Digital Twins in Civil Engineering: Conceptual Framework and Real-World Implementations

TESIS DOCTORAL

Autor:

María Megía Cardeñoso

PROGRAMA DE DOCTORADO EN INGENIERÍA CIVIL

Línea de investigación: Evaluación No Destructiva de Materiales y Estructuras

Directores:

Dr. Manuel Chiachío Ruano, E.T.S. Ingeniería de Caminos, Canales y Puertos

Dr. F. Javier Melero Rus, E.T.S. Ingenierías Informática y Telecomunicación

Departamento de Mecánica de Estructuras e Ingeniería Hidráulica

Universidad de Granada (Granada, España)

Marzo 2024

Editor: Universidad de Granada. Tesis Doctorales
Autor: María Megía Cardeñoso
ISBN: 978-84-1195-330-6
URI: https://hdl.handle.net/10481/92528

List of Figures

List of Tables

List of Algorithms

Contents

La doctoranda María Megía Cardeñoso y los directores de la tesis Dr. Manuel Chiachío Ruano y Dr. Francisco Javier Melero Rus, garantizamos, al firmar esta tesis doctoral, que el trabajo realizado es original y ha sido desarrollado por la doctoranda bajo la dirección de los citados directores de la tesis. Hasta donde nuestro conocimiento alcanza, toda la información proveniente de fuentes externas ha sido debidamente citada y referenciada según las normas académicas establecidas. Además, se garantiza que este trabajo no ha sido presentado anteriormente como parte de ninguna otra evaluación académica y que no infringe los derechos de propiedad intelectual de terceros.

Firmado:

Dtor. de la tesis, Dr. Manuel Chiachío Ruano

Dtor. de la tesis, Dr. Francisco Javier Melero Rus

Doctoranda, María Megía Cardeñoso

Resumen

La ingeniería civil se encuentra en una coyuntura crítica del siglo XXI, enfrentando una confluencia de desafíos y oportunidades que exigen un cambio de paradigma en prácticas y metodologías. A medida que la mayoría de las estructuras construidas a principios del siglo pasado se acercan globalmente a la culminación de su vida útil de diseño, la necesidad de soluciones sostenibles, resilientes y tecnológicamente avanzadas se vuelve primordial. Frente a este reto, la presente tesis doctoral elabora un marco conceptual integral de gemelo digital diseñado para la ingeniería civil, teniendo en cuenta el envejecimiento de las infraestructuras, la digitalización, el impacto ambiental y el imperativo de minimizar los residuos.

Con numerosas construcciones acercándose al final de sus ciclos de vida estimados, el desafío radica no solo en preservar la integridad de estas estructuras sino en reprogramarlas para un futuro sostenible. Para ello, el presente estudio tiene como objetivo investigar estrategias de operación y mantenimiento, modernización, y políticas sostenibles.

En cuanto a la digitalización, si bien se han logrado avances en la adopción de herramientas digitales para el diseño, construcción y gestión de proyectos, persiste un panorama tecnológico fragmentado. Los esfuerzos aislados limitan el potencial transformador de las soluciones digitales. Esta investigación tiene como objetivo proporcionar caminos hacia la implementación de manera cohesiva y colaborativa de las tecnologías bajo el paradigma del gemelo digital.

En la búsqueda de eficiencia y calidad, la ingeniería civil está siendo testigo de una ola de innovación impulsada por las tecnologías emergentes. El Modelado de Información de Construcción (BIM, *Buiding Information Modelling*), el Internet de las Cosas (IoT, *Internet of Things*) y la Inteligencia Artificial (IA) están dando forma al panorama tecnológico en la industria. El presente trabajo evalúa críticamente la adopción e impacto de estas tecnologías, valorando su potencial para revolucionar la práctica de la ingeniería civil en el sector de Arquitectura, Ingeniería, Construcción, y Operaciones y Mantenimiento (AECO, *Architecture, Engineering, Construction, and Operations& Maintenance*), como columna vertebral que proporciona la experiencia técnica necesaria para diseñar, construir y gestionar, operar y mantener los activos físicos que sustentan nuestras sociedades.

