solar and wind energy provide more than 10% of world electricity.

Due to its low operating cost, reduced environmental impact and the possibility of generating electricity in remote areas and without access to the conventional electrical grid, wind energy has become an important and promising way of generating clean and sustainable energy [3]. In 2021, wind electricity generation increased by a record 273 TWh, which was 55% higher growth than that achieved in 2020 and became the highest among all renewable energy technologies [4]. However, despite the significant growth that is taking place, GWEC market intelligence forecasts that by 2030 less than two-thirds of the wind power capacity required for the 1.5°C net-zero pathway established by the International Renewable Energy Agency (IRENA) in its 2050 roadmap [5].

Therefore, to achieve the proposed objectives, greater efforts must be made to accelerate the growth of energy generation using this resource.

With technological development and the ongoing search for improvements that allow increased electrical generation capacities and reduced installation and operating costs, the offshore wind industry has been gaining strength and apogee in the sector. Its great capacity to generate large amounts of energy, due to the strong wind speeds and availability at sea [6], has generated greater growth in recent years in this type of offshore wind farms compared to onshore. Making 2021 the best year in the history of offshore wind power, bringing 21.1 GW of capacity into service, three times more than in 2020 [7].

Despite the many advantages of offshore wind farms, the installation and operation of these farms pose particular challenges compared to onshore wind farms, due to several factors such as the extreme environmental conditions to which they are exposed (wind, waves, and currents) [8], restrictions or difficulties in access, and a larger size and weight of the structures, making maintenance difficult. One of the most complex and critical components of offshore WTs is the base structure, as its installation and design entails greater technical challenges that allow the structure to be resistant to the prolonged impact of large loads generated in the marine environment [9]. However, maintenance management is a critical task in the offshore wind energy industry, with a direct impact on the profitability of wind projects and constitutes an important part of operating and maintenance costs [10]. One of the best known alternatives to facilitate this type of management is the use of structural health monitoring (SHM) strategies.

A predictive maintenance strategy allows monitoring of the integrity of structural components by means of specific sensors to prevent catastrophic failures, with serious consequences for the safety of personnel and the environment [11]. Currently, this strategy is already widely used in large civil infrastructures [12, 13] and aerospace structures [14, 15]. In addition, its use is expanding to other fields such as energy sectors [16, 17], among which wind farms stand out, where early detection of damage can prevent or reduce long downtime, emergency shutdowns, and costly maintenance in wind turbines (WTs), helping to prolong its useful life [18].

The vibration of the machine is a key element for the analysis of the state of a piece of equipment or an element subjected to great loads or extreme conditions, so its analysis is an effective and reliable technique, with a nondestructive nature, which allows maintaining a sustainable monitoring without interfering in the processes [19]. This allows determining the existence of structural damage, evaluating the safety of the base structure of a WT, predicting useful life and making decisions about maintenance strategy if necessary [20]. That is why the development of vibration-based SHM methodologies has been gaining relevance and covering large fields because of its great advantage of real-time monitoring of dynamic characteristics. For this reason, vibration-based SHM strategies are used in this work.

In recent years, the application of machine learning techniques has become one of the most used tools to improve the capacity of SHM systems, since, through data analysis and processing, it allows the creation of more accurate, cost-effective, and reliable models or algorithms [21]. Currently, many models oriented to the detection, classification, and even localization of different types of damage in WTs, have been developed, based on the analysis of vibration signals, such as in [22, 23], where methodologies based on a Siamese convolutional neural network and autoencoder are developed, respectively, for the detection of damage to the WT jacket structure, or on [24] where regression models are developed based on the identification of the modal properties of the most important modes of vibration through the analysis of three common damage scenarios: onshore foundation damage; damage by scour on an offshore foundation; and blade damage. Other related studies are shown in [25–27] where damage to the blades, gearbox, and bearings of WTs are analyzed.

In the aforementioned works, despite the presentation of effective solutions using traditional neural networks, they present some disadvantages, such as high computing cost (that does not allow real-time on-line monitoring) and difficulty in capturing long-term time-series dependencies. However, in 2017, Google researchers, led by Ashish Vaswani presented a new type of model called transformers in their paper called "*Attention is all you need*" [28]. This new model solves the abovementioned problems.

Initially, the transformer deep learning model focused on improving the quality of language translation, proposing an architecture that left aside the use of recurrent neural networks (RNNs) and relied solely on attention mechanisms, achieving greater efficiency and accuracy compared to traditional models. Due to their high performance, their ease of modeling long dependencies between input sequence elements and their ability to process data in parallel [29], these deep learning models have been extended to other fields such as natural language processing (NLP), computer vision (CV), and audio processing [30].

One of its new applications is focused on the analysis of multivariate time series, as shown in [31–33], where it has shown great performance, since it allows analyzing long time series of data and detecting complex patterns in time series more accurately and efficiently than conventional models such as those based on RNN and convolutional neural networks (CNNs).

In this study, for the first time, an SHM methodology based on a transformer model is presented for the detection and localization of different types of structural damage at different levels in the jacket structure of an offshore WT. The objective of this work is to improve the efficiency and safety in the operation and maintenance of WTs, reducing operational and maintenance costs.

While vibration-based SHM and machine learning have been applied to detect various forms of damage in wind turbines, these prior studies have predominantly focused on the blades and tower damage. There has been comparatively far less research on using these techniques to detect crack or missing bolts in offshore wind turbine support structures. In fact, these types of damage may exhibit subtler vibration patterns that conventional neural networks may not effectively capture. In addition, multivariate sensor data require efficient modeling of long-term temporal dynamics. To date, no study has used transformer neural network models, designed specifically for sequential data, for detecting damage in offshore wind turbine jackets. This represents a key research gap that the current work addresses.

Therefore, the objective of this study is to develop and evaluate a transformer neural network model for detecting and localizing damage in an offshore wind turbine jacket support structure.

This document is organized as follows. In Section 2, the experimental setup is presented. To address the proposed research gap, experimental vibration data are needed from an offshore wind turbine jacket structure exhibiting various structural states (healthy and different types of damage). The data acquisition process is presented in Section 3. The preprocessing of the data and the development of the damage detection methodology are explained in detail in Section 4. The results are presented and discussed in Section 5. Finally, the conclusions are detailed in Section 6.

## 2. Experimental Setup

To test the hypothesis that a transformer model can effectively detect damage, vibration data are collected from a down-scaled experimental setup under controlled conditions. For this study, a scale replica of an offshore jacket-type WT is used, which has a height of 2.7 m and is composed of three main parts. The first part, called the nacelle, is the upper component of the WT, and consists of a 1 m long and 0.6 m wide bar, and at its left end an inertial shaker model Data Physics GW-IV47 fed by various excitation signals. These signals are sent from a function generator (GW INSTEK AF-2005) that provides different white noise amplitudes (factors of 0.5, 1, 2, and 3) to mimic the wind speeds of different WT operating regions. The second part is the tower that has been divided into three pieces, which are joined by flanges. The tower has a length of 1.67 m and a diameter of 0.239 m. The third part corresponds to the jacket, located in the lower part and composed of 32 steel bars of S275JR steel, and DC01 LFR steel plates, bolts, and nuts adjusted to a torque equal to 12 Nm. In particular, the S275JR steel is a structural steel grade specified in the EN 10025-2 standard for flat and hot-rolled products. The "*S*" denotes it is a structural steel, the "275" indicates the minimum yield strength of 275 MPa, and the "JR" signifies it undergoes an impact test at room temperature. For structural applications like offshore wind turbine supports, the S275JR grade provides a cost-effective option with sufficient corrosion resistance. The strength allows supporting significant dynamic loads while the ductility provides damage tolerance. The DC01 LFR grade corresponds to a low carbon steel specified in the EN 10130 standard for cold-rolled steel. The "DC01" indicates it is an uncoated deep drawing steel with a minimum yield strength of 140 MPa. The "*L*" signifies it has a low carbon content (< 0.12%). The DC01 LFR grade combines the easy processing of mild steels with increased strength from cold forming. For the jacket structure, the DC01 LFR plates serve as connection joints, where the ductility is beneficial to withstand cyclic loading and environmental factors.

The jacket structure has four levels, where at each level the length of the bars varies, taking into account that level 4 contains the shortest bars and level 1 the longest bars, as shown in Figure 1.

To perform an analysis of the dynamic behavior of the structure based on the measurement of vibrations at different frequencies or amplitudes, eight triaxial accelerometers (PCB R Piezotronic, model 356A17) are used, distributed throughout the WT structure, as illustrated in Figure 2. The optimal location of the sensors is determined according to the sensor elimination by modal assurance criterion (SEAMAC) as comprehensively stated, for this particular down-scaled replica, in Zugasti's PhD thesis [34] (Chapter 3.7, page 53). This is a sensor removal algorithm based on eliminating iteratively, one by one, the degrees of freedom that show a lower impact on the modal assurance criterion (MAC) [35] matrix values. The SEAMAC iterative process stops when you get a default MAC matrix with high values in the diagonal terms and low values in off-diagonal terms. Since the sensors are triaxial, a total of 24 vibration signals are obtained. For the acquisition and processing of the signals measured by each of the accelerometers, six input modules NI 9234 model National Instruments are used, located in a chassis (cDAQ model). For more information on the experiment setup, see [36].

This work seeks to develop a strategy capable of detecting different types of damage and their location at the jacket support. Thus, the cases studied include different types of damage at different locations (levels) of the WT jacket (levels 1, 2, 3, and 4). The different damage scenarios are the following:

(i) Original bar in healthy state.
(ii) Bar with a 5 mm crack.
(iii) Bar with a missing bolt.
(iv) A pristine replica bar is also considered.

