Wiley
Structural Control and Health Monitoring
Volume 2024, Article ID 9997872, 26 pages
https://doi.org/10.1155/stc/9997872

WILEY

Research Article

Generative Adversarial Networks for Improved Model Training in the Context of the Digital Twin

María Megía,[1,2] **Francisco Javier Melero**,[1,3] **Manuel Chiachío**,[1,2] **and Juan Chiachío**[1,2]

[1]*Department of Technological Applications, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, Spain*
[2]*Department of Structural Mechanics & Hydraulics Engineering, University of Granada, Granada, Spain*
[3]*Department of Software Engineering, University of Granada, Granada, Spain*

Correspondence should be addressed to María Megía; mmegia@ugr.es

Received 16 October 2023; Revised 18 June 2024; Accepted 15 November 2024

Academic Editor: Ka-Veng Yuen

Digital twins (DTs) have revolutionised digitalisation practices across various domains, including the Architecture, Engineering, Construction and Operations (AECO) sector. However, DTs often face challenges related to data scarcity, especially in AECO, where tests are costly and difficult to scale. Historical data in this domain are often limited, unstructured and lack interoperability standards. Data scarcity directly affects the accuracy and reliability of the DT models and their decision-making capabilities. To address these challenges, classical methods are used to produce synthetic data based on predefined statistical distributions, which are barely scalable to unpredictable scenarios and prone to overfitting. Alternately, this work presents a novel comprehensive approach that covers every aspect from synthetic data generation to training and testing of these data on the system's models. This strategy not only delivers high-quality data that meets the model's requirements in terms of diversity, complexity and class balance, but also provides the diagnostic and prognostic capabilities of the DT of the system through its trained models. State-of-the-art techniques including generative adversarial networks (GANs), specifically Wasserstein generative adversarial networks with gradient penalty (WGAN-GP), and convolutional neural networks (CNNs) are employed in this novel pervasive approach, participating in the same architecture for generative, diagnostic and prognostic purposes. GANs enable data augmentation and reconstruction, while CNNs excel in spatial pattern recognition tasks. The proposed framework is demonstrated through an experimental case study on damage diagnostics and prognostics of a laboratory-scale metallic tower, where synthetic datasets are generated to supplement limited health monitoring data. The results showcase the effectiveness of the generated data for damage detection, prognostics and operational decision-making within the DT context. The presented method contributes to overcoming data scarcity challenges and improving the accuracy of DT models in the AECO sector. The article concludes with discussions on the application of the results and their implications for decision-making within the DT framework.

Keywords: convolutional neural network (CNN); digital twin (DT); generative adversarial networks (GANs); synthetic data; Wasserstein generative adversarial networks with gradient penalty (WGAN-GP)

1. Introduction

Digital twins (DTs) emerged from a NASA conceptualisation [1] as comprehensive simulations of real-world systems, using physical models, sensor data and historical information to mirror the behaviour of their physical counterparts. The continuous exchange of information between the real and the virtual twin, together with the real-time model updating, facilitates informed decision-making [2].

The DT framework has disrupted digitalisation practices in various domains, including the Architecture, Engineering, Construction and Operations (AECO) sector, which significantly contributes to global energy-related CO_2 emissions [3] and has been slow to adopt new technologies and

digitalisation [4]. Several factors have contributed to this delay, such as the fact that AECO is a fragmented industry with multiple stakeholders, along with a marked risk aversion and high inertia in adapting to new developments due to the high cost of operations [5–7].

Models, as mathematical representations of systems and processes, play a vital role in DTs by providing insights through simulations under different conditions [8]. Although models are typically trained and scaled offline, they are employed online for simulation [9] and require regular model updating based on time or criteria basis [10], depending on the application. Factors such as parameter evolution, changing ambient conditions, performance metrics or data drift may force the model update to adapt to the evolving patterns.

While models have the potential to serve as a data source, DTs often encounter data scarcity challenges, particularly in the AECO sector, where deploying realistic models is complex and conducting tests is expensive, time-consuming and difficult to scale. Historical data in this domain are often limited, unstructured and incomplete [11]. Moreover, available data frequently lack interoperability standards (e.g., Industry Foundation Classes [IFC]), ontologies (e.g., Building Information Modelling [BIM]) and other industry-specific formats [12]. The scarcity of data is widely recognised as a challenge by researchers and industry practitioners [13–16], as it directly affects the accuracy and reliability of the DT models and its decision-making capability. The economic impact of data scarcity can lead to figures in cost overruns of up to 80% and planning delays of 20% as a result of suboptimal resource allocation and inefficiencies due to insufficient data-driven decision-making [17].

Previous attempts to address this matter include scaled laboratory tests, historical data from previous observations, field surveys and numerical simulations, such as the finite element (FE) method. These methods can be used to calibrate and test models, although they are not the optimal source for training, as they fail to reproduce the diversity of real-world situations, without fully capturing the inherent complexity of reality [18].

An alternative approach involves generating synthetic data by mirroring and enhancing the properties of a foundational dataset [19]. Synthetic data can be generated to meet the model's requirements in terms of data size, diversity, complexity and class balance [18], expanding the characteristics of the real data. Industries such as medicine [20], pharmacy [21] and finance [22] have already embraced synthetic datasets to develop models, with the intention of leveraging the knowledge from real data.

Among the artificial intelligence (AI) developments capable of solving complex problems as those related to real-time damage assessment for DT applications, state-of-the-art techniques such as generative adversarial networks (GANs) and convolutional neural networks (CNNs) are well suited due to their related ability. GANs, which were conceived by Ian Goodfellow in 2014 [23], have revolutionised the generation of high-quality synthetic data [24–28] that closely resembles the real world. They are particularly useful in scenarios where there are no data or when a limited amount of real data is available, enabling the simulation of multiple scenarios, which is crucial for testing the robustness of the DTs. Meanwhile, CNNs [29] are specifically designed for tasks involving spatial hierarchies and local patterns [30, 31] such as those related to data in the AECO industry, where spatial relationships in structural features are common.

In the context of this research, a GAN model is formulated employing CNNs in both the generator and the discriminator, due to their flexibility and ability to manage complex grid-like shape data, respectively. This Wasserstein generative adversarial network with gradient penalty (WGAN-GP) and CNN architectural combination leverages the generative capabilities of the WGAN-GP and the feature extraction prowess of CNNs, enhancing the overall performance and robustness of the model. By training a CNN-based GAN on existing data, the underlying patterns are learnt and the model generates new original samples for robust model training and scenario testing. At the same time, this type of training preserves sensitive information and reduces data collection costs in the AECO industry.

The present work develops a data generation framework with a particular focus on the role that the generated data play in the diagnostic and prognostic capabilities of the DT in the AECO sector. The scope of the proposed framework is designed for structures of limited size where pivotal locations can be precisely defined at specific points and their features can be observed through discrete measurements, thereby capturing the utmost critical behaviour exhibited by the entire system. This study does not address the global assessment of damage to large structures achieved by continuous vibrational evaluations, and future research is fostered to explore these dimensions.

Figure 1 provides a visual representation of the proposed methodology as elucidated in this study. The schematic illustration takes the form of an algorithm, encapsulating the key steps and components that integrate the developed approach.

The research gap related to the use of discrete information for DT applications is addressed in this work, as it differs from the commonly used continuous information sources such as vibration signals and other time series data [32–34]. Discrete data are more prevalent in AECO applications because of the challenge of obtaining long-term measurements for slow processes. Furthermore, the use of discrete data is advantageous for the DT deployment, as it facilitates synchronisation, computation on the edge and minimisation of energy consumption in wireless Internet of Things (IoT) sensor devices [35]. It is worth mentioning that edge computing enhances DT capabilities by providing seamless operation near the data source, which increases cybersecurity, reduces the latency of data transmission and reduces the response time for real-time decision-making.

A demonstration of the present method is provided through a specially designed experimental pilot. The proposed case study has focused on the diagnostics and prognostics of damage in a laboratory-scale building-shaped frame. This structure was subjected to a number of load cases in a variety of damage states, where a limited amount of

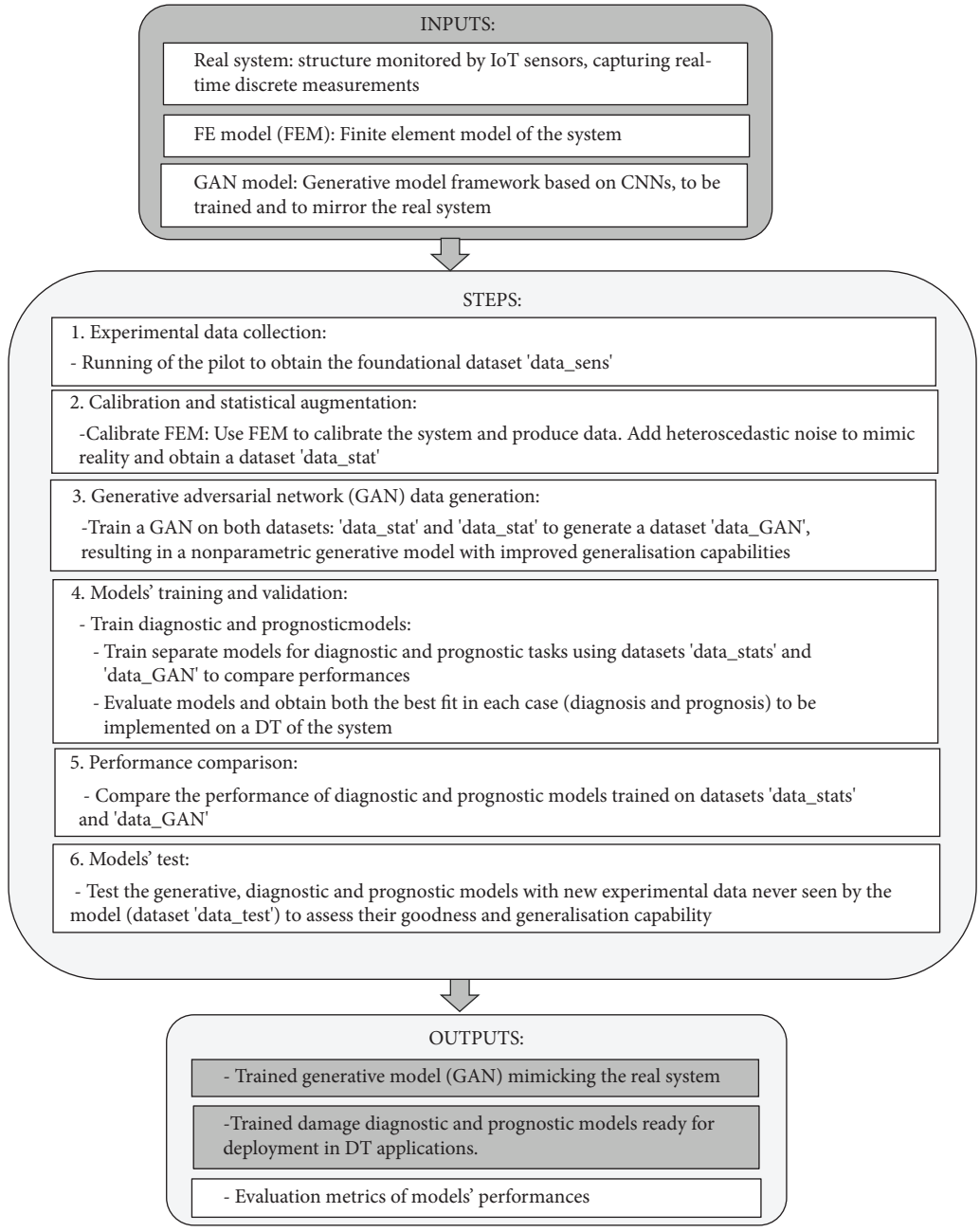


Figure 1: Overview of the proposed data generation framework within the context of the digital twin.

structural health monitoring (SHM) data is available. In this context, the proposed CNN-based GANs are able to produce a realistic synthetic dataset that supplements the limited SHM data. The resulting dataset thoroughly covers all damage scenarios and the range of forces that the structure can withstand without compromising its integrity.

With the purpose of highlighting the effectiveness of the proposed approach, two CNN-based models have been trained using the generated datasets: a damage diagnosis classifier and a prognostic regressor for the Remaining Useful Life (RUL) of the structure. The results show that the method is capable of detecting structural damage with an accuracy of 92% in classifying damage, regardless of the damage stage tested, and predicting the structural RUL with a mean absolute error (MAE) of 12 years over 100, with a goodness of fit (R^2) of 0.9. Following the training phase, a testing phase has evaluated the models' ability to generalise and perform well on unseen data, thereby avoiding potential issues such as overfitting. Finally, the paper demonstrates how the proposed method can be used for operational decision-making within a DT context. Decisions made under this approach are distinguished by their real-time nature, relying on online data streams through the IoT-enabled SHM systems.

In summary, this work aims to address the challenge of data scarcity in SHM, which harnesses the effective adoption

of the DT technology within the AECO sector. The limited availability of data in quality and quantity, essential for training robust DT models, hinders their full potential in providing predictive capabilities for managing structural assets in this industry. Despite classical methods, which barely produce data for calibration and testing due to their limited size and working scenarios, this work presents an innovative approach spanning from data generation to the application of these data for both training and evaluation on the models of the DT system. The methodology consists of the deployment of a CNN-based architecture which serves to generate the data through a GAN and provides the diagnostic and prognostic capabilities of the DT at the same time that test the goodness of the data generated.

The novel contribution lies in the comprehensive innovative approach, which covers from the generation of synthetic data to the training and testing of these data in the DT system's models, and the use of discrete measurements instead of the continuous vibrational approach, typical in SHM developments.

The structure of the remaining paper is organised as follows: Section 2 (Methodology) presents the mathematical description and background of the procedures employed in this work. Section 3 (Case Study Description and Problem Configuration) outlines the practical context in which these methods are applied and provides the model configuration details of the CNN-based GANs. The outcomes of the data generation and validation, along with their impact on model training for damage diagnostics and prognostics, are presented in Section 4 (Results and Discussion). Section 5 (Application Within the DT Context) details the practical integration of the achieved outcomes within the DT framework and the context of decision-making. Finally, Section 6 (Conclusions) delivers the final remarks and summary.

2. Methodology

In engineering, synthetic data generation is a common practice achieved by simulating various scenarios through experimental pilots and/or computational models. The models are further elaborated by incorporating noise or other types of perturbation into the results obtained [36]. This technique not only increases the amount of data but also enhances its diversity. Consequently, it enriches the data-driven models with a better understanding of the underlying physics and improves their robustness.

Within the classical paradigm, the noise introduced typically conforms to a particular statistical distribution, whereas perturbations manifest themselves as diverse geometric transformations, including rotation, translation and scaling. However, existing methods face several limitations that hinder their effectiveness and reliability. Most of them are computationally intensive and time-consuming, making it challenging to scale up the data generation process for large-scale applications and their subjacent predefined statistical distribution and geometrical transformations fail to capture the full diversity and variability inherent in real-world data, lacking robustness and adaptability to unusual situations [18, 24].

Emerging approaches for this purpose encompass AI-based techniques, including deep generative models such as GANs, variational autoencoders (VAEs), autoregressive models (AMs), flow-based models (FBMs) and energy-based models (EBMs), which have gained considerable attention and utility across diverse domains and recently in engineering and DT [37]. These advanced models offer multifaceted applications that effectively address the challenges associated with limited data availability, anomaly detection, damage localisation and predictive maintenance, to cite but any [38]. Within these methodologies, GANs stand out as particularly adept at capturing the underlying distribution of real data, yielding the generation of more realistic synthetic datasets [39]. The superior performance of GANs can be attributed to their ability to capture intricate patterns, while requiring fewer assumptions about the data distribution.