Esta tesis enfrenta diversos desafíos que abarcan desde la conceptualización del gemelo digital (DT, *Digital Twin*) en la ingeniería civil hasta su aplicación práctica en estructuras dentro de los casos de estudio. La resolución de los mismos ha implicado la integración de datos y modelos dentro de un marco

estadstico Bayesiano con el fin de abordar la actualización y cuantificación de la incertidumbre, as como la gestión de los flujos de trabajo del gemelo digital mediante una red de Petri de alto nivel. Además, se ha enfrentado la limitada disponibilidad de datos para el entrenamiento, junto con el establecimiento de una serie de modelos subrogados para facilitar el diagnóstico y pronóstico dentro del marco del gemelo digital. Para abordar estos retos se han empleado estrategias de IA basadas fundamentalmente en redes neuronales (NN, *Neural Networks*) y modelos de aprendizaje profundo (DL, *Deep Learning*).

Summary

Civil engineering stands at a critical juncture in the XXI century, facing a confluence of challenges and opportunities that demand a paradigm shift in practices and methodologies. As most structures built at the beginning of the past century globally approach the culmination of their designed lifespans, the need for sustainable, resilient, and technologically advanced solutions becomes paramount. To confront the complexity, this doctoral thesis endeavours to develop a comprehensive digital twin conceptual framework tailored for civil engineering, aware of ageing infrastructure, digitalisation, environmental impact, and the imperative to minimize waste.

With numerous constructions approaching the end of their expected life cycles, the challenge lies not only in preserving the integrity of these structures, but also in reimagining them for a sustainable future. This work aims to investigate strategies for operation and maintenance, retrofitting, and sustainable policies.

Regarding digitalisation, while strides have been made in embracing digital tools for design, construction, and project management, a fragmented landscape of technologies still persists. Siloed efforts limit the transformative potential of integrated digital solutions. This thesis aims to provide the pathways toward a cohesive and collaborative implementation of technologies under the umbrella of the digital twin. In the pursuit of efficiency and quality, civil engineering is witnessing a wave of innovation driven by emerging technologies. Building Information Modelling (BIM), the Internet of Things (IoT), and Artificial Intelligence (AI) are reshaping the industry landscape. This work critically evaluates the adoption and impact of these technologies, assessing their potential to revolutionise the practice of civil engineering in the Architecture, Engineering, Construction, and Operations and maintenance (AECO) sector, as the backbone providing the technical expertise needed to design, build, manage, operate and maintain the physical assets that support our societies.

This thesis has confronted several challenges, encompassing the thorough conceptualisation of the Digital Twin (DT) for civil engineering with application in structures. It has also involved the integration of data and models within a Bayesian statistical framework to address updating and uncertainty quantification, the management of digital twin workflows through a high-level Petri net, the limited availability of data for training, and the establishment of a pipeline of surrogate models to facilitate diagnosis and prognosis within the DT framework. To overcome these challenges, AI strategies have been introduced, relying on

Neural Networks (NN) and Deep Learning (DL) models.

Agradecimientos

Agradezco a mis directores de tesis, Dr. Manuel Chiachío Ruano y Dr. Francisco Javier Melero Rus por su guía y apoyo a lo largo de toda la tesis, al igual que a mi tutor de departamento, Dr. Enrique Hernández Montes. También quisiera dar las gracias a Ramón Jesús Suárez Pérez por el despliegue tecnológico de los casos de estudio.

Igualmente expreso mi gratitud hacia al proyecto europeo H2020 MSCA-ITN ”ENHAnCE” con identificador 859957, así como al proyecto ”Desarrollo de un sistema de apoyo a la toma de decisiones basado en herramientas de gestión inteligente para el mantenimiento de construcciones existentes y diseño de nuevas” con referencia RTI2018-101841-B-C21, de la Agencia Estatal de Investigación (AEI) y el Fondo Europeo de Desarrollo Regional (FEDER), por la financiación y el aprendizaje que me han brindado.