Controlled damage is introduced by machining cracks of 5 mm length into selected bars to simulate fatigue cracking. Missing bolts are simulated by removing the bolt at specified joints. These scenarios aimed to replicate common fatigue and joint defects that can develop in offshore structures due to cyclic loading and inadequate maintenance. Cracks and loose connections are prevalent damage modes in jacket joints that must be detected. In Figure 3, it can be observed the three different bar states used in this work. Figure 3(a) shows the bar with the crack, where **L** is the length of the bar, $X = L/3$ is the distance the crack is from the left end of the bar, and **d** is the size of the fissure. Figure 3(b) shows the bar detailing the position of the missing bolt, and finally Figure 3(c) shows the replica with a bar of similar characteristics of the original.

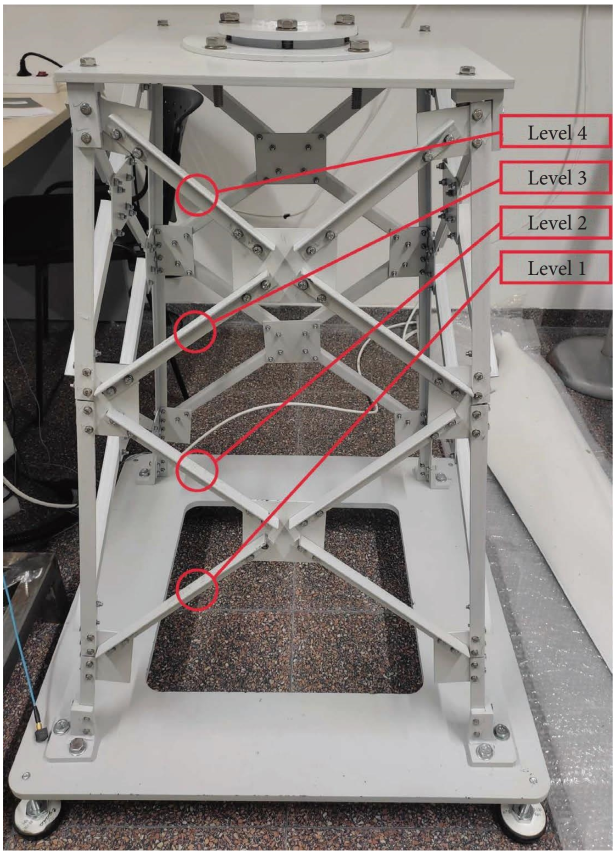


Figure 1: The different levels of the jacket support.

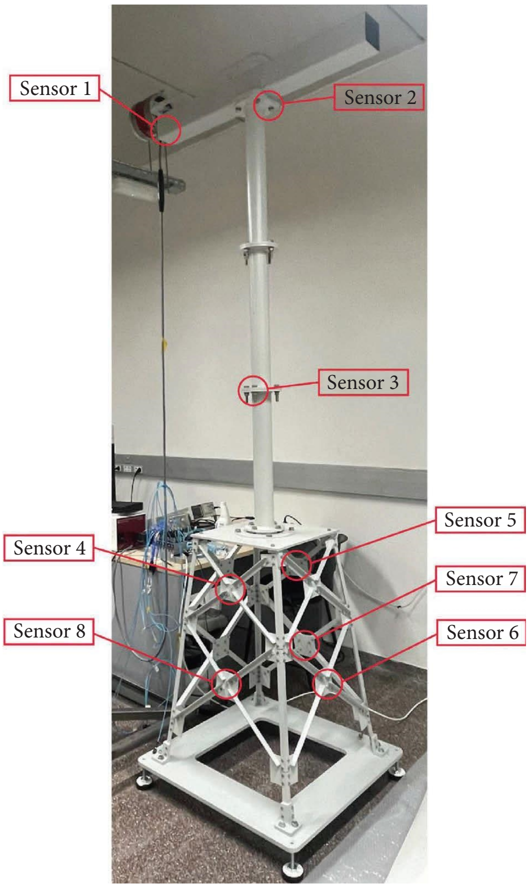


Figure 2: Location of the sensors (accelerometers) in the structure.

## 3. Data Acquisition

This section presents the data acquisition process of the different experiments carried out by simulating different conditions and structural states.

A total of 1140 experiments are performed, of which 180 are with the completely healthy structure and the rest simulating the different damage structural states. Table 1 shows in more detail the different structural states analyzed and the distribution of the experiments carried out according to the white noise amplitude factors. It is important to emphasize that each structure state is simulated at four different levels of the jacket.

The duration of each experiment is 60 seconds at a sampling frequency of 330 Hz, so in each experiment 330 Hz × 60 s = 19800 measurements are obtained for each sensor. Since 8 triaxial accelerometers are used, the number of signals received during each experiment is equal to 8 × 3 = 24 signals. A sampling frequency of 330 Hz is selected for the accelerometers, as this rate falls within the typical range employed in offshore structural monitoring. Sampling rates between 50 Hz [37] and 1000 Hz [38] are generally employed on offshore supports. Although lower sampling rates of around 100 Hz may be sufficient for monitoring the primary platform vibration modes, a rate of 330 Hz is chosen to allow the detection of abnormal high-frequency motions that could indicate developing faults or damage. This higher rate provides broader frequency coverage without excessive data volumes. Finally, the experiments are conducted for a duration of 60 seconds, adhering to standard practice in the Condition Monitoring (CM) system of wind turbines, where it is customary to acquire data for a limited time frame of minutes. Finally, the data associated to the $k$-th experiment are stored in matrix $\mathbf{S}^{(k)}$ with coefficients $s_{n,m}^{(k)}$ $(n = 1, \ldots, N, m = 1, \ldots, M)$ that reads as

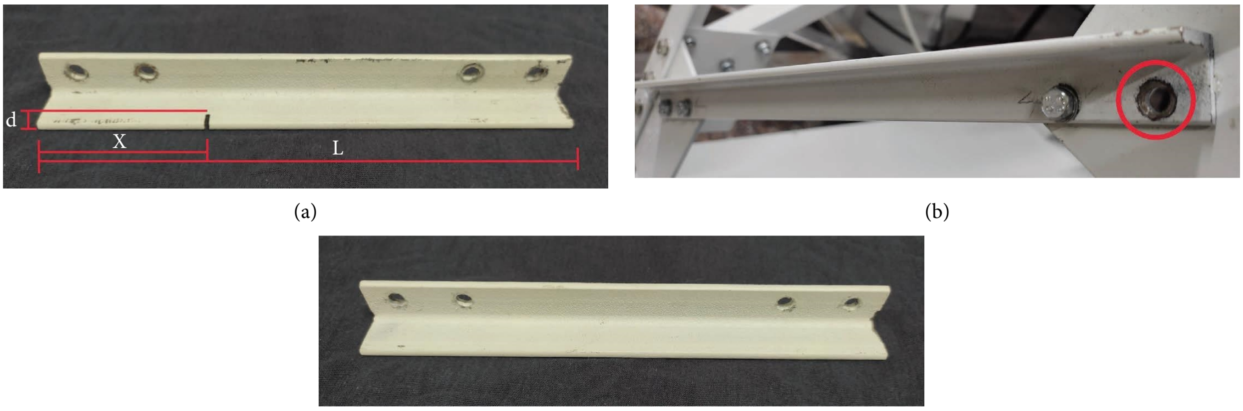
$$\mathbf{S}^{(k)} = \begin{bmatrix} s_{1,1}^{(k)} & s_{1,2}^{(k)} & \cdots & s_{1,M}^{(k)} \\ s_{2,1}^{(k)} & s_{2,2}^{(k)} & \cdots & s_{2,M}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N,1}^{(k)} & s_{N,2}^{(k)} & \cdots & s_{N,M}^{(k)} \end{bmatrix}, \quad (1)$$

considering $k \in [1, K]$, where $K$ is 1140. The two subindices, in the matrix coefficients, are related to the time instant (row) and sensor (column), respectively. More precisely,

(i) $n = 1, \ldots, N$ identifies the time stamp, while $N$ is the number of time stamps per experiment, equal to 19800;

(ii) $m = 1, \ldots, M$ represents the measuring sensor, while $M$ is the total number of sensors, equal to 24.

As a result, each experiment matrix $\mathbf{S}_{N,M}^{(k)} \in \mathbb{M}_{19800\times 24}(\mathbb{R})$.



(a)

(b)

(c)

Figure 3: Different structural states used to perform the experiments. (a) Bar with a 5 mm crack. (b) Bar with missing bolt. (c) Replica bar.

Table 1: Number of experiments distributed according to the structural state of the bar and the white noise (WN) amplitude.

| (Label) structural state | Number of experiments | | | | |
|---|---|---|---|---|---|
| | 0.5 WN | 1 WN | 2 WN | 3 WN | Total |
| (1) Crack level 1 | 20 | 20 | 20 | 20 | 80 |
| (2) Crack level 2 | 20 | 20 | 20 | 20 | 80 |
| (3) Crack level 3 | 20 | 20 | 20 | 20 | 80 |
| (4) Crack level 4 | 20 | 20 | 20 | 20 | 80 |
| (5) Bolt level 1 | 20 | 20 | 20 | 20 | 80 |
| (6) Bolt level 2 | 20 | 20 | 20 | 20 | 80 |
| (7) Bolt level 3 | 20 | 20 | 20 | 20 | 80 |
| (8) Bolt level 4 | 20 | 20 | 20 | 20 | 80 |
| (9) Replica level 1 | 20 | 20 | 20 | 20 | 80 |
| (10) Replica level 2 | 20 | 20 | 20 | 20 | 80 |
| (11) Replica level 3 | 20 | 20 | 20 | 20 | 80 |
| (12) Replica level 4 | 20 | 20 | 20 | 20 | 80 |
| (13) Healthy | 45 | 45 | 45 | 45 | 180 |
| Total | 285 | 285 | 285 | 285 | 1140 |

## 4. Methodology for Damage Detection and Localization

First, this section details the preprocessing of the data, where emphasis is placed on the feature engineering, the division of data into training, validation, and test sets, and data normalization. Subsequently, the background and architecture of the transformer's model based on multivariate time series for the detection and localization of damage is described.