In any case, attention should be paid to ensure the quality and authenticity of the synthetic data. Transparency in the methodology used for their generation embraces explicability and instils confidence in the reliability of the synthetic data. Regarding authenticity, the traceability of the data throughout its lifecycle is essential, verifying the origin, history and transformation of the data, providing a documented history that guarantees that the data have been handled appropriately. In addition to this, there are qualitative and quantitative metrics that evaluate the goodness of synthetic data according to its intended purpose, which will be further detailed in this section.

2.1. GANs. A GAN consists of two neural networks: a generative model or generator, referred to as G, and a discriminative model or discriminator, D. These networks are trained concurrently through a mix-max game, leading to the development of a generator model capable of producing synthetic data that align with the distribution of the real data. To illustrate the main structure of a GAN, Figure 2 presents a schematic representation.

Let $x \in \mathcal{X} \subseteq \mathbb{R}^d$ be the vector representing the real data and π_r: $\mathcal{X} \longrightarrow \mathbb{R}^+$ a probability density function (PDF) associated with these data. Similarly, we denote $\tilde{x} \in \tilde{\mathcal{X}} \subseteq \mathbb{R}^d$ as the latent noise vector, which follows the PDF π_g: $\tilde{\mathcal{X}} \longrightarrow \mathbb{R}^+$ (typically a multidimensional zero mean Gaussian).

The generator G takes $\tilde{x}$ as input and maps it to a vector $G(\tilde{x})$ of dimension d. The discriminator D takes both the real data for training x and the generated vector $G(\tilde{x})$ as inputs. Then, it outputs the probability that the generated sample $G(\tilde{x})$ belongs to π_r. The training of D involves maximising the probability of correctly assigning labels ('real' or 'fake') to the generated samples. Simultaneously, the training of the generator aims to minimise the probability of the discriminator classifying the generated samples as fake. Both training stages of the GAN are conducted using the following loss function [40] described in the following equation:

$$L_{\mathrm{GAN}} = \mathbb{E}_{\pi_r}[\log D(x)] + \mathbb{E}_{\pi_g}[\log(1 - D(G(\tilde{x})))]. \quad (1)$$

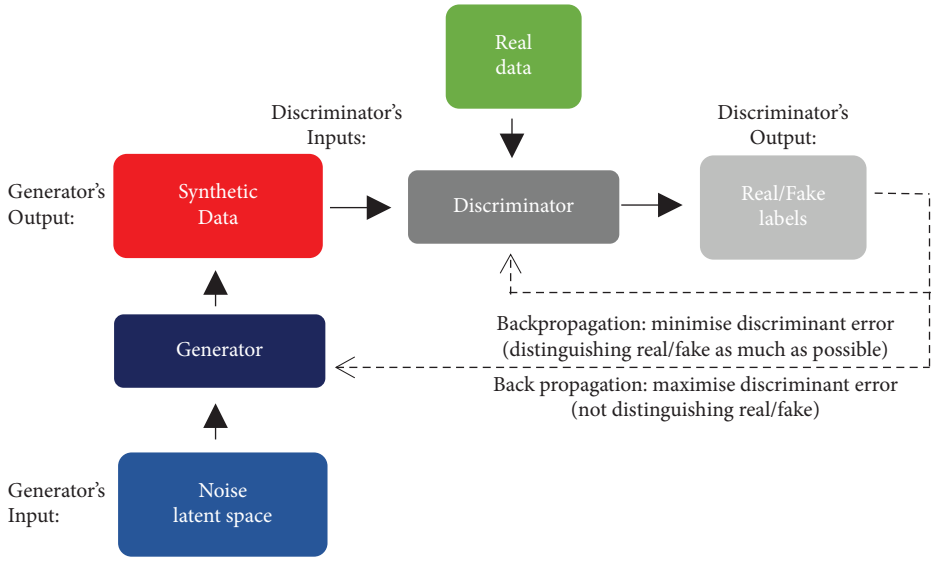


Figure 2: Structure of a generative adversarial network (GAN).

Note that the discriminator D aims to distinguish between inputs originating from the real data distribution and those generated by the generator. Its objective is to maximise the probability of correctly classifying the generated samples as fakes. Conversely, the generator G strives to produce synthetic samples that closely resemble real data, with the goal of minimising the probability of the discriminator classifying a generated sample as fake. Once trained, the GAN can generate synthetic data that follow a distribution similar to the real training dataset [41].

To enhance the stability of GANs, various improvements have been proposed to optimise the objective function, with the recent introduction of the WGAN-GP [42, 43]. The WGAN-GP demonstrates superior performance compared to the original GAN, addressing issues such as mode collapse (where generated samples cluster in specific regions) and facilitating more consistent training. In WGAN-GP, the discriminator is often referred to as the 'critic' since its focus is not on classifying real or fake samples, but on determining the degree of belief, providing confidence or reliability estimates for the generator's predictions [44]. By introducing the Wasserstein-1 distance, the WGAN-GP improves the training stability with minimal hyperparameter tuning, compared to the original GAN [45]. The Wasserstein-1 distance is defined as follows:

$$W\left(\pi_r, \pi_g\right) = \inf_{\prod\left(\pi_r, \pi_g\right)} \mathbb{E}_{\gamma \sim \prod\left(\pi_r, \pi_g\right)}[\|x - \tilde{x}\|], \tag{2}$$

where $\prod(\mathbb{P}r, \mathbb{P}g)$ denotes the set of all joint distributions over π_r and π_g and the function $\gamma(x, \tilde{x}) \sim \prod(\cdot)$ can be interpreted as the measure of the mass that must be transported from x to $\tilde{x}$ in order to transform π_r into π_g. Consequently, the infimum distance corresponds to the cost of the optimal transport plan.

To ensure the stability of the training process, the WGAN-GP incorporates a gradient norm penalty for random samples, which augments the Wasserstein loss and achieves Lipschitz continuity in the critic. Thus, the objective function of the WGAN-GP is described as follows:

$$\begin{aligned} L_{\text{WGAN-GP}} &= \mathbb{E}_{\pi_g}[f(\tilde{x})] - \mathbb{E}_{\pi_r}[f(x)] \\ &\quad + \lambda \mathbb{E}_{\hat{x} \sim \pi_u}\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x})\right\|_2 - 1\right)^2\right]. \end{aligned} \tag{3}$$

In equation (3), f is a 1-Lipschitz continuous function and π_u is the distribution obtained by uniformly sampling $\hat{x}$ along a straight line between the real and generated distributions π_r and π_g. This is justified by the fact that the optimal critic has straight lines with unit gradient norm between the samples coupled from π_r and π_g. The term λ is the penalty coefficient used to weigh the gradient penalty term.

2.2. CNNs. As indicated previously, both the generator and discriminator in this study are founded on CNNs. A CNN is a multistage neural network employed for spatial pattern recognition [46], which consists of a filter phase followed by a classification or prediction phase. This architecture enables hierarchical learning and information extraction in subsequent layers. The filter stage involves convolutional, batch normalisation, activation and pooling layers. The classification/prediction phase utilises fully connected dense layers to establish pattern relationships. Regarding architecture, a small multilayer CNN is enough to train a full regular-size model. CNNs deepen the models by increasing the number of layers needed, but at the same time, they contribute to reducing the number of neurons, as CNNs get good representations of the input signals, improving the overall performance of the whole network [47].

Convolution is a mathematical operation in which a kernel, also referred to as a filter, slides over the input data computing dot products with local receptive fields [48]. This process generates feature maps that indicate the presence of specific patterns. As the CNN progresses, it captures increasingly complex representations, which are utilised by subsequent layers, such as pooling and fully connected layers, for classification, prediction or further processing.

For 1-D CNNs, the convolutional layer applies a sliding time window along the feature series axis to obtain subsequences. Each subsequence is element-wise multiplied with the kernel to obtain the convolution result [49]. The computation of each unit in the convolutional feature signal can be expressed as the following equation:

$$C_m = f\left(\sum_{l=1}^{L} S_l \otimes W_{l,m}\right). \tag{4}$$

Here, S_l represents the l-th input feature signal or subsequence and $W_{l,m}$ is the weight matrix connecting the l-th input feature signal to the m-th output convolutional feature signal. The sign $\otimes$ denotes the convolution operation, f is the activation function, and finally, C_m represents the output, namely, the featured map or the convolutional featured signal [50].

Following the convolution operation, the pooling process is applied to reduce the dimension of the feature signal and enhance the feature invariance for small disturbances. The pooling function that demonstrates better performance in most cases is the maximum function [51], defined as equation (5), where N is the pooling size and q determines the degree of overlap of adjacent pooling windows:

$$p_m = \max_{n=1}^{N}\left(C_{m(q+n)}\right). \tag{5}$$

In each 1-D CNN layer, a forward propagation occurs, which is given by equation (6), where x_k^l represents the input, b_k^l is the bias of the k-th neuron at layer l, s_i^{l-1} is the output of the i-th neuron at layer $l-1$, and w_{ik}^{l-1} is the kernel from the i-th neuron at layer $l-1$ to the k-th neuron at layer l. Finally, $C(\cdot)$ denotes the 1-D convolution operation without zero-padding:

$$x_k^l = \sum_{i=1}^{N_{l-1}} C\left(S_i^{l-1}, w_{ik}^{l-1}\right) + b_k^l. \tag{6}$$

In the last step, the intermediate output y_k^l is obtained by applying the activation function $f(\cdot)$ to the input x_k^l as in the following equation:

$$y_k^l = f\left(x_k^l\right). \tag{7}$$

CNNs are typically trained in a supervised manner using the stochastic gradient descent or backpropagation (BP) algorithm [52]. During each iteration, the gradient magnitude of the network parameters is computed, including the weights of the convolutional and the fully connected layers. These parameter sensitivities are then used to update the CNN parameters iteratively until a stopping criterion is met. The BP method is a well-known procedure in the literature [50] and leads to a CNN architecture that efficiently captures spatial invariance, identifying relevant patterns in the input data regardless of where they occur along the 1-D sequence while maintaining parameter efficiency.

3. Case Study Description and Problem Configuration

This section presents a case study that aims to serve as an application scenario of the procedures presented in Section 2 (Methodology). This application has been intentionally streamlined for the sake of clarity while preserving the significance of the method. The computational developments employed in this case study are open source and compatible with Python [53], drawing on resources previously developed by the authors in [2].

3.1. Problem Definition. The case study implemented involves a 6-storey 1.5-m high laboratory-scale steel frame structure and its corresponding DT, which is informed by displacement and force sensors (Figure 3). The structure is subjected to variable lateral forces and undergoes simulated damage due to the progressive loosening of the bolts. The displacements of the individual storeys were recorded using wireless IoT systems employing ultrasonic methods, whereas the force was measured by a digital transducer.

It should be emphasised that one of the main challenges in this experiment was the use of basic and cost-effective sensors with limited sensitivity and moderate accuracy, transmitting data via the IoT, rather than using professional wired data acquisition systems. This sensing system adheres to the flexible open-source principles of this research and also introduces additional uncertainties in the data that need to be addressed by the DT.

The virtual twin is computationally simulated using a FE model developed in OpenSeesPy [54] considering only the static behaviour according to the following equation:

$$F = K(t) \cdot d, \tag{8}$$

where K is the stiffness of the metal evolving over time (t) due to the progression of the damage, F is the infringed force, and d is the response of the tower in the form of displacement. Both forces and subsequent displacements occur at discrete time steps and are independent of the initial conditions.

Damage is caused by the progressive loosening of the joints when forces are applied. Once it starts, it is assumed to increase with time over a period T, with $T = 100$ (years) being the accepted useful life of a civil structure. The loosening of the bolts has been linearly estimated to reach a 60% reduction in the stiffness of the structure after 100 years of intermittent forces, causing the end of life of the structure. This leads to a stiffness reduction coefficient (α) in the system due to bolt loosening as follows:

$$\alpha = 1 - 0,006 \cdot t, \tag{9}$$

with α varying from 0 to 1 and t being the time of damage, expressed in years.

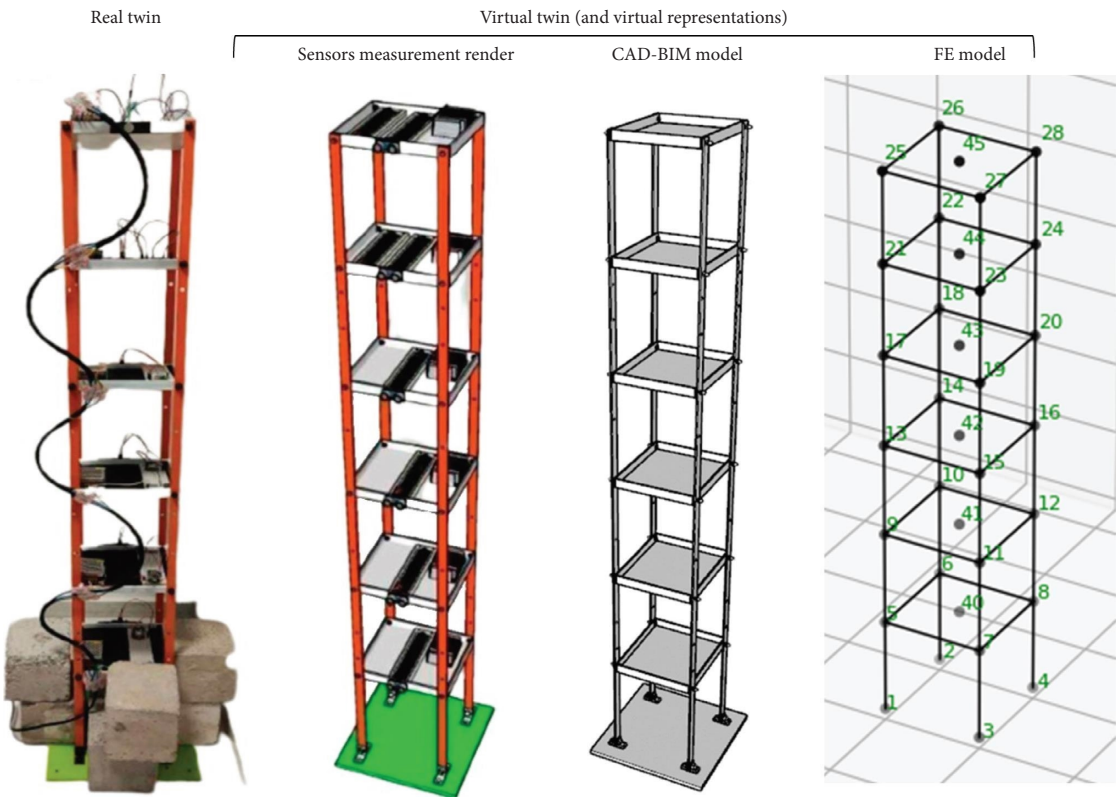


FIGURE 3: Case study: the digital twin of a metal tower.

In the laboratory pilot test, it was only possible to reproduce three scenarios due to experimental limitations: (1) a healthy structure, (2) a semidamaged structure with medium bolt loosening and (3) a fully damaged structure. However, in the FE model, a full range of possible scenarios was simulated by progressively reducing joint stiffness on all floors, one floor at a time, affected by different random forces ranging from 0 to 450 N (the maximum affordable load of the structure), obtaining the displacement of the stories in each scenario.