Acronyms

AECO	Architecture, Engineering, Construction, Operations & Maintenance
AI	Artificial Intelligence
API	Application Programming Interface
AUC	Area Under the ROC Curve
BIM	Building Information Modelling
BP	Back Propagation
CAD	Computer-Aided Design
CI	Confidence Interval
CNN	Convolutional Neural Network
CPS	Cyber Physical System
CS	Communication System
CSV	Comma-Separated Values
CWGAN-GP	Conditional Wasserstein Generative Adversarial Network with Gradient Penalty
DL	Deep Learning
DM	Diffusion Model
DSS	Decision Support System
DT	Digital Twin
EU	European Union
FE	Finite Element
FID	Frechet Inception Distance
FMECA	Failure Modes, Effects, and Criticality Analysis
FNN	Feed forward Neural Networks
FSM	Finite State Machine
FTA	Failure Tree Analysis
GAN	Generative Adversarial Network

GIS	Geographic Information System
GP	Gaussian Process
GSM	Global System for Mobile communication
HLPN	High Level Petri Net
HTTPS	Hypertext Transfer Protocol Secure
IFC	Industry Foundation Class
IoT	Internet of Things
JSON	JavaScript Object Notation
KBS	Knowledge Based System
LeakyReLU	Leaky Rectified Linear Unit activation function
LED	Light-Emitting Diode
MAE	Mean Absolute Error
MC	Monte Carlo method
MCMC	Markov Chain Monte Carlo method
MEM	MicroElectro-Mechanical systems
M-H	Metropolis-Hastings algorithm
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MSE	Mean Squared Error
M2M	Machine-to-Machine
MVC	Model-View-Controller
NDT	Non-Destructive Testing
NLP	Natural Language Processing
NTP	Network Time Protocol
O&M	Operations and Maintenance
OOP	Object-Oriented Programming
OSI	Open Systems Interconnection model
PE	Physical Entity
PDF	Probability Density Function
PHM	Prognostics and Health Management
PINN	Physics-Informed Neural Network
PN	Petri Net
PMIE	Principle of Maximum Information Entropy
ReLU	Rectified Linear Unit activation function
REST	Representational State Transfer
RESTful	REST-compliant system

RFC	Request For Comments standard
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RPN	Risk Priority Number
RUL	Remaining Useful Life
R^2	Coefficient of determination
ROC	Receiver Operating Characteristic curve
SDG	Sustainable Development Goal (UN Agenda 2030)
SDI	Severity Damage Index
SHM	Structural Health Monitoring
SMS	Short Message Service
SOA	Service Oriented Architecture
SoC	System-On-Chip
SSIM	Structural Similarity Index Measure
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
UL	Useful life
UQ	Uncertainty Quantification
VAE	Variational Autoencoder
VE	Virtual Entity
VI	Variational Inference
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Symbols

α	Parameter of the Metropolis-Hastings algorithm (Chapter 7)
	Stiffness reduction coefficient in the Case Studies (Chapters 11 and 12)
A	Incident matrix in a Petri Net
b	Bias in a NN
c	Code or condition vector
	Damping in the state-space equation
$\mathcal{C}$	Especific context in a state-space equation
	Set of conditions associated with transitions in a Petri net
C	Convolution operation in a CNN
CS	Communication system
d	Displacement
$\hat{d}$	Observed / modelled displacement
$\mathcal{D}$	Observation Space
$D()$	Discriminator in a GAN, also called *Critic* in a WGAN-GP
ε	Error from data and models
e	Vector of environmental variables
E	Set of weighted arcs in a Petri net
$\mathbb{E}$	Expectation
F	Force
γ	Hyperparameter for balancing the *data* component of the loss
∇	Gradient operator
$G()$	Generator in a GAN
$\mathcal{HLPN}$	High Level Petri net
$\mathbb{I}$	Indetity matrix
$\mathbb{I}_c$	Boolean condition