### 4.1. Data Preprocessing.

*4.1. Data Preprocessing.* Data preprocessing is composed of a series of data analysis techniques that improve the quality of a data set in order to obtain the most relevant information [39], helping to facilitate training and improve the accuracy of a model.

*4.1.1. Feature Engineering: Data Reshape.* Feature engineering refers to the process of constructing valuable features as input to the model. Some of the most commonly used techniques in feature engineering for machine learning are identification of new data sources, application of new business rules, or data reshaping [40].

In this work, the raw data consist of individual timestamps for each sensor, which provide limited contextual information to the model. Using individual data points rather than sequences removes important temporal relationships in the signals. Reshaping the data into multivariate time series allows the model to analyze patterns and dynamics over a sequence length relevant for damage detection. A multivariate sequence structure allows the transformer model to effectively capture long-term dependencies and interactions between the multiple sensor channels. In particular, when examining the initial obtained matrices from the data acquisition system (see (1)), it can be observed that, in each sample (row), there is only one data (time stamp) per sensor, which may be scarce or insufficient for model development. Therefore, to increase the information of each sensor in each sample (row), a data reshaping is applied. Considering a detection time of

0.5 seconds, since the sampling frequency is 330 Hz, a total of $\omega = 165$ time stamps per sensor (multivariate sequences) is selected to describe a sample that meets the desired detection time. Data reshape is taken on each matrix $\mathbf{S}^{(k)}$, and the new reshaped matrix is denoted as $\mathbf{X}^{(k)}$ and reads as follows:

$$\mathbf{X}^{(k)} = \begin{bmatrix} \left[s_{1,1}^{(k)} \cdots s_{165,1}^{(k)}\right] & \left[s_{1,2}^{(k)} \cdots s_{165,2}^{(k)}\right] & \cdots & \left[s_{1,24}^{(k)} \cdots s_{165,24}^{(k)}\right] \\ \left[s_{166,1}^{(k)} \cdots s_{330,1}^{(k)}\right] & \left[s_{166,2}^{(k)} \cdots s_{330,2}^{(k)}\right] & \cdots & \left[s_{166,24}^{(k)} \cdots s_{330,24}^{(k)}\right] \\ \vdots & \vdots & \ddots & \vdots \\ \left[s_{19636,1}^{(k)} \cdots s_{19800,1}^{(k)}\right] & \left[s_{19636,2}^{(k)} \cdots s_{19800,2}^{(k)}\right] & \cdots & \left[s_{19636,24}^{(k)} \cdots s_{19800,24}^{(k)}\right] \end{bmatrix}. \tag{2}$$

Thus, it can be written as

$$\mathbf{X}^{(k)} = \begin{bmatrix} x_{1,1}^{(k)} & x_{1,2}^{(k)} & \cdots & x_{1,R}^{(k)} \\ x_{2,1}^{(k)} & x_{2,2}^{(k)} & \cdots & x_{2,R}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{Q,1}^{(k)} & x_{Q,2}^{(k)} & \cdots & x_{Q,R}^{(k)} \end{bmatrix}, \tag{3}$$

where $Q = 19800/165 = 120$ is the total number of rows and $R = 24 \times 165 = 3960$ is the total number of columns. That is, the resulting matrix has size $\mathbf{X}_{Q,R}^{(k)} \in \mathbb{M}_{120\times 3960}(\mathbb{R})$.

Figure 4 shows the flowchart of the data acquisition and reshaping.

### 4.1.2. Data Split and Unfolding.

*4.1.2. Data Split and Unfolding.* To develop the model, the data are divided into three different sets, which are training, validation, and test set. These sets allow to train parameters, tune hyperparameters, and finally test the accuracy of the model, respectively.

For this study, the strategy adopted is to divide the data set into 65% for training, 20% for validation, and 15% for testing. Therefore, each matrix $\mathbf{X}^{(k)}$ is partitioned in three matrices: $\mathbf{X}_{\text{train}}^{(k)}$, $\mathbf{X}_{\text{val}}^{(k)}$, and $\mathbf{X}_{\text{test}}^{(k)}$, whose dimensions are $\mathbb{M}_{Q_{\text{train}}\times R}$, $\mathbb{M}_{Q_{\text{val}}\times R}$ and $\mathbb{M}_{Q_{\text{test}}\times R}$, respectively. Note that $Q_{\text{train}} = Q \times 0.65 = 78$, $Q_{\text{val}} = Q \times 0.20 = 24$, and $Q_{\text{test}} = Q \times 0.15 = 18$.

Finally, unfolding is performed [41] by a vertical concatenation of the different experiments, obtaining the matrices $\mathbf{X}_{\text{train}}$, $\mathbf{X}_{\text{val}}$, and $\mathbf{X}_{\text{test}}$, as illustrated in Figure 5. The dimensions of the new matrices are $\mathbb{M}_{P_{\text{train}}\times R}$, $\mathbb{M}_{P_{\text{val}}\times R}$, and $\mathbb{M}_{P_{\text{test}}\times R}$, respectively. Note that $P_{\text{train}} = Q_{\text{train}} \times K = 88920$, $P_{\text{val}} = Q_{\text{val}} \times K = 27360$, and $P_{\text{test}} = Q_{\text{test}} \times K = 20520$.

Figure 5 shows the flowchart for the data splitting and unfolding process. The $\mathbf{X}_{\text{train}}^{(k)}$, $\mathbf{X}_{\text{val}}^{(k)}$, and $\mathbf{X}_{\text{test}}^{(k)}$ matrices have the same structure of the $\mathbf{X}^{(k)}$ matrix plotted in Figure 4; however, for a better understanding of the unfolding, each experiment is assigned a unique color.

For more details, Table 2 shows the distribution of multivariate (MV) sequences for each data set according to the structural state. Furthermore, since MV sequences are also classified by the amplitude of the signal, a distribution based on the amplitude of the white noise is also considered, as summarized in Table 3.

*4.1.3. Feature Scaling: Data Normalization.* Data scaling is a key step applied during data preparation, the main goal of which is to change the numeric data into a common scale. Especially when the features have different ranges, this approach accelerates the network training [42]. In general, several techniques can be applied; however, in this work, $z$-score normalization is implemented [43]. This technique scales the values to a center around the mean, with a value of zero and a unit standard deviation. It is expected $z$-score normalization to enable more effective integration of multimodal sensor data in our model compared to techniques like min–max that do not normalize around a mean. Mapping the signals to a common relative scale around 0 rather than an absolute scale between 0 and 1 (as min–max does) helps prevent scale mismatches during integration that could have imbalanced influence of certain modalities over others. However, further study could systematically compare normalization techniques to confirm the most appropriate method for this application.

In this paper, the $z$-score normalization is employed, which is calculated as

$$\mathbf{x}_\mathbf{r}' = \frac{\mathbf{x}_\mathbf{r} - \mu_r}{\sigma_r}, \quad r = 1, \ldots, R, \tag{4}$$

where $\mu_r$ and $\sigma_r$ are the mean and standard deviation of the measurements in column $r$ including only training and validation datasets to compute them. Note that $\mathbf{x}_r$ is the feature vector at column $r$, and $\mathbf{x}_\mathbf{r}{}'$ is its normalized value. Obviously, normalization is applied to training, validation, and testing datasets.

*4.2. Transformers Based on Multivariate Data Series for the Detection and Localization of Damage.* First, this section details the fundamentals of transformers, from their beginnings to their current fields of application. Subsequently, the architecture and modifications made to the original transformer model for its adaptation using multivariate time data for sequence classification are explained.

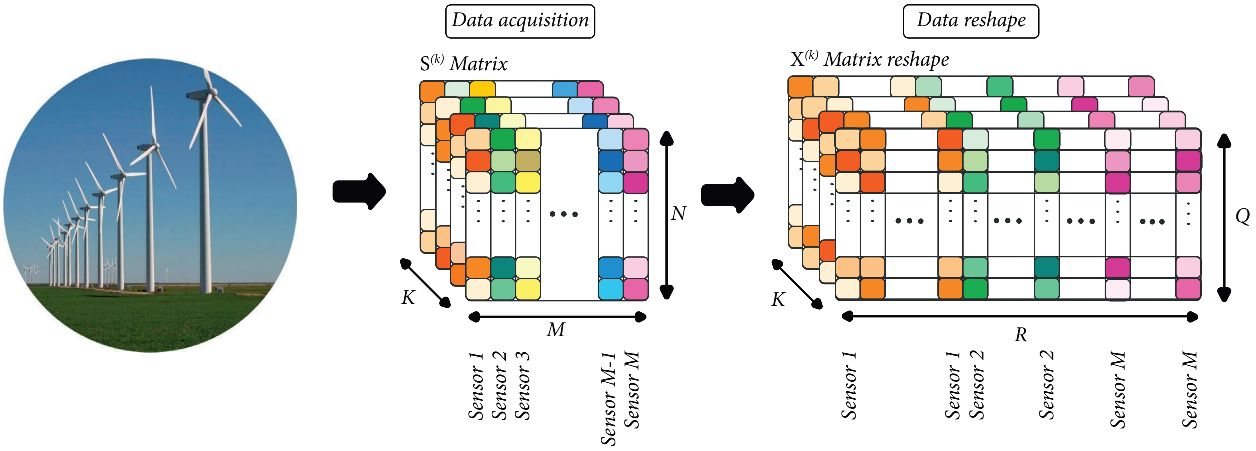


Figure 4: Data acquisition and reshaping flowchart.