3.2. Dataset Generation. A first dataset (data_sens) was created with measurements coming from the sensors and corresponding to the three scenarios aforementioned: (1) a healthy tower, (2) a partially damaged tower with medium loosening of the bolts and (3) a fully damaged tower with total loosening of the bolts (Figure 4).

The FE model was calibrated to these trial data, achieving high accuracy in the results for the three scenarios, as shown in Figure 5. The resulting dataset (data_sens) was composed of these experimental data and only covers specific cases with limited forces, lacking values within the range of 250 to 450 N due to the difficulties in the practical aspects of the experiment.

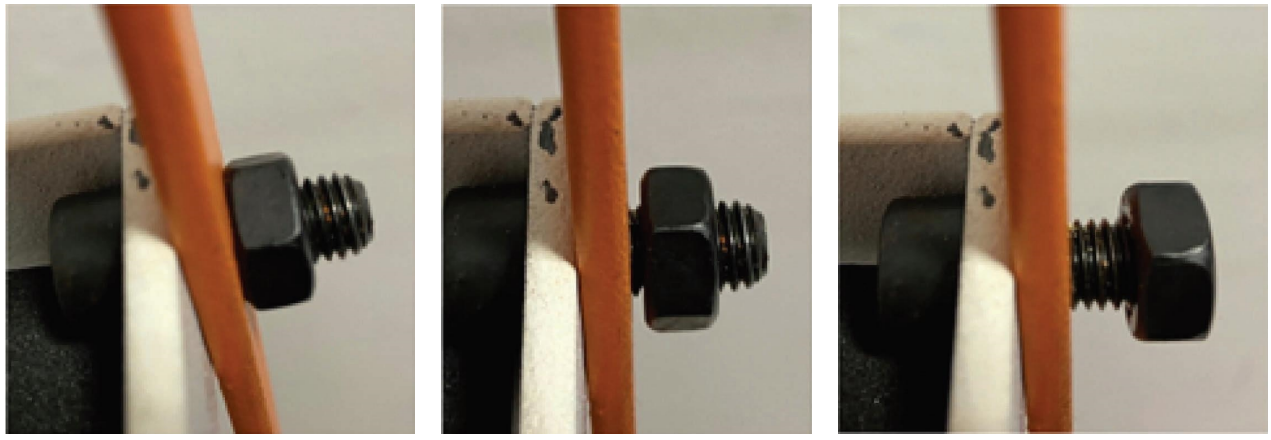
By applying statistical approximation techniques, another dataset is created (referred to as data_stat) simulating the original statistical distribution, assimilated to Gaussian. The data were observed to show heteroscedasticity: The variance of the y values increases with increasing values of x (Figure 6). This is a common phenomenon in engineering [55], where noise can be input-dependent. Considering equation (8) as a linear regression problem and taking $i = 1, \ldots, N$ (number of measurements), the displacements are obtained in the following equation:

$$d_i = K_i^{-1}(t_i) \cdot F_i + \varepsilon_{i(t_i)}, \tag{10}$$

where the dependent variable d_i is equal to the independent random variable F_i times a coefficient plus a random disturbance term that has zero mean and variance depending on both the value of F_i with $\sigma_i^2 = F_i \cdot \sigma^2$ and the time of damage t_i, as it varies from the healthy case ($t = 0$) to the fully damaged ($t = 100$).

Following this approximation, each displacement d_i corresponds to a force F_i distributed as a Gaussian G with a mean (μ_i) equal to the FE force calculated for that displacement and a variance (σ_i^2) that linearly depends on both the force F_i and the time since the damage began t_i. According to the geometry and material properties of the tower, the values of the main parameters are shown in Table 1.

According to the procedure described previously, a dataset with a tensor shape (the third dimension being equal to the time of damage) can be produced covering all possible scenarios considered in this case study, namely six combinations of damage (one per floor) in the case of bolt loosening, with a time span ranging from 0 to 100 years since the damage began. The results are given in Figure 7.



(a) (b) (c)

FIGURE 4: Bolts' loosening: healthy state (a), medium damage state (b) and fully damaged (c).

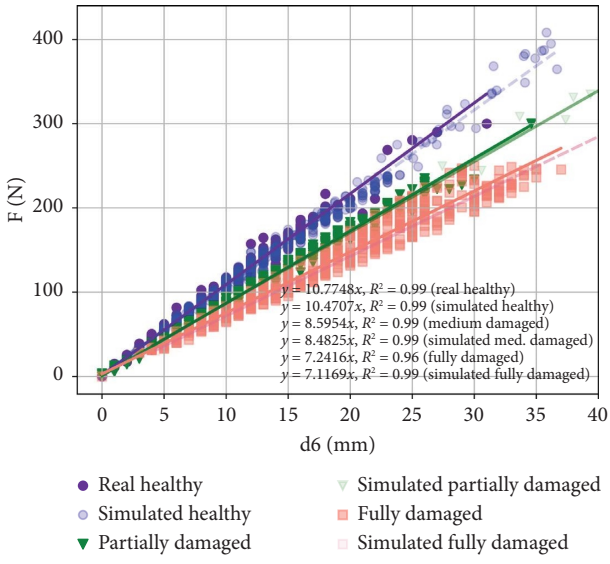


FIGURE 5: Calibration of the FE model in 3 different scenarios with data coming from the sensors (data_sens).

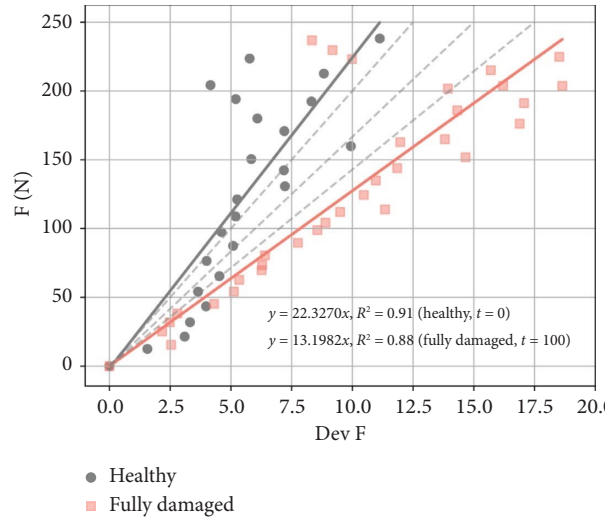


FIGURE 6: Heteroscedasticity of the variance depending on the force F_i and on the time t_i from the healthy state ($t_i = 0$) to the fully damaged ($t_i = 100$).

It is worth highlighting the paramount importance of creating a dataset that closely mirrors real-world data, encompassing not only the error but also its variation (heteroscedasticity). This emphasis on dataset quality arises from the challenging objective of training a deep learning (DL) model capable of real-time diagnosis, discerning whether sensor measurements correspond to a healthy or damaged state, even without the need for preprocessing as occurs when operating at the edge. This means that the model needs to digest the error and understand it, finding a compromise between the bias and the variance trade-off.

Taking advantage of the two datasets: data_sens that is the experimental dataset obtained from the measurements of the sensor not covering the full case scenario and *data_stat* that is smarter in terms of bigger size, diversity, complexity and class balance, the field is prepared to launch the GAN method, which will mirror the real data distribution in a nonparametric form. In the context of generative models, the 'nonparametric' term refers to the model's ability to learn directly from the real data without explicitly defining its probability distribution, allowing models to generate completely new data points that closely resemble the real ones without assuming an underlying distribution. In this way, the training dataset *data_stat* will be given to the WGAN-GP, which will create *data_GAN* from randomness.

The validation and testing of the generated data are performed in Section 4 (Results and Discussion): First, a validation using a subset of 20% randomly partitioned from the datasets, *data_stat* and *data_GAN*, and subsequently tested with new values coming from the dataset, data_test, never seen before by the models. The criteria for stopping the GAN's training will be the stability reached by the loss functions and the metrics adopted, consisting of several quantitative figures: the aforementioned Wasserstein

Table 1: Parameters of the system (steel tower of 6 stories, 1.5-m high).

Name	Value	Units
Young's modulus (E)	$2.10 \cdot 10^{11}$	N/m^2
Shear modulus (G)	$8.10 \cdot 10^{10}$	N/m^2
Poisson coefficient (ν_i)	0.3	—
Equation of the std. deviation of F (dev F)	$\sigma_i = 0.0758 \cdot t_i + 0.0448$	N

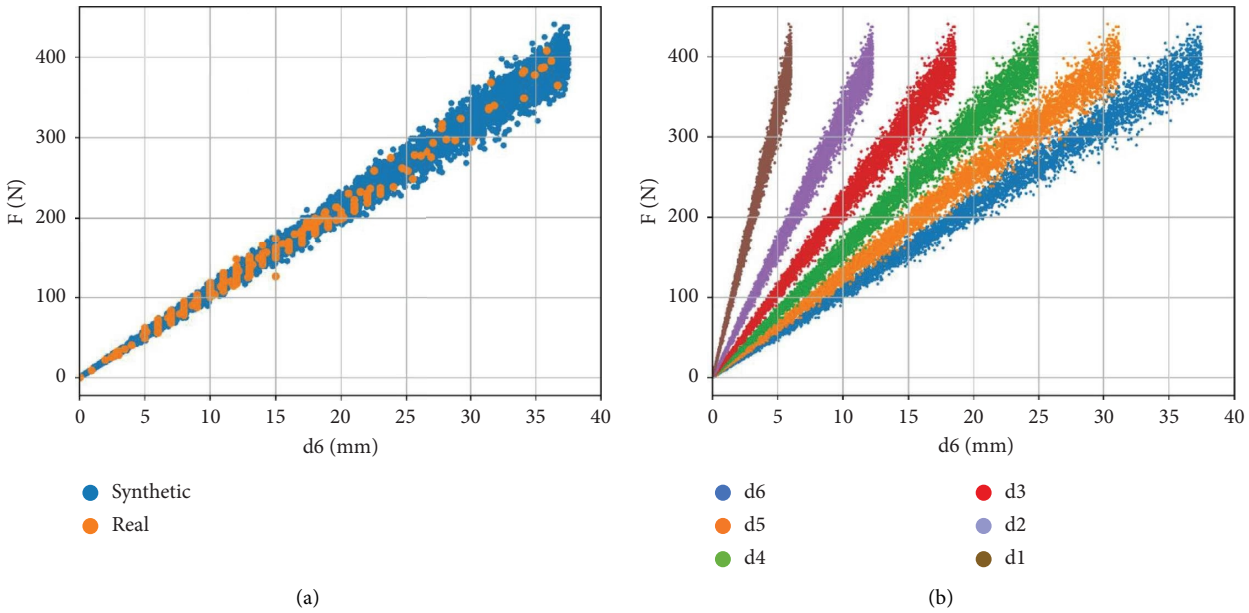


Figure 7: Data produced by statistical approximation (data_stat) for the healthy state (time $t_i = 0$). (a) Displacements on the sixth floor, synthetic and real. (b) Displacements on each floor, first to sixth.

distance [42] between the generated and the real distribution, the Frechet Inception Distance (FID) [56] and the Structural Similarity Index Measure (SSIM) [57], along with the qualitative visual inspection of the results.

Consistent with the previously mentioned, CNNs are employed within the generator and the critic due to their recognised proficiency in processing data with a grid-like shape [58], the same shape as the displacement vector over the six stories of the tower in the case study. The patterns shown by this signal (Figure 8) will enable the models to discern that when the tower is in a healthy condition, the signal exhibits a consistent linear tilt across the six stories.

On the contrary, when the tower is damaged, the signal will register different tilts depending on the location of the perturbed story and the time of damage. In this way, the resulting GAN can generate a realistic dataset considering all case scenarios for the healthy state and the damaged state for the range of forces that the tower is able to resist without compromising its integrity (maximum allowed displacement for a force of 450 N). Here, the label is '0' if the tower is healthy and '1' if damaged, and this information is treated as a feature by the CNNs.

The architecture of both the generator and the discriminator consists of a multilayered CNN (Figure 9), as noted earlier. The generator employs CNNs, batch normalisation and ReLU activation functions followed by fully connected dense layers; before the output layer,

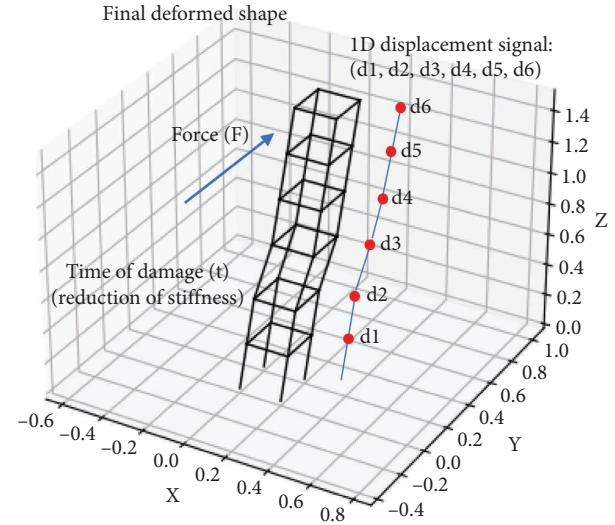


Figure 8: Plot of a 1-D displacement signal corresponding to a force F and a time t of damage affecting one of its floors.

a nonlinear hyperbolic tangent activation function was introduced. The discriminator also accounts for CNNs, layer normalisation and LeakyReLU which performs better with gradient penalty. In addition, dropout is also applied to avoid overfitting.

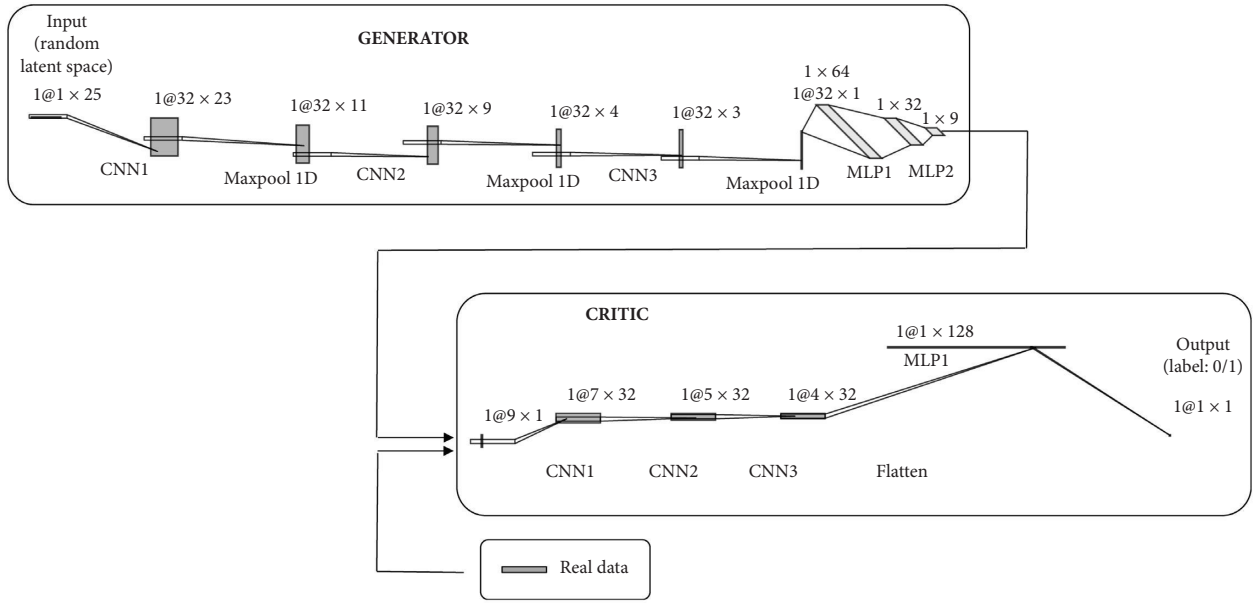


Figure 9: Configuration of the CNN-based WGAN-GP architecture.