k	Execution time or state in a Petri net
K	Stiffness coefficient in the state-space equation
$L()$	Loss function in a NN
l_r	Learning rate of an optimiser in a NN
λ	Gradient penalty coefficient
	Hyperparameter for balancing the *regularisation* component of the loss
μ	Hyperparameter for balancing the *physics* component of the loss
m	Mass in the state-space equation
	Size of a batch in a NN
$m()$	Model instance
$\mathcal{M}$	Model class
M	Marking function in a Petri Net
M_0	Initial marking function in a Petri net
M_k	Marking function at state k in a Petri Net
N	Number of samples
$\mathcal{N}()$	Gaussian probability distribution
$p(\cdot)$	Probability density function
p_j	Place, with $j = 1, \ldots, np$; np : number of places in a Petri Net
π_r	PDF associated to the real data
π_g	PDF associated to the noise random latent vector
P	Set of places in a Petri net
$P(\cdot)$	Probability
PN	Petri Net
σ	Standard deviation
$\sigma()$	Activation function in a NN
$s()$	Real state-space
$\hat{s}()$	Observed / modelled state-space
S	Subsequence in a CNN
θ	Model parameters
θ_g	Generator model parameters
θ_c	Critic model parameters
Θ	Set of plausible values of model parameters
t_i	Transition, with i=1,...,nt; nt: number of transitions in a Petri net
t	Time
T	Set of transitions in a Petri net
	End of the UL

u	Vector of input variables
u_k	Firing vector at execution time k in a Petri net
$\mathcal{U}()$	Uniform probability distribution
w	Vector of input measurements
W	Weight of a NN
$W()$	Wasserstein-1 distance fucntion
x	Real data
	Real input (independent variable)
$\tilde{x}$	Latent noise vector
$\hat{x}$	Observed/modelled input
$x_{physics}$	Physics data input
y	Real output (dependent variable)
$\hat{y}$	Observed/model output
	Model's prediction
$y_{physics}$	Physics data output
z	Random number

Part I

Introduction

Chapter 1

Context and motivation

Most of the existing civil engineering structures in the world, including bridges, dams and big buildings, date from the first half of the 20th century. Given the optimistic average assessment of a 100-year Useful Life (UL) for reinforced concrete and steel structures, a considerable number of these constructions are approaching the culmination of their operational lifespan [1]. Furthermore, societal factors such as increased population growth, higher traffic volumes, heavier vehicles, changes in use, or structural modifications, have resulted in the overloading of many existing structural elements within bridges, buildings and other structures. Thus, they must be maintained properly to adapt them to changing conditions of use, including unforeseen climate-driven events. Significant resources are being invested, although they are employed mostly for inspection and correction using the current approaches. Diagnosis methods, often applied visually and even belatedly, lead to partial structural assessment and sometimes do not prevent structural failures. The decision-making process regarding whether to repair or demolish also consumes valuable time, and the implications of suboptimal or delayed solutions inevitably impact safety and sustainability, together with economy [2]. Construction, being one of the sectors with the highest mobilisation of economic resources, amplifies the repercussions of such decisions. According to the United Nations Environment Programme (UNEP), the sector that includes Architecture, Engineering, Construction, Operations & Maintenance (AECO) contributes to nearly 40 % of global CO_2 emissions related to energy, mainly attributed to both energy consumption and materials employed in construction [3].

Simultaneously, the digitalisation of society is rapidly advancing. The development of monitoring through sensors connected to the Internet (IoT, Internet of Things) alongside the advancement of analytical tools, enable the real-time collection of extensive environmental data. These data can be promptly acquired, processed, analysed, and utilised to draw conclusions for subsequent decision making. The entire process is executed ensuring immediacy and achieving greater efficiency at a reduced economic cost. In this context, the notion of the Digital Twin (DT) emerges as one of the fast-evolving digital technologies that support the broader digital transformation of society.