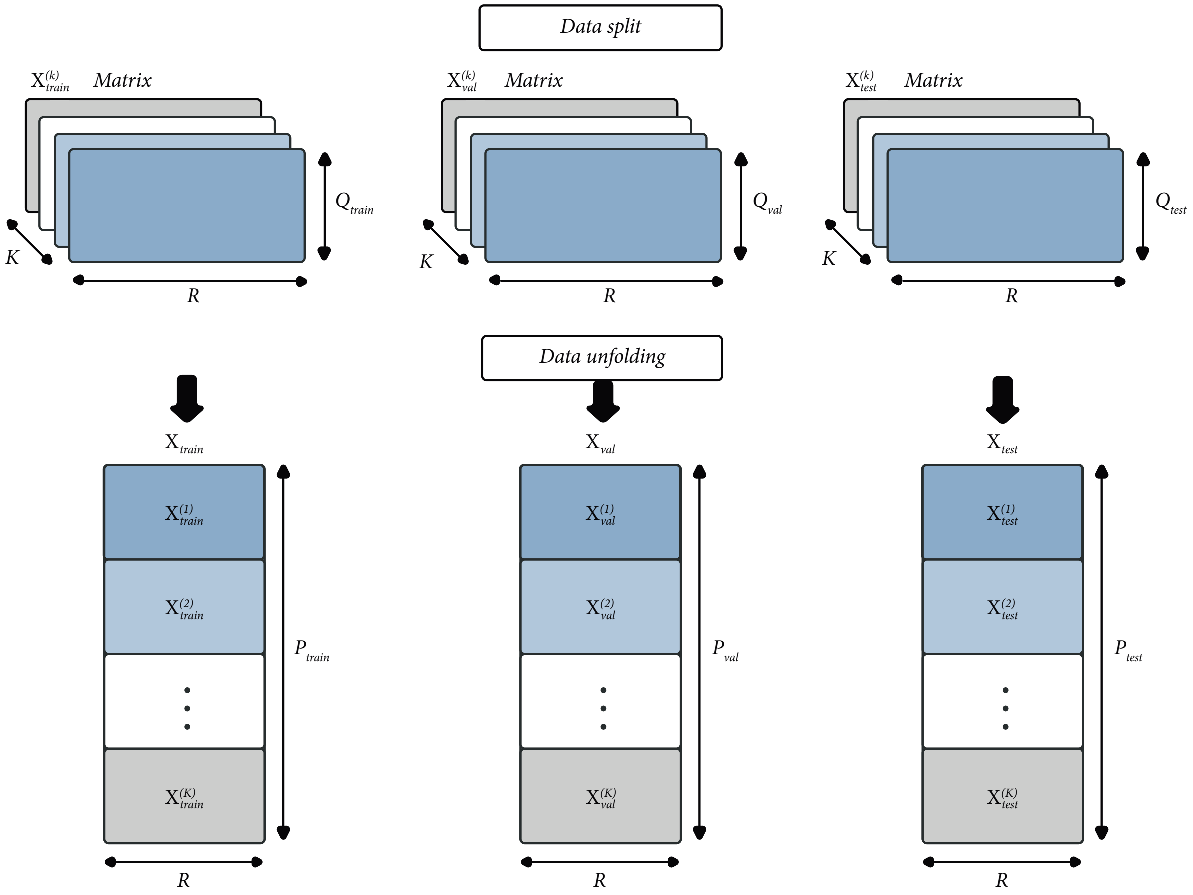


Figure 5: Data splitting and unfolding flowchart.

*4.2.1. Background.* The transformer models have demonstrated high accuracy and efficiency in NLP, becoming the reference architecture in this area. Moreover, due to the great capacity achieved in the modeling of dependencies and long-range interactions in sequential data, it has become a very attractive alternative for time-series modeling.

The original transformer model used in NLP works with sequences of text tokens; however, when working with time series, it is necessary to adapt the model and overcome the obstacles presented when working with this type of sequences. Multiple variants in the original architecture have been proposed. These variants have been successfully applied in several tasks using time series [44], such as forecasting [45, 46], classification [47, 48] and anomaly detection [32, 49].

In general, transformer models have proven to be effective in a wide variety of applications, so new applications and approaches for these models are being discovered as more advanced techniques are developed for their training and use.

Table 2: Distribution of MV sequences for training, validation, and testing based on the structural state.

| Structural state | Total of experiments | Training MV sequences ($P_{\text{train}}$) | Validation MV sequences ($P_{\text{val}}$) | Testing MV sequences ($P_{\text{test}}$) | Total of MV sequences |
|---|---|---|---|---|---|
| Crack level 1 | 80 | 6240 | 1920 | 1440 | 9600 |
| Crack level 2 | 80 | 6240 | 1920 | 1440 | 9600 |
| Crack level 3 | 80 | 6240 | 1920 | 1440 | 9600 |
| Crack level 4 | 80 | 6240 | 1920 | 1440 | 9600 |
| Bolt level 1 | 80 | 6240 | 1920 | 1440 | 9600 |
| Bolt level 2 | 80 | 6240 | 1920 | 1440 | 9600 |
| Bolt level 3 | 80 | 6240 | 1920 | 1440 | 9600 |
| Bolt level 4 | 80 | 6240 | 1920 | 1440 | 9600 |
| Replica level 1 | 80 | 6240 | 1920 | 1440 | 9600 |
| Replica level 2 | 80 | 6240 | 1920 | 1440 | 9600 |
| Replica level 3 | 80 | 6240 | 1920 | 1440 | 9600 |
| Replica level 4 | 80 | 6240 | 1920 | 1440 | 9600 |
| Healthy | 180 | 14040 | 4320 | 3240 | 21600 |
| Total | 1140 | 88920 | 27360 | 20520 | 136800 |

Table 3: Distribution of MV sequences for training, validation, and testing for each white noise amplitude.

| Dataset | Number of MV sequences | | | | |
|---|---|---|---|---|---|
| | 0.5 WN | 1 WN | 2 WN | 3 WN | Total |
| Training | 22230 | 22230 | 22230 | 22230 | 88920 |
| Validation | 6840 | 6840 | 6840 | 6840 | 27360 |
| Test | 5130 | 5130 | 5130 | 5130 | 20520 |
| Total | 34200 | 34200 | 34200 | 34200 | 136800 |

*4.2.2. Model Architecture.* Transformers are deep learning models composed of several encoding and decoding blocks that process data; all these blocks are identical to each other [30] and are characterized by a multihead attention mechanism, a position feedforward network, layer normalization modules, and residual connectors [50].

For this work, because a classification of the different structural states analyzed is required, it is not feasible to work directly with the original architecture of the transformer model, since this architecture works sequence to sequence. Therefore, based on [51], where a framework for multivariate time series regression and classification is presented that fits the type of data at hand, this proposed transformer model is used. This model, unlike the original architecture described in the original paper by [28], uses only encoding blocks and omits the decoding blocks from the architecture, in order to make the architecture compatible with multivariate time series classification problems.

Recall that each training sample is a multivariate time series of length $\omega$ (recall that $w = 165$ is used, see (2)) and $M$ different variables (sensors). Then, the input is linearly projected onto a $d$-dimensional vector space, where $\mathbf{d}$ is the dimension of the transformer model sequence element representations (also called *model dimension*). The linear projection of the feature vector is given by

$$\mathbf{U} = \mathbf{W_p}\left(\mathbf{X}'\right)^T + \mathbf{b_p}, \tag{5}$$

where $(\mathbf{X}')^T \in \mathbb{R}^{R\times w}$ is the transposed matrix of the $\mathbf{X}'$ and $\mathbf{W_p} \in \mathbb{R}^{d\times R}$ and $\mathbf{b_p} \in \mathbb{R}^d$ are the learning parameters.

Figure 6 describes the network architecture used in this work. The left side of the figure shows the different layers, including the transformer encoder, and the right side describes in more detail the feedforward network.

To provide the model with information about the order of the sequence, positional encodings are added to the input embedding, which consists of a linear block. The positional encodings have the same dimension $d$ in order to be summed with the linear projection. As proposed in the [51] framework, a learnable positional encoding is applied in this architecture as it demonstrated better performance, in contrast to the fixed sinusoidal encodings proposed in the original network [28], described as $\mathbf{W_{pos}} \in \mathbb{R}^{d\times w}$ which is added to the input vectors after the first linear layer $\mathbf{U} \in \mathbb{R}^{d\times w}$, where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_w]$, then $\mathbf{U}' = \mathbf{U} + \mathbf{W}_{\text{pos}}$.

The architecture is composed of several linear layers, dropout, encoding, and activation functions. Dropout is a regularization technique [52] that randomly deactivates a proportion of neurons during training to prevent overfitting. The dropout rate parameter controls the proportion of neurons that are randomly dropped out during training. In this research, the dropout rate is set to 0.1 after testing a range of values. This dropout rate of 0.1 allows the model to generalize well to new data without overfitting or losing too much representational power. Recall that, in general, the dropout rate parameter ranges from 0 to 1, where lower values like 0.1 mean fewer neurons are randomly dropped during training. In addition, the dimension of the linear layer $(d_{ff})$ is adjusted in different tests, as explained in Section 5.