The primary hyperparameters are set according to the recommended values for using the WGAN-GP [43] and are included in Table 2. The remaining parameters such as the number of neurons and layers, the architecture of each layer and the number of epochs were determined through an empirical refinement process. This procedure consists of a trial-and-error iteration until convergence is reached and the validation metrics reveal adequate performance.

The implementation deployed in Figure 9 and further detailed in Table 3 shows, at the convolutional stage, a first kernel size of 3, with a filter of 3 units of width in the input sequence, moving across with a stride of 1 and scanning 3 consecutive units at a time. In the final layer, the width was decreased to 2. Additionally, a filter of 32 guarantees that there will be 32 unique filters applied to the input data. So, the CNN will apply 32 different filters of width 3 to the input sequence, capturing several features at different positions in the sequence. Batch normalisation is implemented right after both the convolutional layers and the fully connected layers to accelerate the training process. Furthermore, dropout is also applied after every convolutional layer to prevent overfitting [59], together with a maxpool function performing local max operation to reduce parameters and obtain location-invariant features. Finally, the classification stage in this case was composed of two fully connected layers for classification, with decreasing sizes of 64 and 32.

The architectural framework presented will be subsequently evaluated in the context of the DT diagnostic and prognostic models in Section 4 (Results and Discussion). These models follow the same configuration, although adapted to their specific tasks: classification [60] and regression [61]. It is reasonable to assert that an architecture capable of generating authentic data is likely to demonstrate an inherent capacity for accurate interpretation, thus facilitating classification and prediction within real-world contexts.

Table 2: Hyperparameters of the WGAN-GP.

Hyperparameters	Values
Gradient penalty coefficient (λ)	10
Number of critic iterations per generator iteration (n_{critic})	5
Learning rate (α^1)	0.0001
Adam optimiser hyperparameters (β_1, β_2)	(0.5, 0.9)
Batch size	100
Latent space dimension	25
Dropout	0.3

4. Results and Discussion

This section presents the results obtained when using synthetically generated data on the proposed methodology, all within the case study described previously. Furthermore, in this section, an assessment of the diagnostic and prognostic models' performance is conducted. These models have been trained and validated on different subsets of the generated data to evaluate their accuracy and efficiency in the tasks of detecting damage and predicting the RUL of the system.

4.1. Synthetic Dataset Generation and Quality Evaluation. The results of data generation using the proposed CNN-based WGAN-GP approach are presented in Figure 10, which depicts the triads: {force, displacement, damage}. It should be noted that damage in this study is quantified in terms of years, following a temporal stiffness reduction law due to bold loosening, as described by equation (9). Both qualitative (visual comparison) and quantitative evaluations (numerical index values) are employed to assess the generated data.

Figure 11 illustrates a visual comparison between the generated dataset (represented by orange dots), the measured data (depicted by blue dots) and the statistically generated data (shown in green) at different damage levels, namely: healthy

Table 3: Architecture of the WGAN-GP model.

Generator	
Layer	**Shape**
Input (latent space)	1@1 × 25
CNN 1D (ReLU, 32, 3)	1@32 × 23
BatchNormalisation/Maxpooling	1@32 × 11
CNN 1D (ReLU, 32, 3)	1@32 × 9
BatchNormalisation/Maxpooling	1@32 × 4
CNN 1D (ReLU, 32, 2)	1@32 × 3
BatchNormalisation/Maxpooling	1@32 × 1
Flatten ()	1@32 × 1
Dense (ReLU, 64)	1 × 64
Dense (ReLU, 32)	1 × 32
Dense (Tanh, 1)	1 × 9
Output (F, t, d1, d2, d3, d4, d5, d6, label)	(9, 1)
Discriminator (critic)	
Layer	**Shape**
Input (F, t, d1, d2, d3, d4, d5, d6, label)	1@9 × 1
CNN 1D (LeakyReLU, 32, 3)	1@7 × 32
BatchNormalisation/dropout	
CNN 1D (LeakyReLU, 32, 3)	1@5 × 32
BatchNormalisation/dropout	
CNN 1D (LeakyReLU, 32, 2)	1@4 × 32
Flatten ()	1@1 × 28
Dense (, 1)	1@1 × 1
Output (critic's value)	(1)

state, medium damage state and fully damaged state, depicted as panels (a) to (c), respectively. The results demonstrate high accuracy in prediction when compared to the measured data, as well as the ability of the data generation approach to introduce novel data points including outliers and enrich the dataset in underrepresented regions of the testing dataset.

At this standpoint, it is worth mentioning that visually evaluating large-sized numeric data is more challenging compared to images, whose features can be visually perceived in the qualitative approach. Thus, the quantitative approach is desirable in such cases through the use of performance metrics, even when accounting for similarity at the same time that the novelty and complexity between distributions are also complex [57]. In this work, three metrics are used, namely the Wasserstein distance [42], the FID [56] and the SSIM [57].

The Wasserstein distance, according to equation (2), captures the 'cost' required to transform one distribution (real) into another (generated), accommodating distributions with non-conventional shapes and outperforming metrics like Euclidean distance or Kullback–Leibler divergence in such cases. The lower the Wasserstein distance, the better the GAN performance.

The FID, also known as the Wasserstein-2 distance, assumes that the real and the generated data follow a multidimensional Gaussian distribution and measures the distance between these two Gaussians in the feature space by calculating their respective mean and variance, as in equation (11). Again, a lower FID indicates greater similarity between the compared data:

$$FID(x,\tilde{x}) = \left\|\mu_x - \mu_{\tilde{x}}\right\|_2^2 + \mathrm{Tr}\left(\sigma_x + \sigma_{\tilde{x}} - 2\left(\sigma_x\sigma_{\tilde{x}}\right)^{1/2}\right), \quad (11)$$

where μ and σ stand, respectively, for the mean and the variance of the compared distributions and Tr refers to the trace linear algebra operation.

Finally, the SSIM evaluates the similarity between two datasets based on three aspects: inheritance, creativity and diversity. While creativity and diversity remain relevant for data types other than images, inheritance is no longer necessary. The score obtained from equation (12) ranges from 0 to 1, with 1 indicating exact similarity and 0 representing complete dissimilarity. In the context of GANs, the objective is to generate outputs that are similar to the real data (creative) but different from each other (diverse), and a maximum score of 0.8 is often considered:

$$\mathrm{SSIM}(x,\tilde{x}) = \frac{\left(2\mu_x\mu_{\tilde{x}} + C_1\right)\left(2\sigma_{x\tilde{x}} + C_2\right)}{\left(\mu_x^2 + \mu_{\tilde{x}}^2 + C_1\right)\left(\sigma_x^2 + \sigma_{\tilde{x}}^2 + C_2\right)}, \quad (12)$$

with σ_x and $\sigma_{\tilde{x}}$ being the variances of the compared distributions, $\sigma_{x\tilde{x}}$ the covariance, and C_1 and C_2 the constants to stabilise the division, typically set to 0.01 and 0.03, respectively.

Table 4 presents the metrics obtained during training and testing, demonstrating good results for the metrics derived from the WGAN-GP in generating a dataset of over 100,000 samples. Note that although the test values are expected to be worse than the training values, they remain above acceptable thresholds.

The stability of the training process is another crucial factor in the evaluation, as it should remain consistent to ensure convergence. As depicted in Figure 12, during the initial stages of training, both the critic and the generator experience some peaks, almost imperceptible in the case of the critic. The critic takes the lead in this process, as it has access to the training dataset and possesses more knowledge than the generator. However, after a few epochs, the generator begins to effectively learn the gradients and generate data that closely resemble real samples. Eventually, both the critic and the generator converge, indicating the successful training of the GAN model. The number of epochs was fine-tuned through an iterative process of trial and error.

4.2. Model Training Using Synthetic Datasets. Damage-induced modifications in a structure, such as stiffness and mass reduction, lead to alterations in its displacement response, which can be effectively assessed through SHM. In this context, machine learning supervised techniques are employed to train models using the synthetic datasets generated in the preceding section. The objective is twofold: further test the generated dataset and detect damage in the laboratory-scale structure while predicting its RUL.

For this purpose, damage detection and damage prognostic models will be trained and validated on the aforementioned synthetic datasets. The performance validation of these is made according to the 'train-validation split' procedure. In this approach, the dataset is divided into two subsets: the training set (80% of the data is used to train the model) and the validation set (a portion of 20% of the dataset

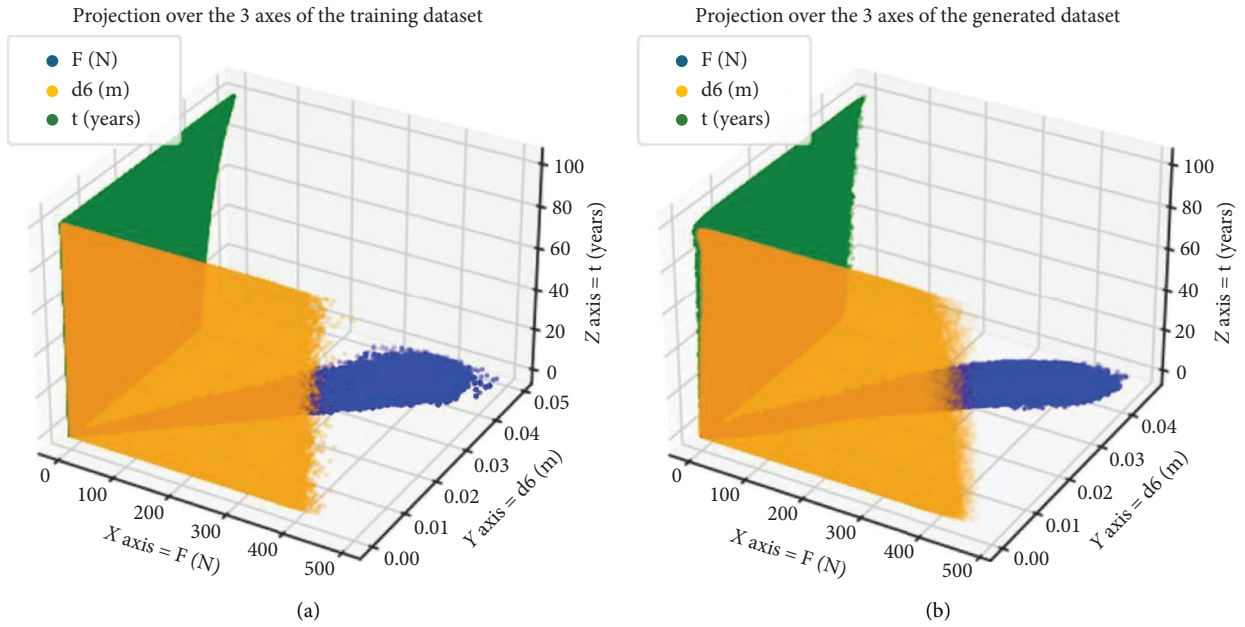


FIGURE 10: Three-dimensional visual comparison of the outcomes using the proposed data generation method. The x-axis represents the applied force (F, in Newtons), the y-axis corresponds to the displacement on the top floor (d6, in millimetres), and the z-axis depicts the duration of damage (t, in years). Panel (a) shows the dataset generated during the GAN training process utilising statistical approximation (data_stat). Panel (b) depicts the dataset produced using the proposed CNN-based WGAN model (data_GAN).

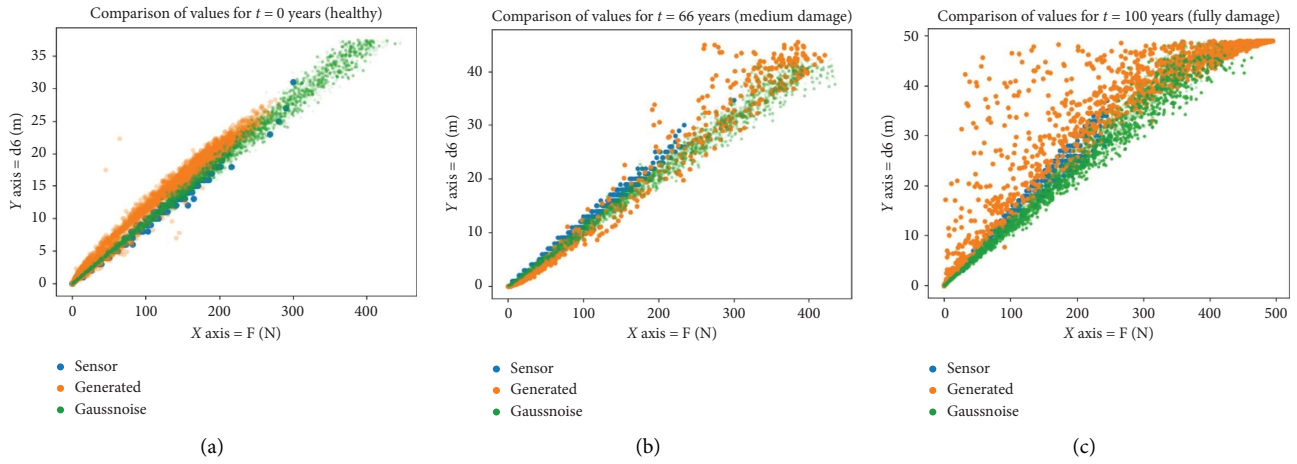


FIGURE 11: Visual comparison of three datasets: data_sens, data_stat and data_GAN. Panel (a) shows the dataset corresponding to $t = 0$ (healthy state); panel (b) depicts the dataset corresponding to $t = 66$ years (medium damage state) and panel (c) presents the dataset corresponding to $t = 100$ years (fully damaged state).

TABLE 4: Results in terms of performance metrics of the proposed GAN applied to the case study.

Metric	Training	Test
Wasserstein distance	0.75	2.18
Frechet Inception Distance (FID)	33.12	265.98
Structural Similarity Index Measure (SSIM)	0.63	0.60

is reserved for validating the model, helping fine-tune the hyperparameters and prevent overfitting). The data are randomly shuffled before splitting to avoid biases in subset composition. The final evaluation of the models is performed on a separate dataset (external to the training and validation sets), which assesses how well the models generalise to new unseen data.

The datasets employed in the development of this case study are described in Table 5, and their main statistical descriptors are included in Table 6:

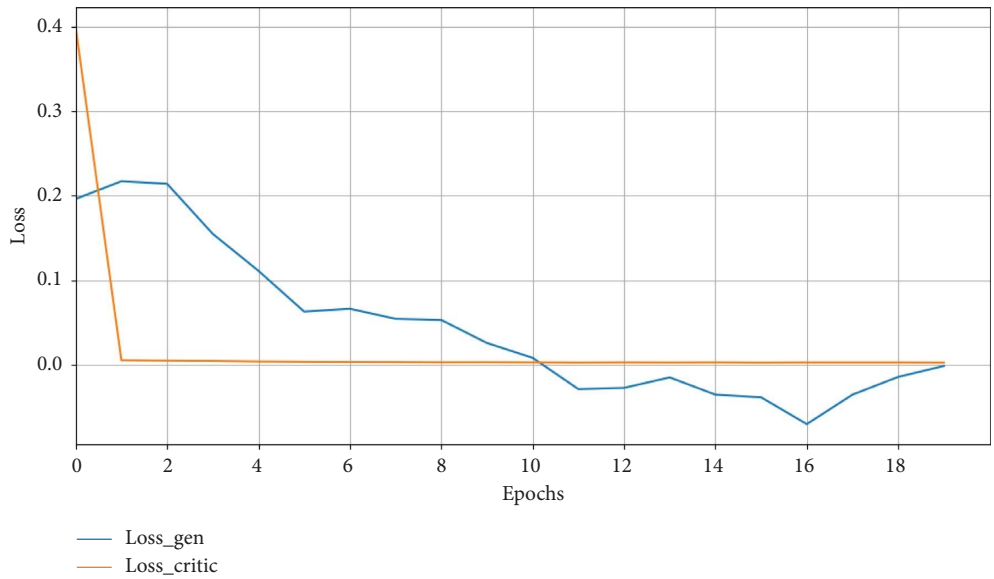


FIGURE 12: Plot of the loss values in generator and critic versus the number of epochs.