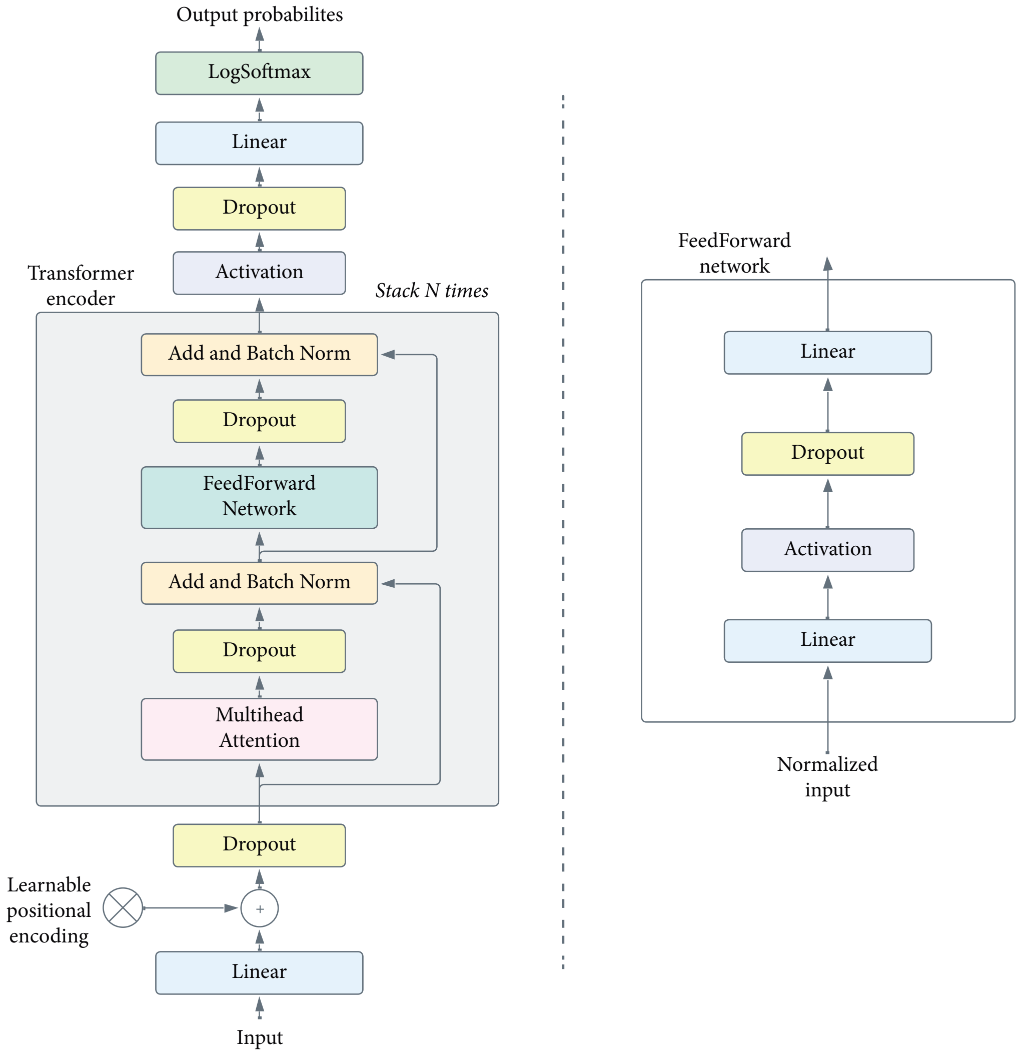


Figure 6: Transformers network architecture.

As stated in [53], activation functions can have a significant impact in reducing the topological complexity of the input data and thus improve model performance. According to [54], the most commonly used activation functions in transformer model development are the Rectified Linear Unit (ReLU) and Gaussian Error Linear Unit (GeLU) functions, which are graphed in Figure 7. In general, the ReLU function is a linear mathematical function often used as a default option in hidden layers of neural networks due to its simplicity and effectiveness [56]; instead, the GeLU function is a nonlinear function that approximates the ReLU function in the positive region and is smooth in the negative region, weighting its inputs by their value rather than by their sign when setting a threshold [57]. Both functions are often widely used because they allow avoiding the existence of saturation, as well as allowing a computationally less expensive implementation than exponential, sigmoidal, and other functions. Currently, due to its fluidity and ability to model more complex relationships in the data, the GeLU function is usually recommended for output layers and very deep neural networks [58]. In this paper, different configurations are designed using these two functions, for further comparison and selection of the best model.

The most important block in the architecture of this network is the transformer encoder, which is detailed below.

*(1) Transformer Encoder*. The transformer encoder is composed of a stack of identical blocks that are stacked consecutively. Where each block is composed of two sublayers, the first is a multihead attention mechanism, and the second is a fully positionally connected feedforward network. The sublayers are described below:

Multihead attention: An attention function is described as mapping a query and a set of key-value pairs to an output vector, multihead attention consists of a linear projection of a weight matrix $W^O$ and a concatenated output of attention heads head$_i$ where $i = 1, \ldots, h$

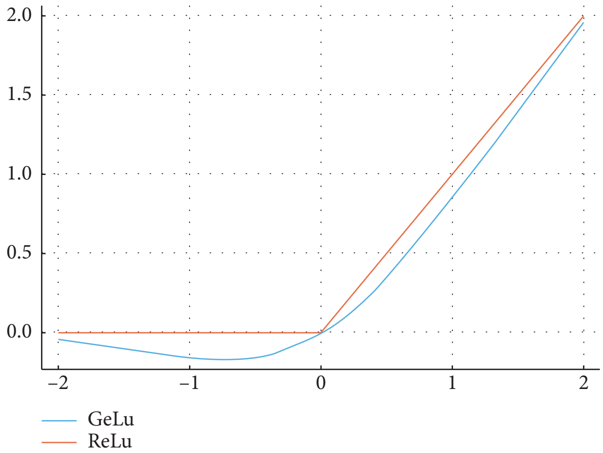


Figure 7: Comparison of the ReLU and GeLU activation functions [55].

which is the linear projection between queries, keys, and values **h** times (number of attention heads), each head performs an attention function described as [28]

$$\begin{gathered}\text{Multihead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O,\\ \text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right),\end{gathered} \tag{6}$$

where $Q$, $K$, and $V$ are the matrix of queries, keys, and values, respectively. Each head$_i$ is a result of a single attention function characterized by its own learned projection matrices $W^O \in \mathbb{R}^{hd_v \times d_k}$, $W_i^Q \in \mathbb{R}^{d\times d_k}$, $W_i^K \in \mathbb{R}^{d\times d_k}$, and $W_i^V \in \mathbb{R}^{d\times d_v}$, in addition $d_k = d_v = d/h$. The input of the multihead attention layer is $\mathbf{U}'$ as keys, queries, and values which is the result of the sum of the input embedding (linear layer) and the positional encoding.

Feedforward network: The right side of Figure 6 describes the fully connected feedforward network applied, and this block consists of two linear transformations, either ReLU or GeLU activation functions, and a regularization dropout.

Around each of the two sublayers multihead attention and the feedforward network, a residual connection is considered followed by a batch normalization ($\epsilon = 1\times 10^{-5}$), it is important to highlight that batch normalization is considered instead of layer normalization, as it mitigates the effect of outliers in time series [51] and vanishing gradient [59]. To reduce the complexity of the residual connections, all sublayers as embedded layers are linearly projected onto a vector space $(d)$-dimensional.

On the one hand, the residual connections in the encoder follow the standard formulation:

$$y = F(x) + x, \tag{7}$$

where $F(x)$ represents the transformer sublayers—multihead attention and feedforward network—$x$ is the input and $y$ is the output fed to the next layer. This skips connections and directly adds the sublayer outputs to the unmodified inputs before applying the batch norm. Some key benefits are easing gradient flow during training and enabling very deep networks. On the other hand, batch norm addresses internal covariate shift and is applied after each sublayer. The formulation is

$$\text{BN}(x_i) = \gamma\left(\frac{x_i - \mu_\beta}{\sigma_\beta}\right) + \beta, \tag{8}$$

where $x_i$ is the input feature map, $\mu_\beta$ and $\sigma_\beta$ are the mean and standard deviation computed within each transformer batch $\beta$ across the channel for normalization, and $\gamma$ and $\beta$ are learnable scale/shift parameters. So in summary, residuals connections propagate signals directly while batch norm stabilizes layer-wise dynamics.

Additionally, since a classification task is necessary for the detection of the structural state of the offshore WT, the last part includes a logarithmic softmax function that is applied to the input to compute the distribution between classes. The cross-entropy loss is used to calculate the sampling error, in contrast to the categorical field-truth labels.

The learning rate is a key hyperparameter in the training of machine learning models because very high learning rates can cause the gradient descent to be very fast and skip the real local minimum to optimize time, while very small rates can cause a very slow training or even never converge because it does not find the local minimum [60]. That is why, for this paper, it is decided to use the RAdam optimizer algorithm, which has an adaptive learning rate which is automatically adjusted during the training process, achieving greater stability in the training and reaching a fast but safe convergence. Additionally, the RAdam optimizer allows rectifying the variance of the adaptive learning rate that was present with Adam's optimizer [61], especially in the initial stage of the model training, obtaining a consistent variation that allows avoiding divergence problems.

For the selection of the batch size, two important points are considered, the first related to accessible computational resources and the second to guarantee variability in each batch, ensuring that at least one sample of each structural state is taken in each batch.

Finally, for the selection of the other hyperparameters $d_{ff}$, $d$, $h$, and $e$, different values are studied, testing with several configurations in the model architecture, as detailed in Section 5, where it is shown that there is no significant variation in the model accuracy, therefore the analysis and selection of the best architecture is performed according to the metrics and the computing time.