The sizes of the datasets involved in this research aim to balance the benefits and drawbacks associated with large magnitudes. The reason is that the disadvantages, such as extended training time and increased computational resource demands, may outweigh the benefits derived from the prospective incorporation of pertinent supplementary information. In this study, the sizes of the experimental datasets (data_sens and data_test) align with the typical dimensions of a test sample, representing a day-long survey with data recorded at a frequency of 1 data point per minute for 24 h. On the other hand, the sizes of synthetically generated datasets are flexible and tailored to the specifications of the research. The initial statistically produced dataset (data_stat) exhibited considerable volume (600 k); nevertheless, it was noted that smaller sizes yield comparable results in training. Consequently, a second dataset (data_GAN) was generated with a reduced size (115 k), yet it showed comparable performance.

4.2.1. Training for Damage Detection. CNNs are applied at this point in the diagnostic model due to their aforementioned capacity to extract complex geometric patterns within the data [62]. The damage diagnostic model consists of a binary classifier discerning between a healthy and damaged state. It is trained using the dataset produced by statistical methods *data_stat* along with the dataset generated by the proposed GAN *data_GAN*. During training, 20% of the data was reserved for validation in each case. In both instances, 100 epochs were sufficient for the model to learn the features of the problem, namely the forces and displacements of the six stories of the structure. The architecture of the diagnostic model is detailed in Table 7.

Performance metrics, including accuracy (correctly predicted labels over the total), recall (ratio of true positives [TPs] out of all correctly predicted values) and precision (fraction of TPs out of the real and false positives [FPs]), are employed to evaluate the model's performance, and the results are presented in Table 8.

Note that the performance metrics in training and validation when using data_stat are very similar because both datasets come from the same distribution. The model tends to overfit with this type of training dataset, resulting in high rates. However, the performance metrics are slightly different when the model is trained on data_GAN. It is also notable that the learning curve shown in Figure 13 is more progressive when training with data_GAN due to the diversity of the data, as depicted in Figure 13(b).

Furthermore, the results are analysed using a normalised confusion matrix [63], which categorises the model predictions into TP, FP, false negative (FN) and true negative (TN) based on the comparison between the predicted labels and the true labels.

The normalised confusion matrix demonstrates perfect accuracy (1.0) in detecting the healthy state for the model trained on data_stat (Figure 14(a)), while achieving a value of 0.98 for detecting the damaged state. The results also show that, for earlier stages of damage that are often negligible, there is a tendency for FNs (0.02). It is important to note that in this model, the label '0' represents a healthy state, so the aforementioned situation might imply misdiagnosing a damaged state as healthy (FN), which can be risky. This tendency can be addressed by employing techniques such as oversampling early damage cases. On the other hand, the model trained on data_GAN (Figure 14(b)) exhibits slightly lower performance, achieving a value of 0.96 for detecting the healthy state and 0.92 for detecting the damaged state. Once again, there is an increasing trend of FNs (0.08), which can be rectified, as mentioned above.

4.2.2. Training for Damage Prognostics. The prognostic model employed in this study also uses a CNN architecture, which enables the estimation of the structural RUL by

Table 5: Description of the datasets involved in the case study: data_sens, *data_stat*, *data_GAN* and data_test.

Name	Description	Size	Type	Format	Features	Space definition	Collection method
data_sens	Data collected from the real system, serving as the foundational dataset	2000	Numeric	JSON	(t, F, d1, d2, d3, d4, d5, d6)	3 mm resolution	IoT sensors
data_test	Data collected from the real system, serving as a tester	1100	Numeric	JSON	(t, F, d1, d2, d3, d4, d5, d6)	3 mm resolution	IoT sensors
data_stat	Data produced with a FE model with added noise	600,000	Numeric, categorical	CSV	(t, F, d1, d2, d3, d4, d5, d6, label)	1 mm	Statistical generation
data_GAN	Synthetic data generated from the datasets data_sens and *data_stat*	115,000	Numeric, categorical	CSV	(t, F, d1, d2, d3, d4, d5, d6, label)	1 mm	GAN generation

Note: d: displacement of the related floor, label: 0 (healthy) or 1 (damaged).
Abbreviations: F, force; t, time.

Table 6: Statistical descriptors for the main variables of the case study datasets.

Name	Range			Mean			StDev			95% CI			IQR		
	d6	F	t	d6	F	t	d6	F	t	d6	F	t	d6	F	t
data_sens	0–39	0.0–400.1	0–100	13.8	122.4	49.4	7.2	56.6	44.4	0.3	5.2	0.6	10.0	83.7	100
data_test	0–49	0.3–463.8	0–100	20.3	194.0	54.1	12.0	114.1	41.8	0.5	5.2	1.9	20.0	186.9	100
data_stat	0–49	0.0–494.7	0–100	19.7	199.8	50.0	11.5	116.0	33.2	0.0	0.2	0.1	19.6	199.5	97
data_GAN	0–49	0.0–489.2	0–100	15.9	164.5	55.1	12.6	129.7	33.4	0.1	0.6	0.1	22.2	232.9	91

Note: d6: 6th floor displacement (mm), F: force (N), t: time of damage (years).
Abbreviations: CI, confidence interval; IQR, interquartile range; StDev, standard deviation.

Table 7: Architecture of the CNN used as damage diagnostic model.

Layer	Output shape	Parameters
Input (d1, d2, d3, d4, d5, d6)	(6, 1)	0
CNN 1D (ReLU, 32, 3)	(4, 32)	128
CNN 1D (ReLU, 32, 3)	(2, 32)	3104
CNN 1D (ReLU, 32, 2)	(1, 32)	2080
Flatten ()	(32)	0
Dense (ReLU, 64)	(64)	2012
Dropout (0.3)	(64)	0
Dense (Sigmoid, 1)	Output (label 0,1) (1)	65
Total parameters	—	7489

Table 8: Comparative results of the diagnostic model trained on different synthetic datasets (*data_stat* and *data_GAN*).

Model	Dataset for the training	Metrics					
		Train			Validation		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
Diagnostic (binary classifier)	*data_stat*	0.987	0.975	0.998	0.981	0.973	0.996
	data_GAN	0.938	0.956	0.921	0.933	0.908	0.916

predicting the progression of its damage once detected. The regressor specifically forecasts the time during which the structure has suffered damage since it began, taking into account the applied force and a displacement vector measured on the six floors of the structure.

The RUL is calculated using one hundred years as the reference time required to fully develop the damage state and, therefore, as the maximum available structural life. Consequently, the following equation (13) is played to obtain the RUL, where t_i is the predicted time of damage at a discrete time i, expressed in years:

$$\mathrm{RUL}\,(\mathrm{years}) = 100 - t_i. \tag{13}$$

Similarly to the damage diagnostics model introduced previously, the prognostic model has been trained separately on both datasets: *data_stat* and *data_GAN*, saving 20% of the data in each for validation. The architecture of the CNN prognostic model is presented in Table 9.

The learning curves of the prognostic models, depicted in Figure 15, exhibit different patterns. Notably, when utilising data_GAN (Figure 15(b)), the model displays a need for more training epochs due to the increased diversity of the dataset.

At the same time, the learning process during training and validation differs between the two datasets, as illustrated in Figure 16. Learning with *data_stat* (Figure 16(a)) demonstrates a more linear approach, while data_GAN (Figure 16(b)) explores a wider range of potential solutions.

The performance of the prognostic model has been evaluated using metrics such as the mean squared error (MSE) to measure the average squared difference between the estimated values and the true values; its variant using absolute values: the MAE; and the R^2 score or coefficient of determination, which calculates the proportion of variance in the dependent variable that is explained by the independent variables in the regression model, which scores from 0 to 1 (the greater, the better).

It is worth noting that the task of forecasting the damage is not straightforward, as the model initially has to identify the specific floor that is affected by the damage and subsequently predict the corresponding time of damage for that particular floor and no other, which could mislead the result. Due to this fact, the errors are significant, as can be seen in Table 10. Nevertheless, the performance of the prognostic model is good when trained on *data_stat*, and good enough when the model is trained on *data_GAN*.

4.3. Testing the Models on Real Data. To assess the models' functionality within a more realistic setting, their performance is evaluated using novel real data obtained from sensors that were not previously encountered by the models. The testing phase often reveals common errors made by diagnostic and prognostic models trained with poor-quality data, such as inaccurate classifications, resulting in FPs and

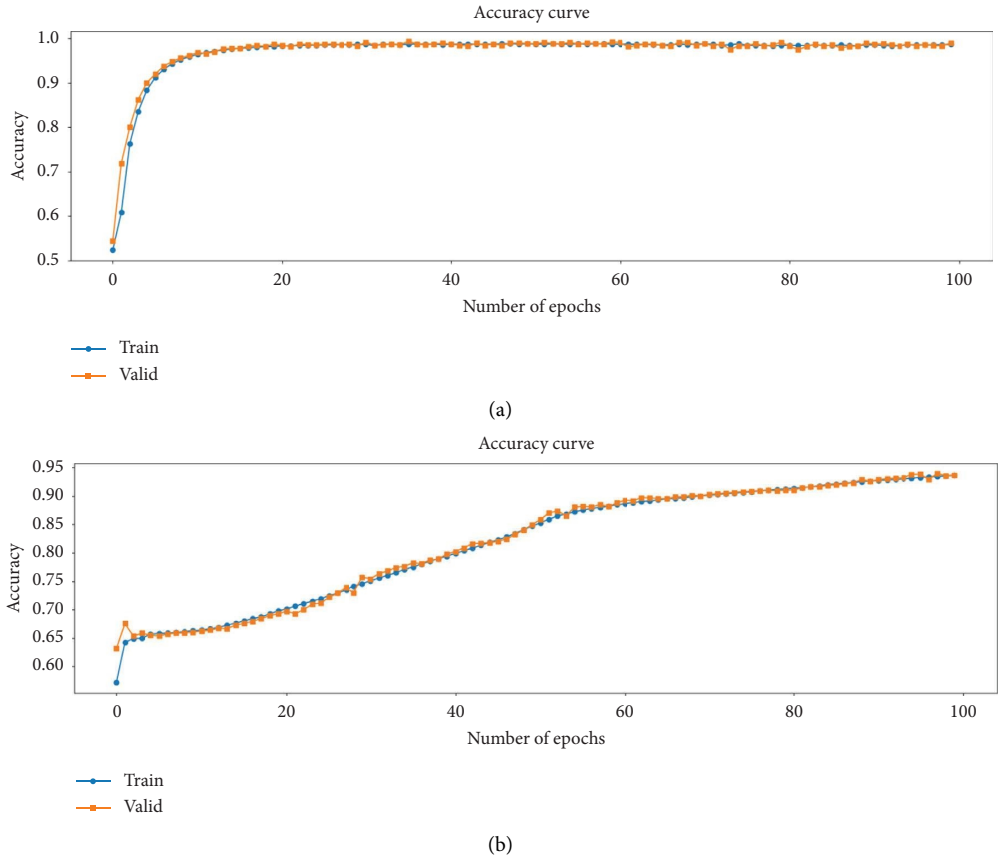


FIGURE 13: Learning curves of the classifier model trained on (a) *data_stat* and (b) *data_GAN*.

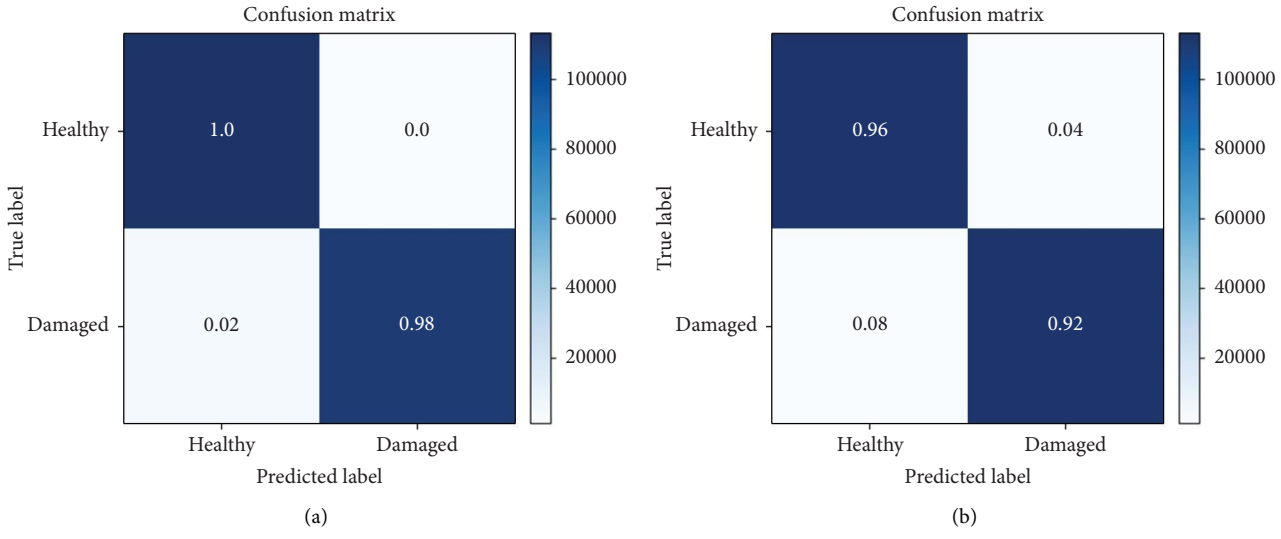


FIGURE 14: Normalised confusion matrices for the classifier model trained on (a) *data_stat* and (b) *data_GAN*.

FNs. These errors occur when the models incorrectly predict nonexistent issues or fail to detect actual problems. Additionally, models often misclassify rare events due to imbalanced training data, leading to a bias towards more common outcomes. Overgeneralisation is another significant issue, as models trained on limited or homogeneous data tend to perform poorly on diverse or unseen data. Furthermore, if the training data are biased, the models can exhibit a bias towards specific features, causing systematic errors and misinterpretations of patterns. Addressing these challenges requires robust data generation techniques to improve the models' generalisation and accuracy. This

Table 9: Architecture of the prognostic model.