A summary of all the hyperparameters used in the model is described in Table 4.

## 5. Results

This section presents the results obtained with the multivariate time series based transformer model for the detection and localization of different types of structural damage in the jacket of an offshore WT. It should be noted that, as detailed in Table 1, that in total 13 different structural states are analyzed.

A laptop computer with the macOS Monterey operating system, with an Apple M1 chip and 16 GB of RAM, was used to train the model. No GPU was used during this process.

To select the best hyperparameters, 8 tests are carried out by training the model with different architecture configurations, as summarized in Table 5. The modified hyperparameters are the activation function, feedforward dimension ($d_{ff}$), model dimension ($d$), number of heads ($h$), and number of transformer encoder blocks ($e$). The other unmentioned hyperparameters are kept as detailed in Table 4.

The activation function is an important component of the architecture of neural network models. In the multivariate time series-based transformer model, two different activation functions are evaluated. The comparative results in Table 5 show that the choice of activation function does not have a major impact on model accuracy across the different configurations. The ReLU and GeLU versions of each architecture achieve highly similar accuracy, precision, recall, and $F1$-scores. However, the GeLU activation confers a noticeable increase in training time compared to the equivalent ReLU model, requiring 13–19 minutes on average instead of 6–17 minutes. This is consistent with the higher computational complexity of the GeLU function. Based on these validation and test results, the ReLU activation function is selected for the final transformer model configuration.

In addition to the activation function, the other aforementioned hyperparameters were also varied. Prior research on developing transformer models for time series data has commonly used between 4 and 8 attention heads [31]. Using more heads increases model capacity but also increases computational complexity. A moderate number balances representational power and efficiency. The projection dimensions $d_k$, $d_v$ are set equal to the model dimension $d$ divided by the number of heads $h$. This ensures each head receives a proportional chunk of the feature dimensionality for attention. Configurations with 4 and 8 heads are evaluated on the data, with minimal gains from 8 heads but substantially longer training times. Based on these experiments, 4 heads with $d_k = d_v = d/4 = 32/4 = 8$ is deemed optimal to balance accuracy and complexity.

Table 4: Hyperparameters used to train the transformer model.

| Parameter | Value |
|---|---|
| Activation function | GeLU/ReLU |
| Learning rate | 0.001 |
| Pos. encoding | Learnable |
| Dropout | 0.1 |
| Batch size | 64 |
| Batch norm | $\epsilon = 1 \times 10^{-5}$ |
| Feedforward dimension | $d_{ff} = 16$ |
| Model dimension | $d = 32$ |
| Number of heads | $h = 4$ |
| Number of transformer encoder blocks | $e = 2$ |

To avoid overfitting the model, the early stopping technique is used to store the model with the best performance, just when the evaluation metric on the validation set stops improving during training. In this case, the evaluation metric used is accuracy. The model was trained for a total of 35 epochs, but the early stopping technique is used to store the model at the best epoch, which was epoch number 14. At that point, the accuracy of the validation set stopped improving and began to decrease, indicating that the model began to overfit.

In all tests performed with different configurations, the model accuracy was above 99.9%, demonstrating that the model is highly effective in detecting and locating the structural states analyzed, regardless of the hyperparameters used. To obtain a computationally simpler model, with a shorter training time and high accuracy, configuration 1 is chosen, detailed in the first row of Table 5.

To ensure that the model is not overfitting, a visual and analytical analysis of the loss curves during training is performed. The chosen model does not present overfitting because, as shown in Figure 8, the loss values of the training and validation set are similar in the best epoch, obtaining a value of 0.0009 and 0.0016. Likewise, the model accuracy data are presented in Figure 9, obtaining values of 0.9997 and 0.9994 for the training and validation set, respectively, in the best epoch. For both plots, a 35 epoch training was simulated, resulting in the best epoch number 14 because it is the epoch that presents the best performance in the loss, precision average, recall average, $F1$-score average, and accuracy metrics with a value of 0.0016, 0.9993, 0.9993, 0.9993, and 0.9994, respectively, at a lower training time in the validation dataset; this can be evidenced in Tables 6 and 7, which details the performance of the different epochs as a summary, in the training and validation datasets, respectively. Table 8 details the results obtained in the test dataset, when testing the model in each of the epochs once it has been trained, in order to visualize how the accuracy varies in each model obtained from each training epoch.

In Figure 10, the confusion matrix is presented summarizing the performance of the classification model in the test set for 13 different structural states. Each column of the matrix represents the true class of the samples, and each row represents the class predicted by the model. On the main diagonal of the confusion matrix, the number of true predictions can be observed for each of the 13 structural states; that is, the number of samples that were correctly classified. On the other hand, outside the main diagonal, few

Table 5: Results obtained by testing the methodology with different architecture configurations of the transformer model.

| Configuration number | Activation function | $d_{ff}$ | $d$ | $h$ | $e$ | Inference time (ms) | Avg training time (min) | Best epoch | Validation accuracy | Training accuracy | Test accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ReLU | 16 | 32 | 4 | 2 | 2.87 | 6.13 | 14 | 0.9994 | 0.9997 | 0.9996 |
| 2 | ReLU | 16 | 16 | 8 | 3 | 1.96 | 12.32 | 22 | 0.9996 | 0.9998 | 0.9998 |
| 3 | ReLU | 128 | 32 | 8 | 3 | 2.24 | 14.24 | 33 | 0.9997 | 0.9999 | 0.9994 |
| 4 | ReLU | 256 | 64 | 8 | 3 | 2.72 | 17.54 | 22 | 0.9994 | 0.9944 | 0.9996 |
| 5 | GeLU | 16 | 32 | 4 | 2 | 2.82 | 6.20 | 20 | 0.9995 | 0.9998 | 0.9993 |
| 6 | GeLU | 16 | 16 | 8 | 3 | 5.41 | 13.66 | 17 | 0.9996 | 0.9998 | 0.9994 |
| 7 | GeLU | 128 | 32 | 8 | 3 | 5.99 | 16.11 | 23 | 0.9995 | 0.9990 | 0.9994 |
| 8 | GeLU | 256 | 64 | 8 | 3 | 8.81 | 19.98 | 25 | 0.9995 | 0.9926 | 0.9995 |

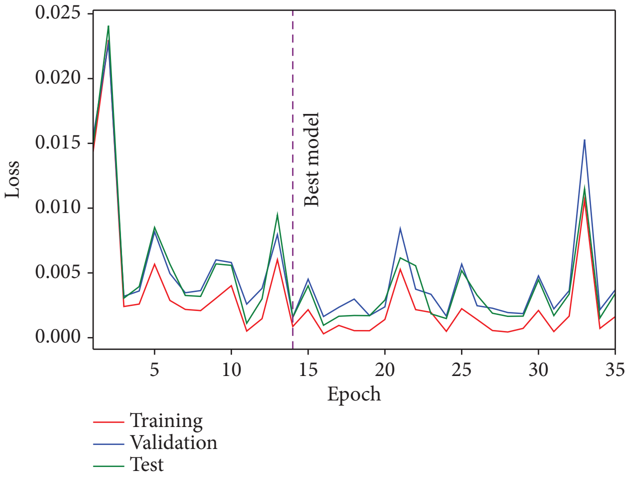


Figure 8: Training, validation, and testing loss curves.

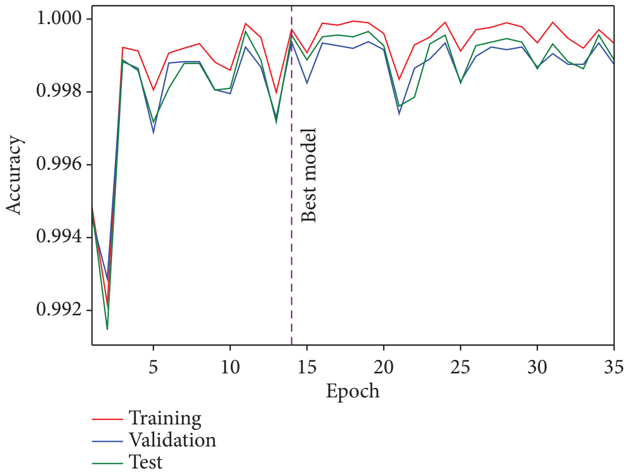


Figure 9: Training, validation, and testing accuracy curves.

misclassifications can be observed. In total, only 5 samples from the class "Crack level 1" were misclassified as "Crack level 2," 3 samples from the class "Bolt level 2" were misclassified as "Bolt level 1," and 1 sample from the class "Bolt level 1" was misclassified as "Bolt level 2." In conclusion, the model does not have difficulty distinguishing between the different structural states, presenting very few errors in the detection of the 13 types, which were only at the location level, but not the type of damage (crack or missing bolt). The final accuracy achieved with the test data is 99.97%.

Table 6: Precision, recall, $F1$-score, and accuracy metrics after 35 epochs on training data set.

| Epoch | Precision average | Recall average | $F1$-score average | Accuracy | Loss |
|---|---|---|---|---|---|
| 1 | 0.9947 | 0.9943 | 0.9943 | 0.9948 | 0.0144 |
| 2 | 0.9921 | 0.9914 | 0.9914 | 0.9922 | 0.0230 |
| 3 | 0.9992 | 0.9991 | 0.9992 | 0.9992 | 0.0024 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 13 | 0.9978 | 0.9978 | 0.9978 | 0.9980 | 0.0060 |
| 14* | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.0008 |
| 15 | 0.9990 | 0.9990 | 0.9990 | 0.9991 | 0.0022 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 33 | 0.9991 | 0.9991 | 0.9991 | 0.9992 | 0.0112 |
| 34 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.0005 |
| 35 | 0.9993 | 0.9993 | 0.9993 | 0.9993 | 0.0014 |

The asterisk identifies the best model.

Table 7: Precision, recall, $F1$-score, and accuracy metrics after 35 epochs on validation data set.

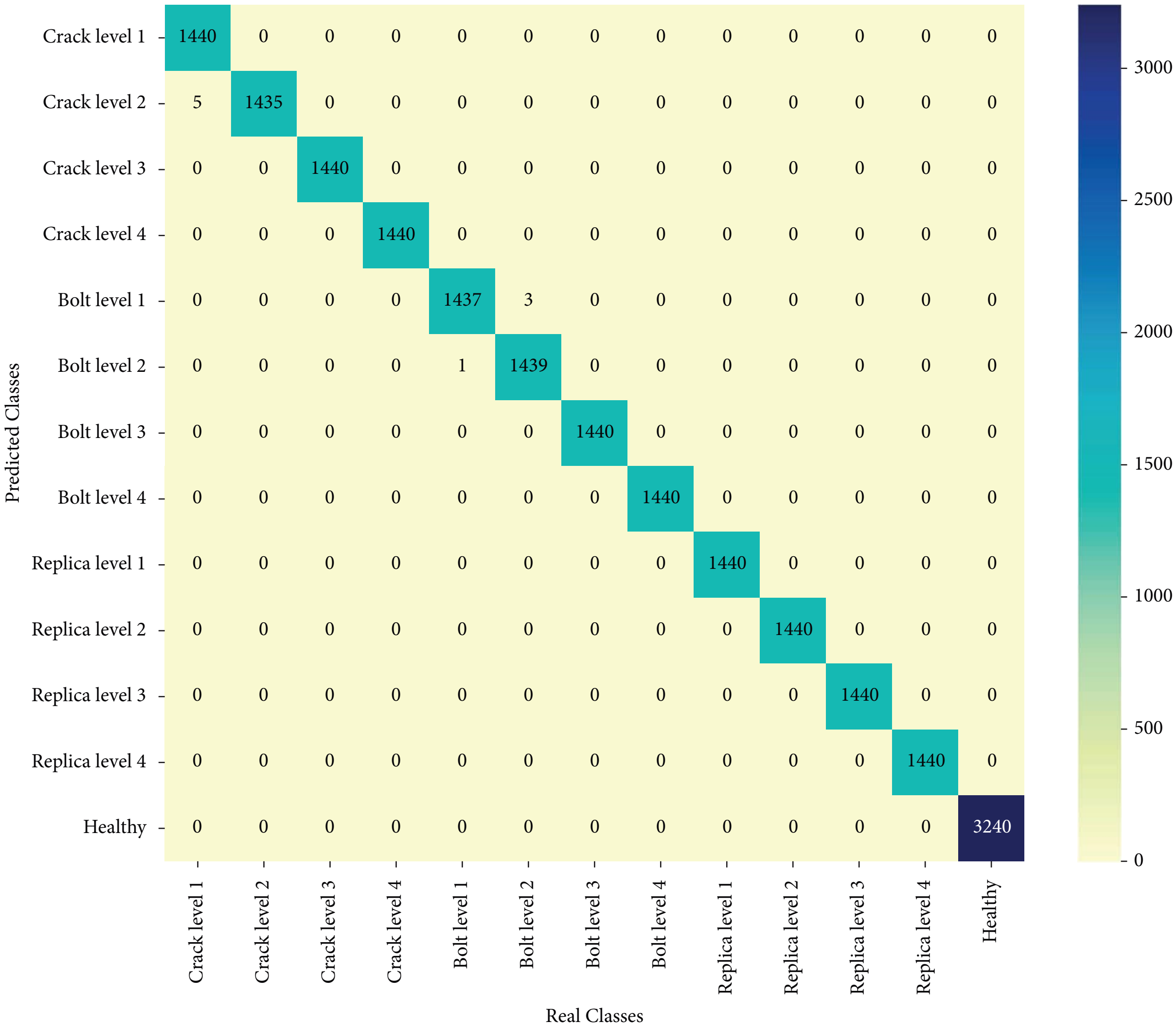
| Epoch | Precision average | Recall average | $F1$-score average | Accuracy | Loss |
|---|---|---|---|---|---|
| 1 | 0.9943 | 0.9940 | 0.9940 | 0.9945 | 0.0155 |
| 2 | 0.9928 | 0.9922 | 0.9922 | 0.9929 | 0.0227 |
| 3 | 0.9987 | 0.9987 | 0.9987 | 0.9988 | 0.0032 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 13 | 0.9971 | 0.9970 | 0.9970 | 0.9973 | 0.0080 |
| 14* | 0.9993 | 0.9993 | 0.9993 | 0.9994 | 0.0016 |
| 15 | 0.9981 | 0.9981 | 0.9981 | 0.9982 | 0.0045 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 33 | 0.9986 | 0.9986 | 0.9986 | 0.9988 | 0.0153 |
| 34 | 0.9993 | 0.9993 | 0.9993 | 0.9993 | 0.0024 |
| 35 | 0.9987 | 0.9986 | 0.9986 | 0.9988 | 0.0037 |

The asterisk identifies the best model.

Table 8: Precision, recall, $F1$-score, and accuracy metrics after 35 epochs on test data set.

| Epoch | Precision average | Recall average | $F1$-score average | Accuracy | Loss |
|---|---|---|---|---|---|
| 1 | 0.9946 | 0.9943 | 0.9943 | 0.9948 | 0.0148 |
| 2 | 0.9914 | 0.9907 | 0.9906 | 0.9915 | 0.0241 |
| 3 | 0.9988 | 0.9988 | 0.9988 | 0.9989 | 0.0031 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 13 | 0.9970 | 0.9969 | 0.9969 | 0.9972 | 0.0095 |
| 14* | 0.9995 | 0.9995 | 0.9995 | 0.9996 | 0.0016 |
| 15 | 0.9988 | 0.9988 | 0.9988 | 0.9989 | 0.0040 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 33 | 0.9985 | 0.9985 | 0.9985 | 0.9986 | 0.0114 |
| 34 | 0.9995 | 0.9995 | 0.9995 | 0.9996 | 0.0015 |
| 35 | 0.9988 | 0.9988 | 0.9988 | 0.9989 | 0.0035 |

The asterisk identifies the best model.



| Predicted Classes \ Real Classes | Crack level 1 | Crack level 2 | Crack level 3 | Crack level 4 | Bolt level 1 | Bolt level 2 | Bolt level 3 | Bolt level 4 | Replica level 1 | Replica level 2 | Replica level 3 | Replica level 4 | Healthy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Crack level 1 | 1440 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Crack level 2 | 5 | 1435 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Crack level 3 | 0 | 0 | 1440 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Crack level 4 | 0 | 0 | 0 | 1440 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bolt level 1 | 0 | 0 | 0 | 0 | 1437 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bolt level 2 | 0 | 0 | 0 | 0 | 1 | 1439 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bolt level 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1440 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bolt level 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1440 | 0 | 0 | 0 | 0 | 0 |
| Replica level 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1440 | 0 | 0 | 0 | 0 |
| Replica level 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1440 | 0 | 0 | 0 |
| Replica level 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1440 | 0 | 0 |
| Replica level 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1440 | 0 |
| Healthy | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3240 |

Figure 10: Confusion matrix on the testing data set using the best model after epoch number 14.

## 6. Conclusions

In this work, a model for the detection and localization of different types of structural states located at four different levels of the jacket structure of an offshore WT is proposed. The methodology presented is based on a vibration-response only analysis for the development of the transformer model based on multivariate time series. Data were collected by eight sensors (triaxial accelerometers) in 1140 different experiments performed at different frequencies, so the model is adaptable for all regions of operation of WTs.

Since the original transformers model is based on a sequence-to-sequence problem, for this work, the architecture is successfully modified to adapt to the multivariate time data classification problem. Achieving almost perfect classification of the 13 different structural states, with a model accuracy of 99.96%, a precision average of 99.95%, a recall average of 99.95%, *F*1-score average of 99.95%, and loss of 0.0016 for the test set is obtained. The results obtained suggest that the model is highly effective, allowing early detection and localization of damage and allowing the wind farm operator to take corrective measures before damage becomes major or even catastrophic. Thus, the results successfully demonstrate the transformer model's efficacy for detecting and localizing damage, achieving the research objective.

Finally, the key advantage of the transformer neural network approach from an applied perspective is its highly parallelizable computational architecture. Using self-attention instead of recurrent processing, the model can rapidly analyze all points in a multivariate input sequence simultaneously. This allows the transformer-based model to evaluate lengthy 60-second windows of vibration data for accurate damage detection in only a few milliseconds of inference time per sample. The ability to perform rapid detection on high-rate vibration streams with minimal latency means that the transformer methodology can be readily integrated into real-time offshore monitoring systems for continuous structural health assessment.

## Data Availability

The data that support the findings of this study are available on request from the corresponding author.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is partially funded by the Spanish Agencia Estatal de Investigación (AEI)—Ministerio de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo Regional (FEDER) through the research projects PID2021-122132OB-C21 and TED2021-129512B-I00; and by the Generalitat de Catalunya through the research project 2021-SGR-01044. Open Access funding was enabled and organized by CRUE-CSUC Gold.

## References

[1] H. E. Murdock, D. Gibb, T. André et al., *Renewables 2021-global Status Report*, United Nations Environment Programme, Nairobi, Kenya, 2021.

[2] Ren, "Renewables 2022: global status report," *Renewable and Sustainable Energy Reviews*, United Nations Environment Programme, Nairobi, Kenya, 2022.