Layer	Output shape	Parameters
Input (F, d1, d2, d3, d4, d5, d6)	(7, 1)	0
CNN 1D (ReLU, 32, 3)	(4, 32)	128
CNN 1D (ReLU, 32, 3)	(2, 32)	3104
CNN 1D (ReLU, 32, 2)	(1, 32)	2080
Flatten ()	(32)	0
Dense (ReLU, 64)	(64)	4160
Dense (ReLU, 32)	(32)	2080
Dense (1)	Output (t) (1)	33
Total parameters	—	11,585

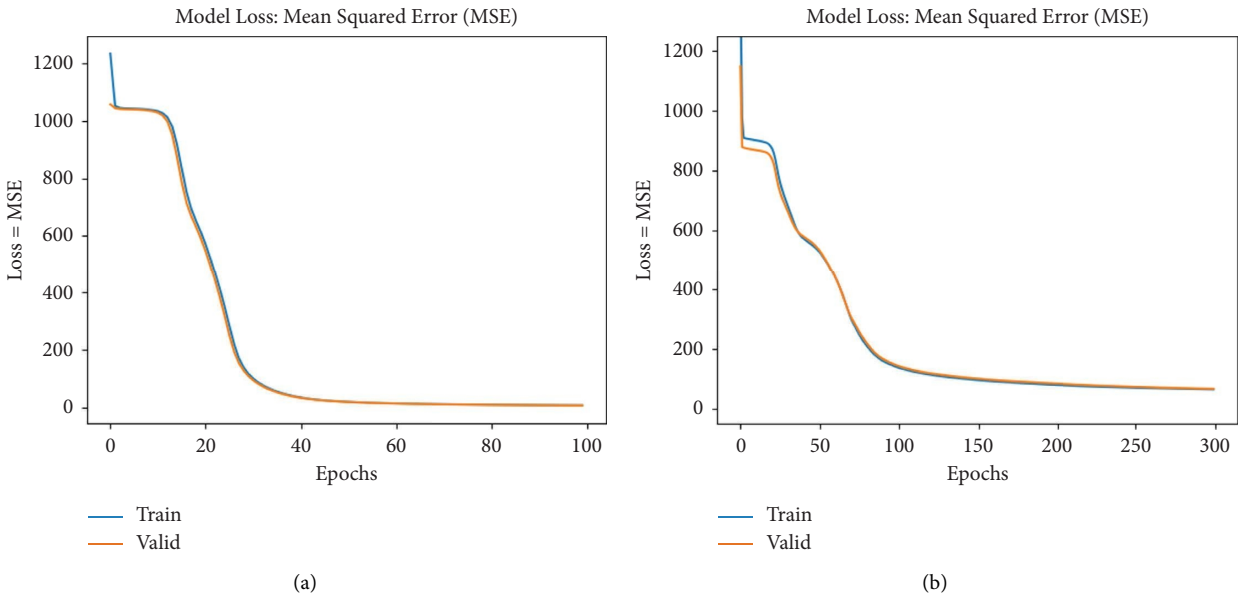


Figure 15: Learning curves of the regression model trained on (a) *data_stat* and (b) *data_GAN*.

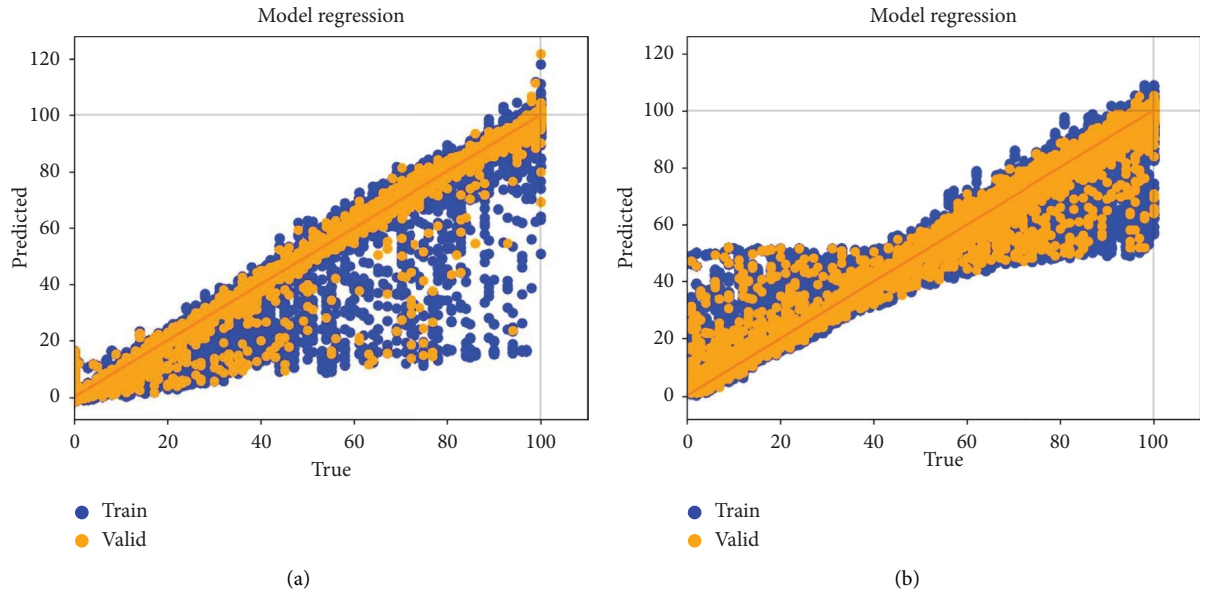


Figure 16: Predicted values versus true values during training with (a) *data_stat* and (b) *data_GAN*.

Table 10: Comparison of the prognostic model performance trained on different synthetic datasets (*data_stat* and *data_GAN*).

Model	Dataset for the training	Metrics					
		Train			Validation		
		MSE	MAE	R^2	MSE	MAE	R^2
Prognostic (regressor)	data_stat	5.080	0.850	0.995	5.371	0.851	0.994
	data_GAN	91.898	5.106	0.909	93.876	5.109	0.911

procedure enables the monitoring of the GAN-generated data as a training dataset to instruct the models [64].

The new data for testing, which comprise more than 1000 samples, were collected in the aforementioned three damage scenarios: healthy (stiffness coefficient $[\alpha = 1]$), medium damage (stiffness coefficient $[\alpha = 0.6]$) and fully damaged (stiffness coefficient $[\alpha = 0, 4]$). Damage was inflicted independently on each of the six stories, and samples were labelled '0' for the healthy state and '1' for the damaged scenarios. Measurements included both force and displacement, as illustrated in Figure 17, where the displacement at the sixth floor is represented.

During the testing phase, the diagnostic and prognostic models were comparatively evaluated using the new dataset, referred to as *data_test*, and the synthetically generated dataset (*data_GAN*), as shown in Table 11. The results clearly demonstrate that the models trained with *data_GAN* outperform the models trained without it. This indicates that the GAN-generated dataset enhances the model's generalisation capability, effectively capturing the noise present in real data and enabling accurate diagnosis and prognostic predictions.

Regarding diagnosis, the performance of the models with the new dataset *data_test* has been similar on average in the confusion matrices (Figure 18), with slightly better accuracy of the model trained on *data_GAN* (accuracy = 0.92) than the model trained on *data_stat*, as seen in Table 11. However, the most important aspect is that the recall is the highest in the case of the model trained on *data_GAN* concerning the security of the tower, as it minimises to 0 the existence of FNs (being damaged but predicted as healthy).

To graphically deploy the accuracy of a binary classifier, the area under the [65] receiver operating characteristic (ROC) curve (AUC) is used. A higher AUC and a ROC curve closer to the top left corner indicate better performance [66], as happens with the model trained using *data_GAN* (Figure 19(b)).

In the case of the prognostic regressor, the results show accurate values for the predicted median and quartiles for the three scenarios tested ($t = 0$ for the healthy state, $t = 66$ for the medium damaged and $t = 100$ for the fully damaged), with better performance for the case of the models trained on data_GAN as seen in Figure 20 and Table 12. In this case, the interquartile range exhibited a wider span. However, the predicted median showed greater proximity to the actual value and, importantly, no outliers were observed.

According to the results obtained, it can be concluded that the quality of the synthetic data employed to train and validate the models directly impacts their ability to generalise to unseen real-world scenarios. It confers enhanced robustness to variability, enabling an accurate damage diagnosis and prognosis. It is also appreciated that the number

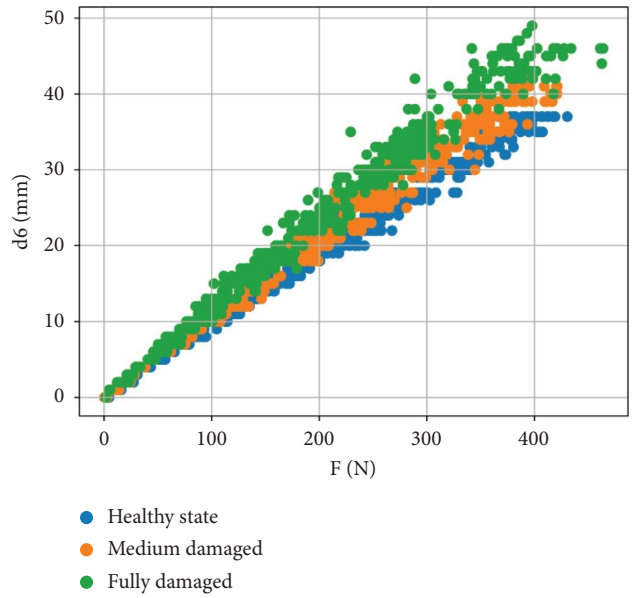


Figure 17: Real data never seen before by the models, obtained in a test and included in data_test.

Table 11: Comparison of the models' performance trained on different synthetic datasets (*data_stat* and *data_GAN*) and tested on the new real data (data_test).

Dataset for the training	Metrics					
	Diagnosis			Prognostic		
	ACC	REC	PREC	MSE	MAE	R^2
data_stat	0.91	0.96	0.81	273.548	12.046	0.843
data_GAN	0.92	1.00	0.81	264.601	11.746	0.871

Note: R^2: Coef. of determination.
Abbreviations: ACC, accuracy; MAE, mean absolute error; MSE, mean squared error; PREC, precision; REC, recall.

of resulting FNs (predicted to be healthy when damaged) is minimal, reinforcing the safety of the proposed method.

The experiments and analyses were conducted on a computer with the following specifications: an Intel Core i7-8700K processor, Intel Dual Band Wireless-AC 8265 adapter, 16 GB DDR4 RAM, 512 GB SSD and running on Windows 10 Pro. This setup ensured sufficient performance for the computational tasks involved in this research.

The efficiency of the data generation process was evaluated by measuring the time required to generate synthetic datasets of varying sizes by performing both methods. The generation times were recorded to assess scalability. Similarly, the efficiency of the diagnostic and prognostic model's training and testing was assessed by recording the training

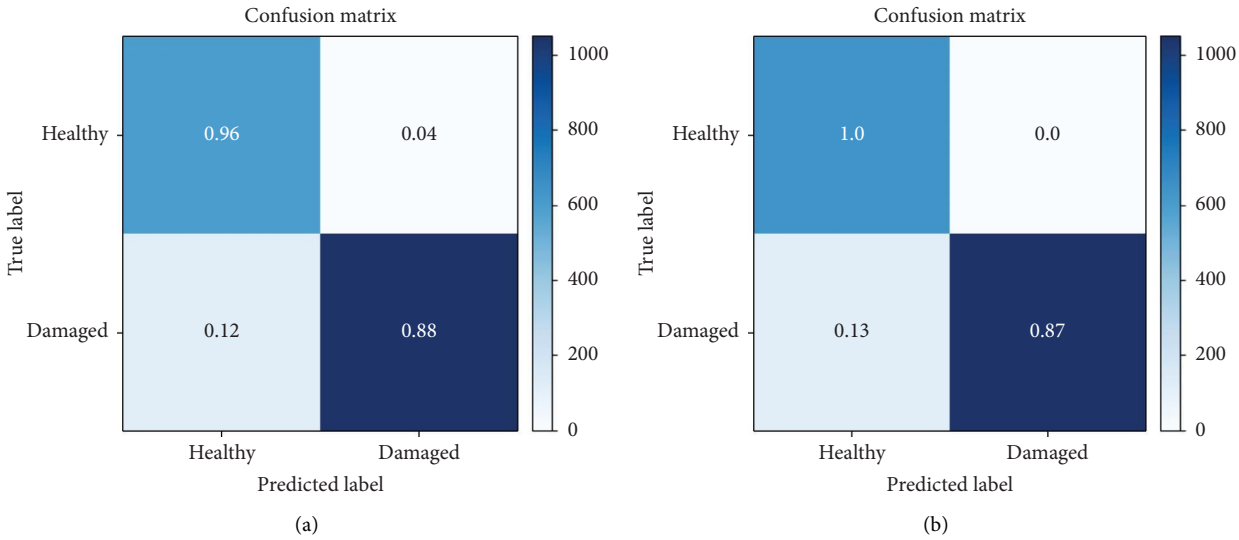


FIGURE 18: Normalised confusion matrices for the classifier model trained on (a) *data_stat* and (b) *data_gan* performing on the new dataset data_test.

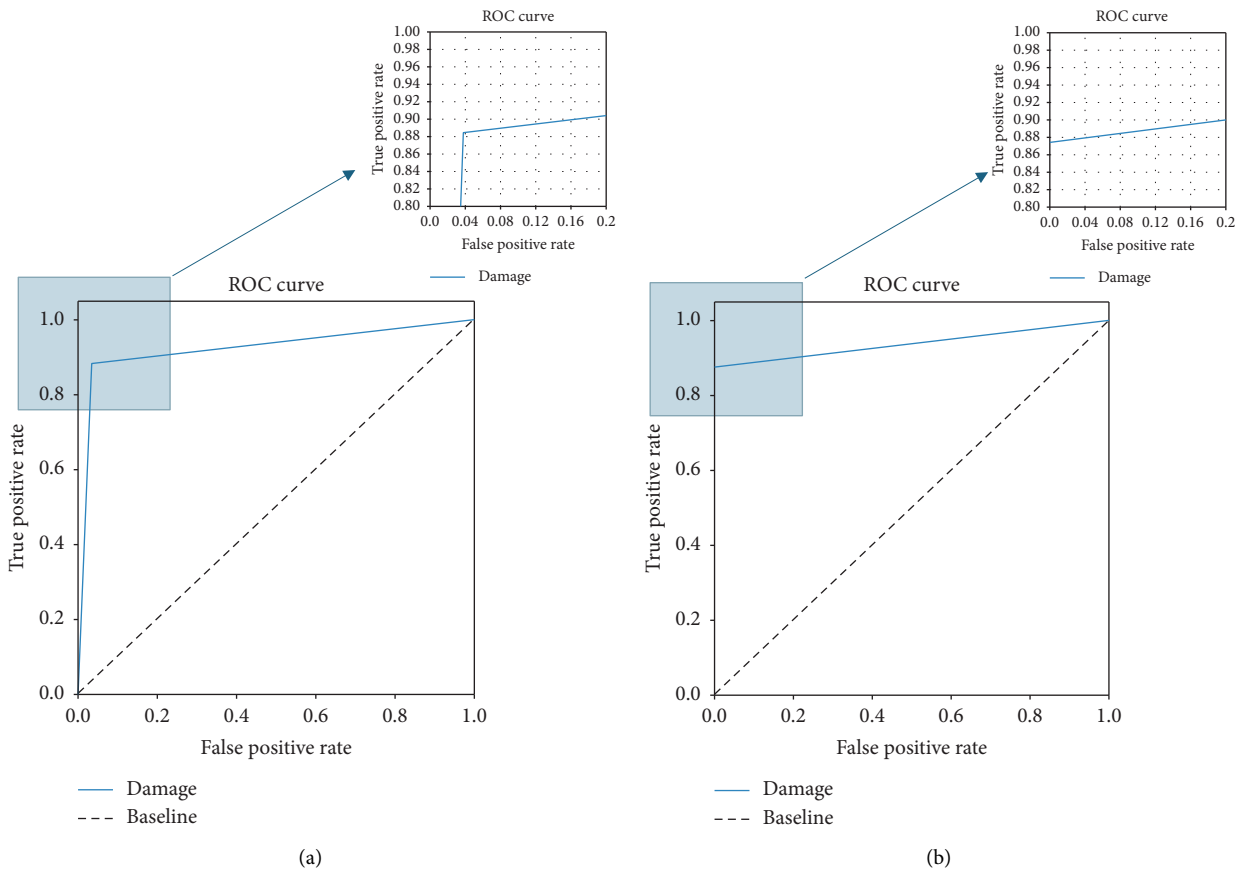


FIGURE 19: ROC curves for the damage classifier model trained on (a) *data_stat* and (b) *data_GAN*, performing on the new dataset data_test.

epochs and the time consumed. The results included in Table 13 indicated that while the training time per epoch increased with more complex datasets, such as the WGAN-GP CNN-based, models maintained reasonable training durations, illustrating their suitability for scalable applications.