[3] D. Y. Leung and Y. Yang, "Wind energy development and its environmental impact: a review," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 1, pp. 1031–1039, 2012.

[4] P. Bojek, *Wind Electricity*, IEA, Paris, France, 2022.

[5] Council Gwe, *Global Wind Report 2022*, Global Wind Energy Council, Brussels, Belgium, 2022.

[6] X. Sun, D. Huang, and G. Wu, "The current state of offshore wind energy technology development," *Energy*, vol. 41, no. 1, pp. 298–312, 2012.

[7] G. W. E. C. Gwec, *Global Wind Report 2022*, GWEC, Brussels, Belgium, 2022.

[8] W. J. Lai, C. Y. Lin, C. C. Huang, and R. M. Lee, "Dynamic analysis of Jacket Substructure for offshore wind turbine generators under extreme environmental conditions," *Applied Sciences*, vol. 6, no. 10, p. 307, 2016.

[9] M. Arshad and B. C. O'Kelly, "Offshore wind-turbine structures: a review," *Proceedings of the Institution of Civil Engineers-Energy*, vol. 166, no. 4, pp. 139–152, 2013.

[10] M. Shafiee, "Maintenance logistics organization for offshore wind energy: current progress and future perspectives," *Renewable Energy*, vol. 77, pp. 182–193, 2015.

[11] G. Rinaldi, P. R. Thies, and L. Johanning, "Current status and future trends in the operation and maintenance of offshore wind turbines: a review," *Energies*, vol. 14, no. 9, p. 2484, 2021.

[12] J. Seo, J. W. Hu, and J. Lee, "Summary review of structural health monitoring applications for highway bridges," *Journal of Performance of Constructed Facilities*, vol. 30, no. 4, Article ID 04015072, 2016.

[13] P. Palma and R. Steiger, "Structural health monitoring of timber structures–Review of available methods and case studies," *Construction and Building Materials*, vol. 248, Article ID 118528, 2020.

[14] G. C. Kahandawa, J. Epaarachchi, H. Wang, and K. Lau, "Use of FBG sensors for SHM in aerospace structures," *Photonic Sensors*, vol. 2, no. 3, pp. 203–214, 2012.

[15] F. G. Yuan, *Structural Health Monitoring (SHM) in Aerospace Structures*, Woodhead Publishing, Sawston, UK, 2016.

[16] T. J. Eason, L. J. Bond, and M. G. Lozev, *Structural Health Monitoring of Localized Internal Corrosion in High Temperature Piping for Oil Industry*, American Institute of Physics, Maryland, MD, USA, 2015.

[17] C. P. Fritzen, P. Kraemer, and M. Klinkov, *An Integrated SHM Approach for Offshore Wind Energy Plants*, Springer, Singapore, 2011.

[18] R. Yan and S. Dunnett, "Improving the strategy of maintaining offshore wind turbines through Petri net modelling," *Applied Sciences*, vol. 11, no. 2, p. 574, 2021.

[19] D. Goyal and B. Pabla, "The vibration monitoring methods and signal processing techniques for structural health monitoring: a review," *Archives of Computational Methods in Engineering*, vol. 23, no. 4, pp. 585–594, 2016.

[20] Y. Yang, Y. Zhang, and X. Tan, "Review on vibration-based structural health monitoring techniques and technical codes," *Symmetry*, vol. 13, no. 11, p. 1998, 2021.

[21] X. Ye, T. Jin, and C. Yun, "A review on deep learning-based structural health monitoring of civil infrastructures," *Smart Structures and Systems*, vol. 24, no. 5, pp. 567–585, 2019.

[22] J. Baquerizo, C. Tutivén, B. Puruncajas, Y. Vidal, and J. Sampietro, "Siamese neural networks for damage detection and diagnosis of jacket-type offshore wind turbine platforms," *Mathematics*, vol. 10, no. 7, p. 1131, 2022.

[23] M. D. C. Feijóo, Y. Zambrano, Y. Vidal, and C. Tutivén, "Unsupervised damage detection for offshore jacket wind turbine foundations based on an autoencoder neural network," *Sensors*, vol. 21, no. 10, p. 3333, 2021.

[24] G. Oliveira, F. Magalhães, Á Cunha, and E. Caetano, "Vibration-based damage detection in a wind turbine using 1 year of data," *Structural Control and Health Monitoring*, vol. 25, no. 11, p. e2238, 2018.

[25] 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.

[26] 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, Article ID 0309524X221123968, 2023.

[27] F. Castellani, L. Garibaldi, A. P. Daga, D. Astolfi, and F. Natili, "Diagnosis of faulty wind turbine bearings using tower vibration measurements," *Energies*, vol. 13, no. 6, p. 1474, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar et al., *Attention Is All You Need*, CoRR, Leawood, KS, USA, 2017.

[29] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: a survey," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–41, 2022.

[30] T. Lin, Y. Wang, X. Liu, and X. Qiu, *A Survey of Transformers*, AI Open, San Francisco, CA, USA, 2022.

[31] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Association for Computing Machinery*, pp. 2114–2124, ACM, New York, NY, USA, 2021.

[32] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: deep transformer networks for anomaly detection in multivariate time series data," 2022, https://arxiv.org/abs/2201.07284.

[33] İY. Potter, G. Zerveas, C. Eickhoff, and D. Duncan, "Unsupervised multivariate time-series transformers for seizure identification on EEG," 2023, https://arxiv.org/abs/2301.03470.

[34] E. Zugasti Uriguen, *Design and validation of a methodology for wind energy structures health monitoring*, Ph.D. thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2014.

[35] M. Pastor, M. Binda, and T. Harčarik, "Modal assurance criterion," *Procedia Engineering*, vol. 48, pp. 543–548, 2012.

[36] B. Puruncajas, Y. Vidal, and C. Tutivén, "Vibration-response-only structural health monitoring for offshore wind turbine jacket foundations via convolutional neural networks," *Sensors*, vol. 20, no. 12, p. 3429, 2020.

[37] V. Pettas, M. Kretschmer, A. Clifton, and P. W. Cheng, "On the effects of inter-farm interactions at the offshore wind farm Alpha Ventus," *Wind Energy Science*, vol. 6, no. 6, pp. 1455–1472, 2021.

[38] W. Li, C. Hancock, Y. Yang, J. Wang, and X. Meng, "Dynamic deformation monitoring of an offshore platform structure with accelerometers," *Journal of Civil Structural Health Monitoring*, vol. 12, no. 2, pp. 275–287, 2022.

[39] J. Luengo, D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, *Big Data Preprocessing*, Springer, Cham, Switzerland, 2020.

[40] Explorium, "Feature Engineering- the ultimate guide," 2019, https://www.explorium.ai/wiki/feature-engineering/.

[41] J. A. Westerhuis, T. Kourti, and J. F. MacGregor, "Comparing alternative approaches for multivariate statistical analysis of batch process data," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 397–413, 1999.

[42] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, 2021.

[43] C. Andrade, "Z scores, standard scores, and composite test scores explained," *Indian Journal of Psychological Medicine*, vol. 43, no. 6, pp. 555–557, 2021.

[44] Q. Wen, T. Zhou, C. Zhang et al., "Transformers in time series: a survey," 2022, https://arxiv.org/abs/2202.07125.

[45] S. Wu, X. Xiao, Q. Ding, P. Zhao, Y. Wei, and J. Huang, "Adversarial sparse transformer for time series forecasting," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17105–17115, 2020.

[46] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Etsformer: exponential smoothing transformers for time-series forecasting," 2022, https://arxiv.org/abs/2202.01381.

[47] Y. Yuan, L. Lin, Q. Liu, R. Hang, and Z. G. Zhou, "SITS-Former: a pre-trained spatio-spectral-temporal representation model for Sentinel-2 time series classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, Article ID 102651, 2022.

[48] M. Liu, S. Ren, S. Ma et al., "Gated transformer networks for multivariate time series classification," 2021, https://arxiv.org/abs/2103.14438.

[49] Y. Li, X. Peng, J. Zhang, Z. Li, and M. Wen, "DCT-GAN: dilated convolutional transformer-based gan for time series anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3632–3644, 2023.

[50] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, https://arxiv.org/abs/1607.06450.

[51] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, *A Transformer-Based Framework for Multivariate Time Series Representation Learning*, CoRR, Leawood, KS, USA, 2020.

[52] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, https://arxiv.org/abs/1207.0580.

[53] W. Finnoff, F. Hergert, and H. G. Zimmermann, "Improving model selection by nonconvergent methods," *Neural Networks*, vol. 6, no. 6, pp. 771–783, 1993.

[54] N. Shazeer, "Glu variants improve transformer," 2020, https://arxiv.org/abs/2002.05202.

[55] M. Ureña-Pliego, R. Martínez-Marín, B. González-Rodrigo, and M. Marchamalo-Sacristán, "Automatic building height estimation: machine learning models for urban image analysis," *Applied Sciences*, vol. 13, no. 8, p. 5037, 2023.

[56] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *Proceedings of the 2018 Chinese Control And Decision Conference (CCDC)*, pp. 1836–1841, Shenyang, China, June 2018.

[57] M. Lee, "GELU activation function in deep learning: a comprehensive mathematical analysis and performance," 2023, https://arxiv.org/abs/2305.12073.

[58] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2016, https://arxiv.org/abs/1606.08415.

[59] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," 2016, https://arxiv.org/abs/1603.09025.

[60] S. Ray, "A quick review of machine learning algorithms," in *Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-ITCon)*, pp. 35–39, Faridabad, India, February 2019.

[61] L. Liu, H. Jiang, P. He et al., "On the variance of the adaptive learning rate and beyond," 2019, https://arxiv.org/abs/1908.03265.