Table 12: Comparison of predicted value distribution obtained from the models trained on different synthetic datasets (*data_stat* and *data_GAN*) and tested on the real data (*data_test*).

Dataset for the training	Properties of the predicted value distributions							
	t (years)	Median	Q1	Q3	IQR	MIN	MAX	OUT
data_stat	0	16.998	12.796	25.334	12.565	12.768	39.850	7
	66	60.369	52.137	65.475	13.338	40.045	69.938	0
	100	95.716	86.804	97.919	11.115	70.001	105.359	5
data_GAN	0	12.453	4.458	27.684	23.226	−2.838	42.813	0
	66	67.749	56.841	78.809	21.968	45.051	89.942	0
	100	94.748	92.894	101.481	8.587	90.034	109.225	0

Note: Q1: percentile 25%, Q3: percentile 75%.
Abbreviations: IQR, interquartile range; MAX, maximum value; MIN, minimum value; OUT, outliers.

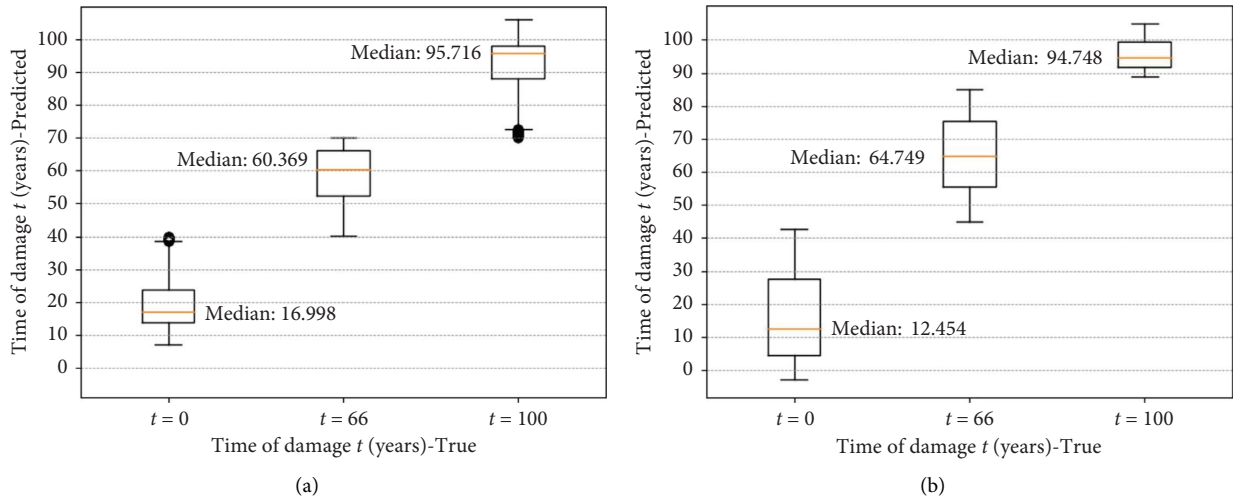


Figure 20: Time of damage boxplots predicted by the prognostic models: (a) model trained on *data_stat* and (b) model trained on *data_GAN*.

5. Application Within the DT Context

This section shows how CNN-based diagnostics and prognostics models trained on synthetically GAN-generated datasets can be efficiently used within the workflow of a DT to enable decision-making. The models, which are trained offline and utilised online, are updated when new state parameters, such as changes in stiffness or mass, appear in the system. Updates can occur either online or offline, based on a predefined interval or when specific threshold values are reached by the system. The workflow can be managed autonomously by a Petri net (PN)–based framework, which represents the logic of discrete events in a dynamically distributed system and effectively handles the workflow of the DT [2].

In Figure 21, a PN designed for the presented case study is illustrated, consisting of eight places (p_1 to p_8) representing discrete system states, seven transitions (t_1 to t_7) denoting symbolic or conditional actions (including post-firing actions) and two cold transitions ϵ for data arrival and system rearm. Changes in the DT state are the outcomes of automated actions triggered by firing transitions t_1 to t_7.

The system is assumed to initiate at time $T = 0$ when new data are received from the force and displacement sensors. At this point, the virtual twin is updated with the physical twin, as indicated by one token in both p_1 and p_4. Subsequently, firing transition t_4 occurs, placing one token in p_5 and removing it from p_4. This signifies that the structure is subjected to a new force, and a decision must be made regarding the potential consequences for the system, including the update of the virtual twin. This is accomplished through transitions t_1 and t_5, each based on their respective transition conditions (Table 14).

When t_5 is activated indicating a healthy state, a token is produced for p_4, which means that the DT does not require an update of the virtual twin with respect to the real one, and thus, the DT keeps its previous 'updated' state. However, if the diagnostic model reveals a damaged state, the workflow sequence p_2, t_2, p_3, t_3 will occur by placing one token in p_4 and causing the system to return to an 'updated' state.

While the DT remains in the 'updated' state, marked by the presence of a token in p_4, an evaluation is conducted on the measured force value to determine whether it exceeds a safety threshold. When the force exceeds this threshold, transition t_6 is triggered. This transition activates a sequence of warning states and actions represented by nodes p_7, t_7, p_8, which autonomously indicate that the structure is exposed to a force that may compromise its integrity. In such cases, the

Table 13: Comparative results about computational efficiency.

Method	Data generation					Diagnostic model (classifier)			Prognostic model (regressor)		
	Dataset generated		Model's training time (h)	Data generation time (h)	# epochs	Training		Testing	Training		Testing
	Name	Size (instances)				# epochs	Time (h)	Time (h)	# epochs	Time (h)	Time (h)
Classical method (FEM + statistical noise)	*data_stat*	600,000	—	46.32	—	100	0.08	0.001	100	0.06	0.001
WGAN-GP CNN-based	*data_GAN*	150,000	1.47	0.002	20*	100	0.21	0.002	300	0.18	0.002

Abbreviation: h = hours.

*Every generator epoch implies 5 times more epochs in the critic as designed by [43].

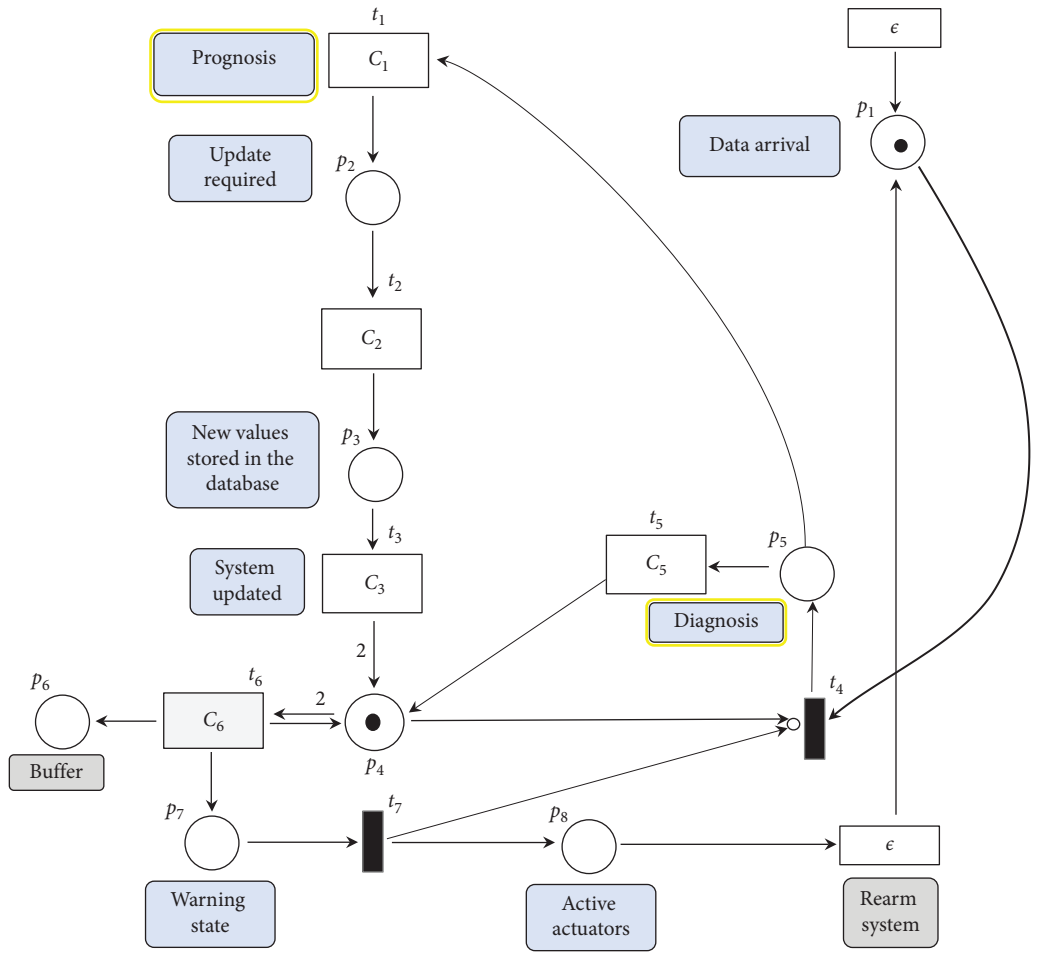


Figure 21: PN workflow of a DT assisted by the surrogate models of diagnostic and prognostic (highlighted).

Table 14: Transitions in the PN-assisted workflow of the DT.

Transition	Type	Description
t_1	Conditional ($\mathscr{C}_1$)	If diagnostic label equals 1 (damage), execute prognostic
t_2	Conditional ($\mathscr{C}_2$)	When prognostic executed, save updated values in the database
t_3	Conditional ($\mathscr{C}_3$)	When prognostic executed, update models with the new parameter's values
t_4	Conditional ($\mathscr{C}_4$)	When data arrive, continue the flow
t_5	Conditional ($\mathscr{C}_5$)	Perform diagnosis and obtain the diagnostic label: '0' if healthy, '1' if damaged
t_6	Conditional ($\mathscr{C}_6$)	Evaluates if the measured force exceeds threshold force
t_7	Conditional ($\mathscr{C}_7$)	Triggers the actuator

actuators trigger an alarm (through the firing of t_7), and the system enters a 'warning state' that demands predictive maintenance based on the calculated RUL, or alternatively, corrective action if the measured force has exceeded the threshold. At this stage, the system is rearmed and awaits new data, represented by the cold transition ϵ, which dismisses the warning state until a new evaluation is conducted. Importantly, when the warning sequence is activated, a token is collected in p_6, serving as an information buffer that can be utilised for diagnostic purposes. This buffer provides insight into the number of instances when the structure has experienced force values surpassing the integrity threshold.

Remarkably, this workflow presents the capability to operate in real time thanks to the surrogate models of diagnosis and prognosis. These models were effectively trained with GAN-generated synthetic datasets, which conferred valuable properties of interpreting raw data directly from sensor measurements and demonstrated generalisation ability to make accurate predictions.

6. Conclusions

In this study, a new framework has been developed to overcome the data scarcity problem that affects the implementation of DT in the AECO field. The contribution consists of a comprehensive procedure for data generation, model training and testing for 1-D discrete signals coming from an IoT-based SHM system. This entitles the DT to

process raw data at the edge, minimise resource consumption and enable real-time decision-making through the use of surrogate models. These models possess a deep understanding of a wide range of possible scenarios, as they were trained on synthetic datasets generated by the proposed procedure.

To assess the efficacy of the synthetic dataset generation, classification (diagnosis) and regression (prognostic) models were trained and evaluated using a laboratory-scale structure as a case study. The results obtained from the GAN-generated dataset were compared to a dataset created using traditional statistical techniques (e.g., FE method results with added heteroscedastic Gaussian noise), demonstrating superior performance in the former case. Subsequently, the trained models were integrated into the DT, thereby enhancing its decision-making capabilities in real-time scenarios.

Future research endeavours should focus on expanding the capabilities of the models to precisely localise damage within the system and assess its severity. This entails training a multiclass classifier, increasing the number of scenarios in the synthetic GAN-generated dataset and designing a strategy to discretise the severity levels of the experienced damage. These steps are crucial for advancing the operations and maintenance strategy of the DT. Upcoming efforts will also leverage the inherent scalability of GAN-GP CNN-based methods to tackle more extensive systems and complex datasets.

The present research has the potential for a broader applicability across other sectors beyond AECO. One of these sectors is the energy industry, where discrete measurements of environmental parameters and energy values can be effectively monitored and modelled. The scheme is also adaptable to transportation and traffic flows, encompassing not only vehicular traffic but also dynamic flows such as pipelines, and similarly, industrial processes and supply chains, to cite any. These sectors are suitable for IoT discrete sensor monitoring IoT that captures the relevant features essential for operational and maintenance deployments. These constitute the foundational data for the generative and behavioural models that can be seamlessly integrated into the DTs of their respective systems, as expounded in this study.

In conclusion, this study leveraged a limited collection of discrete data as a foundation for the generation of an enhanced synthetic dataset representing the observed system, upon which diagnostic and prognostic models are built and refined. The methodological innovation laid in a comprehensive approach that fully covers from the generation of data mimicking the real system to the training and testing of the models. This provides distinct advantages in terms of real-time computing on the edge for immediate decision-making and minimisation of energy consumption. The present work has been developed using state-of-the-art AI, specifically employing architectures made of advanced DL models such as WGAN-GP and CNNs.

Moving forward, further experimentation should be conducted to assess the real-time performance and generalisation capabilities of the presented framework in benchmark real-world AECO scenarios and broader applications. This will involve rigorous testing and comparison with other state-of-the-art deep NN-based approaches to further gauge the robustness of this methodology.

The impact of this research extends beyond the experimental setting, offering valuable insight into other sectors in which main features can be monitored and modelled to be seamlessly integrated into a DT implementation. By addressing the data scarcity issue along with other challenges such as cybersecurity compliance or the reduction of costs of communications and data storage promoting sustainability, this work contributes to the advancement of the DT application in the AECO sector and laid the groundwork for broader applications in other fields, such as energy or manufacturing industry.

To end with, it is paramount to underscore the critical importance of safeguarding the security of data generation processes, maintaining data integrity to prevent tampering and ensuring fairness by mitigating bias. Furthermore, robust measures must be implemented to protect data against cyber threats, and transparent usage policies along with compliance with legal standards are imperative. Additionally, proactive steps are essential to prevent potential misuse and misinterpretation of synthetic data through the provision of comprehensive documentation and guidelines. These concerted efforts are crucial for optimising the advantages of synthetic data while upholding the highest standards of security and ethical conduct.

Data Availability Statement

The data of this study are available from the corresponding author upon request. A fundamental Python implementation of the models developed in this research is available on GitHub. It can be accessed via the following link: https://github.com/mmmmaria/Digital-twin.git.

Disclosure

The content of this paper is included in a published thesis [67], which is accessible through the following link: https://digibug.ugr.es/handle/10481/92528.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This research has received funding from the following projects: ENHAnCE ITN (https://www.h2020-enhanceitn.eu/) from the European Union Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie grant agreement No. 859957 and the Grant Nos. PID 2020-119478GB-I00, PLEC2021-007798 and PID2020-116644RB-I00 of the Spanish Ministry of Science and Education. The authors appreciate the support of these projects.

References

[1] E. Glaessgen and D. Stargel, "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles," in *Structural Dynamics and Materials Conference* (AIAA 53rd AIAA/ASME/ASCE/AHS/ASC Structures, 2012).

[2] M. Chiachío, M. Megía, J. Chiachío, J. Fernandez, and M. L. Jalón, "Structural Digital Twin Framework: Formulation and Technology Integration," *Automation in Construction* 140 (2022): 104333, https://doi.org/10.1016/j.autcon.2022.104333.

[3] "CO2 Emissions from Buildings and Construction Hit New High, Leaving Sector off Track to Decarbonize by 2050," *UN Environmental Programme* (2022): https://www.unep.org/news-and-stories/press-release/co2-emissions-buildings-and-construction-hit-new-high-leaving-sector.

[4] A. A. Sezer, M. Thunberg, and B. Wernicke, "Digitalization Index: Developing a Model for Assessing the Degree of Digitalization of Construction Projects," *Journal of Construction Engineering and Management* 147, no. 10 (2021): 04021119, https://doi.org/10.1061/(asce)co.1943-7862.0002145.

[5] O. Lundberg, D. Nylén, and J. Sandberg, "Unpacking Construction Site Digitalization: The Role of Incongruence and Inconsistency in Technological Frames," *Construction Management & Economics* 40, no. 11-12 (2022): 987–1002, https://doi.org/10.1080/01446193.2021.1980896.

[6] A. Singh, "The Significance of Digitalization of the Construction Sector," in *International Conference on Intelligent Computing & Optimization* (Springer, 2022), 1069–1077.

[7] A. Corneli, B. Naticchia, A. Carbonari, and M. Vaccarini, "A Framework for Development and Integration of Digital Twins in Construction," in *Ecppm 2021–eWork and eBusiness in Architecture, Engineering and Construction* (2021), 291–298.

[8] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the Digital Twin: A Systematic Literature Review," *CIRP Journal of Manufacturing Science and Technology* 29 (2020): 36–52, https://doi.org/10.1016/j.cirpj.2020.02.002.

[9] T. Goodwin, J. Xu, N. Celik, and C.-H. Chen, "Real-time Digital Twin-Based Optimization With Predictive Simulation Learning," *Journal of Simulation* 18 (2022): 47–64, https://doi.org/10.1080/17477778.2022.2046520.

[10] M. Wang, S. Feng, A. Incecik, G. Królczyk, and Z. Li, "Structural Fatigue Life Prediction Considering Model Uncertainties through a Novel Digital Twin-Driven Approach," *Computer Methods in Applied Mechanics and Engineering* 391 (2022): 114512, https://doi.org/10.1016/j.cma.2021.114512.

[11] S. O. Abioye, L. O. Oyedele, L. Akanbi, et al., "Artificial Intelligence in the Construction Industry: A Review of Present Status, Opportunities and Future Challenges," *Journal of Building Engineering* 44 (2021): 103299, https://doi.org/10.1016/j.jobe.2021.103299.

[12] G. B. Ozturk, "Interoperability in Building Information Modeling for AECO/FM Industry," *Automation in Construction* 113 (2020): 103122, https://doi.org/10.1016/j.autcon.2020.103122.

[13] D. Chakraborty, N. Kovvali, B. Chakraborty, A. Papandreou-Suppappola, and A. Chattopadhyay, "Structural Damage Detection With Insufficient Data Using Transfer Learning Techniques," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, 7981 (2011), 1175–1183.

[14] Y. Du, L.-fang Li, R.-rong Hou, X.-you Wang, W. Tian, and Y. Xia, "Convolutional Neural Network-Based Data Anomaly Detection Considering Class Imbalance with Limited Data," *Smart Structures and Systems* 29.1 (2022): 63–75.

[15] A. Bansal, R. Sharma, and M. Kathuria, "A Systematic Review on Data Scarcity Problem in Deep Learning: Solution and Applications," in *ACM Computing Surveys (CSUR)* (2022), 1–29.

[16] A. Akrim, C. Gogu, R. Vingerhoeds, and M. Salaün, "Self-Supervised Learning for Data Scarcity in a Fatigue Damage Prognostic Problem," *Engineering Applications of Artificial Intelligence* 120 (2023): 105837, https://doi.org/10.1016/j.engappai.2023.105837.

[17] R. Agarwal, S. Chandrasekaran, and M. Sridhar, in *Imagining Construction's Digital Future* (McKinsey & Company, 2020).

[18] F. Luleci, F. N. Catbas, and O. Avci, "Generative Adversarial Networks for Labeled Acceleration Data Augmentation for Structural Damage Detection," *Journal of Civil Structural Health Monitoring* 13, no. 1 (2023): 181–198, https://doi.org/10.1007/s13349-022-00627-8.

[19] S. James, C. Harbron, J. Branson, and M. Sundler, "Synthetic Data Use: Exploring Use Cases to Optimise Data Utility," *Discover Artificial Intelligence* 1, no. 1 (2021): 15, https://doi.org/10.1007/s44163-021-00016-y.

[20] R. J. Chen, M. Y. Lu, T. Y. Chen, D. F. K. Williamson, and F. Mahmood, "Synthetic Data in Machine Learning for Medicine and Healthcare," *Nature Biomedical Engineering* 5, no. 6 (2021): 493–497, https://doi.org/10.1038/s41551-021-00751-8.

[21] A. Sahu, J. Mishra, and N. Kushwaha, "Artificial Intelligence (AI) in Drugs and Pharmaceuticals," *Combinatorial Chemistry & High Throughput Screening* 25, no. 11 (2022): 1818–1837, https://doi.org/10.2174/1386207325666211207153943.

[22] S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, and M. Veloso, "Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls," in *Proceedings of the First ACM International Conference on AI in Finance* (2020), 1–8.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative Adversarial Networks," *Communications of the ACM* 63, no. 11 (2020): 139–144, https://doi.org/10.1145/3422622.

[24] F. Luleci, F. N. Catbas, and O. Avci, "A Literature Review: Generative Adversarial Networks for Civil Structural Health Monitoring," *Frontiers in Built Environment* 8 (2022): 1027379, https://doi.org/10.3389/fbuil.2022.1027379.

[25] G. Tsialiamanis, D. J. Wagg, N. Dervilis, and K. Worden, "On Generative Models as the Basis for Digital Twins," *Data-Centric Engineering* 2 (2021): e11, https://doi.org/10.1017/dce.2021.13.

[26] H. Lee, J. Kim, E. K. Kim, and S. Kim, "Wasserstein Generative Adversarial Networks Based Data Augmentation for Radar Data Analysis," *Applied Sciences* 10, no. 4 (2020): 1449, https://doi.org/10.3390/app10041449.

[27] Y. Zhuang, J. Qin, B. Chen, C. Dong, C. Xue, and S. M. Easa, "Data Loss Reconstruction Method for a Bridge Weigh-In-Motion System Using Generative Adversarial Networks," *Sensors* 22, no. 3 (2022): 858, https://doi.org/10.3390/s22030858.

[28] M. G. Kapteyn and K. E. Willcox, "Predictive Digital Twins: Where Dynamic Data-Driven Learning Meets Physics-Based Modeling," in *Dynamic Data Driven Applications Systems: Third International Conference, DDDAS 2020* (Boston, MA, October 2020), 3–7.

[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature* 521, no. 7553 (2015): 436–444, https://doi.org/10.1038/nature14539.

[30] K. Dunphy, M. N. Fekri, K. Grolinger, and A. Sadhu, "Data Augmentation for Deep-Learning-Based Multiclass Structural

Damage Detection Using Limited Information," *Sensors* 22, no. 16 (2022): 6193, https://doi.org/10.3390/s22166193.

[31] 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* 20, no. 10 (2020): 2778, https://doi.org/10.3390/s20102778.

[32] Yi-C. Zhu, D. Wagg, E. Cross, and R. Barthorpe, "Real-time Digital Twin Updating Strategy Based on Structural Health Monitoring Systems," in *Model Validation and Uncertainty Quantification, Volume 3: Proceedings of the 38th IMAC, A Conference and Exposition on Structural Dynamics 2020* (Springer, 2020), 55–64.

[33] L. Xiang, H. Liu, and C. Serratella, "Application of Structural Health Monitoring for Structural Digital Twin," in *Offshore Technology Conference Asia* (OnePetro, 2020).

[34] Y. Gao, H. Li, G. Xiong, and H. Song, "AIoT-Informed Digital Twin Communication for Bridge Maintenance," *Automation in Construction* 150 (2023): 104835, https://doi.org/10.1016/j.autcon.2023.104835.

[35] H. Salehi, S. Das, S. Biswas, and R. Burgueño, "Data Mining Methodology Employing Artificial Intelligence and a Probabilistic Approach for Energy-Efficient Structural Health Monitoring With Noisy and Delayed Signals," *Expert Systems with Applications* 135 (2019): 259–272, https://doi.org/10.1016/j.eswa.2019.05.051.

[36] C. R. Farrar and K. Worden, *Structural Health Monitoring: A Machine Learning Perspective* (John Wiley & Sons, 2012).

[37] F. Luleci and F. N. Catbas, "A Brief Introductory Review to Deep Generative Models for Civil Structural Health Monitoring," *AI in Civil Engineering* 2 (2023): 9–11, https://doi.org/10.1007/s43503-023-00017-z.

[38] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep Generative Modelling: A Comparative Review of Vaes, Gans, Normalizing Flows, Energy-Based and Autoregressive Models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[39] A. Jabbar, Xi Li, and B. Omar, "A Survey on Generative Adversarial Networks: Variants, Applications, and Training," *ACM Computing Surveys* 54, no. 8 (2021): 1–49, https://doi.org/10.1145/3463475.

[40] P.-H. Lu, P.-C. Wang, and C.-Mu Yu, "Empirical Evaluation on Synthetic Data Generation with Generative Adversarial Network," in *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics* (2019), 1–6.

[41] A. Antoniou, A. Storkey, and H. Edwards, "Data Augmentation Generative Adversarial Networks," (2017), https://arxiv.org/abs/1711.04340.

[42] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *International Conference on Machine Learning* (PMLR, 2017), 214–223.

[43] G. Ishaan, F. Ahmed, M. Arjovsky, D. Vincent, and A. C. Courville, "Improved Training of Wasserstein Gans," in *Advances in Neural Information Processing Systems*, 30 (2017).

[44] R. F. P. Toby, C. E. Heaney, E. Benmoufok, et al., "Multi-Output Regression With Generative Adversarial Networks (MOR-GANs)," *Applied Sciences* 12.18 (2022): 9209.

[45] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent Progress on Generative Adversarial Networks (GANs): A Survey," *IEEE Access* 7 (2019): 36322–36333, https://doi.org/10.1109/access.2019.2905015.

[46] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals," *Sensors* 17, no. 2 (2017): 425, https://doi.org/10.3390/s17020425.

[47] W. Lu, B. Liang, Yu Cheng, D. Meng, J. Yang, and T. Zhang, "Deep Model Based Domain Adaptation for Fault Diagnosis," *IEEE Transactions on Industrial Electronics* 64, no. 3 (2017): 2296–2305, https://doi.org/10.1109/tie.2016.2627020.

[48] Y. LeCun, B. Boser, J. S. Denker, et al., "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation* 1, no. 4 (1989): 541–551, https://doi.org/10.1162/neco.1989.1.4.541.

[49] C. Jiang, Q. Zhou, J. Lei, and X. Wang, "A Two-Stage Structural Damage Detection Method Based on 1D-CNN and SVM," *Applied Sciences* 12, no. 20 (2022): 10394, https://doi.org/10.3390/app122010394.

[50] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D Convolutional Neural Networks and Applications: A Survey," *Mechanical Systems and Signal Processing* 151 (2021): 107398, https://doi.org/10.1016/j.ymssp.2020.107398.

[51] K. Yang, Z. Huang, X. Wang, and X. Li, "A Blind Spectrum Sensing Method Based on Deep Learning," *Sensors* 19, no. 10 (2019): 2270, https://doi.org/10.3390/s19102270.

[52] H. Mizutani, "The Back Propagation Method for CNN," in *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, 6 (IEEE, 1994), 463–466, https://doi.org/10.1109/iscas.1994.409626.

[53] G. Van Rossum and F. L. Drake, in *Python 3 Reference Manual* (Scotts Valley, CA: CreateSpace, 2009).

[54] M. Zhu, F. McKenna, and M. H. Scott, "OpenSeesPy: Python Library for the OpenSees Finite Element Framework," *SoftwareX* 7 (2018): 6–11, https://doi.org/10.1016/j.softx.2017.10.009.

[55] H. Shi, K. Worden, and E. J. Cross, "A Cointegration Approach for Heteroscedastic Data Based on a Time Series Decomposition: an Application to Structural Health Monitoring," *Mechanical Systems and Signal Processing* 120 (2019): 16–31, https://doi.org/10.1016/j.ymssp.2018.09.036.

[56] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," in *Advances in Neural Information Processing Systems*, 30 (2017).

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing* 13, no. 4 (2004): 600–612, https://doi.org/10.1109/tip.2003.819861.

[58] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE* 86, no. 11 (1998): 2278–2324, https://doi.org/10.1109/5.726791.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research* 15, no. 1 (2014): 1929–1958.

[60] C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, 1995).

[61] D. Specht, "A General Regression Neural Network," *IEEE Transactions on Neural Networks* 2, no. 6 (1991): 568–576, https://doi.org/10.1109/72.97934.

[62] T. J. Gibbons, G. Pierce, K. Worden, and I. Antoniadou, "Convolutional Neural Networks for the Detection of Damaged Fasteners in Engineering Structures," in *Proceedings of the 9th European Workshop on Structural Health Monitoring (EWSHM 2019)* (NDT. net, 2018).

[63] D. Bowes, T. Hall, and D. Gray, "Comparing the Performance of Fault Prediction Models Which Report Multiple Performance Measures: Recomputing the Confusion Matrix," in

Proceedings of the 8th International Conference on Predictive Models in Software Engineering (2012), 109–118.

[64] 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* 21, no. 4 (2022): 1906–1955, https://doi.org/10.1177/14759217211036880.

[65] L. Omar and I. Ivrissimtzis, "Using Theoretical ROC Curves for Analysing Machine Learning Binary Classifiers," *Pattern Recognition Letters* 128 (2019): 447–451, https://doi.org/10.1016/j.patrec.2019.10.004.

[66] M. Gönen, "Receiver Operating Characteristic (ROC) Curves," *SAS Users Group International (SUGI)* 31 (2006): 210–231.

[67] María Megía Cardeñoso, "Digital Twins in Civil Engineering: Conceptual Framework and Real-World Implementations" (https://digibug.ugr.es/handle/10481/92528PhD thesis, Spain: University of Granada, 2024).