The foundation of the DT lies in an advanced cyber-physical system (CPS), on which bidirectional communication is established between the real and the digital world, ensuring continuous and seamless interconnection. The DT concept, originally formulated as a digital replica of a physical asset, originated at NASA [4] during the 1960s whithin the Apollo mission (Figure 1.1, from [4]). The concept involved creating a *living model* of the Apollo spacecraft, where NASA constructed a replica subjected to identical conditions as the one sent into space. This approach, which leveraged high-fidelity digital models encompassing both physical systems and operating environments, allowed the solutions implemented to solve technical challenges in each instance to serve as mutual learning. Afterwards, the term was coined in 2002 by Michael Grieves at the University of Michigan [5] although its first practical application to structures took place in 2012 in the aerospace sector [6, 7]. Nowadays, the DT undergoes real-time autonomous analytics and AI processes, encompassing capabilities such as diagnostics, referred in the engineering field as *Structural Health Monitoring* (SHM), and prediction under *Prognosis and Health Management* (PHM), all enabled by its underlying models.

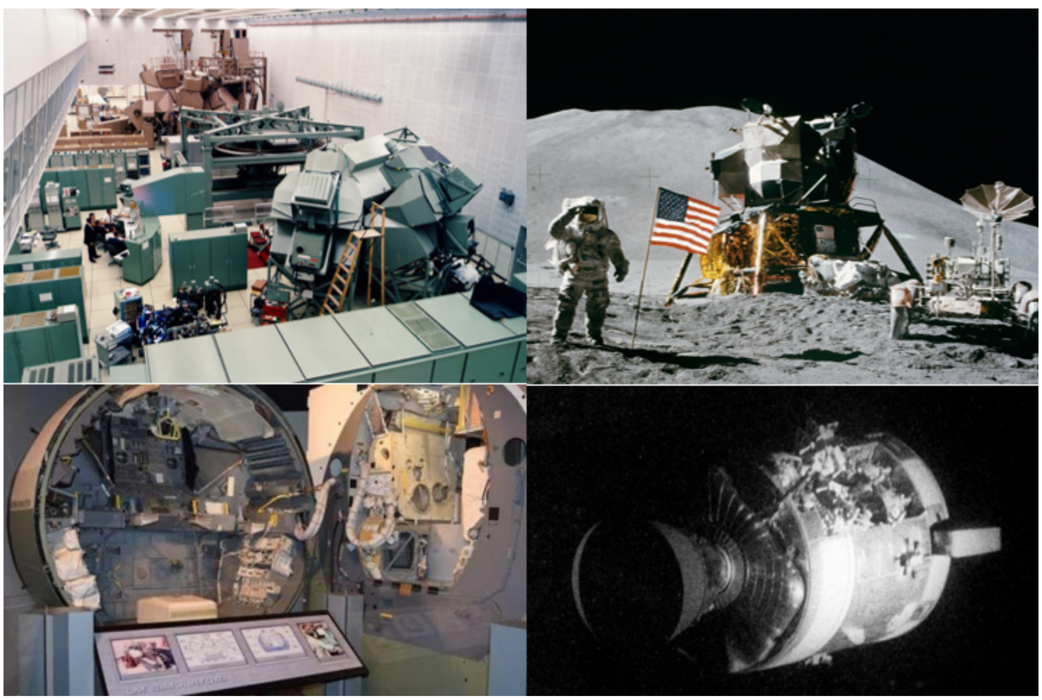


Figure 1.1: NASA's DT. On the left side, the DT located in NASA's headquarters, while on the right, the physical counterpart was in space.

It is widely acknowledged that the DT is not itself a single technology but rather the orchestrated aggregation of multiple technologies to serve the operated asset. When applied to civil engineering, the DT's primary objective is to provide optimal decision support thereby enhancing the management, reliability, and sustainability of structures [8, 9]. In this field, the DT can encompass the infrastructure from design and construction stages to ongoing management in operations and maintenance throughout its entire lifecycle, optimising each stage, as illustrated in Figure 1.2. Consequently, the definition of the

DT varies based on the operable asset and its configuration, although in essence, it involves the virtual representation of the asset's properties, states, and behaviours, relying on theoretical models and real data. This virtual representation remains synchronised with the physical object, enabling real-time monitoring of physical changes (via sensors) and facilitating assisted and/or automatic actions (using actuators). These actions are guided by the implemented analytics with the aim of optimising the asset's performance.

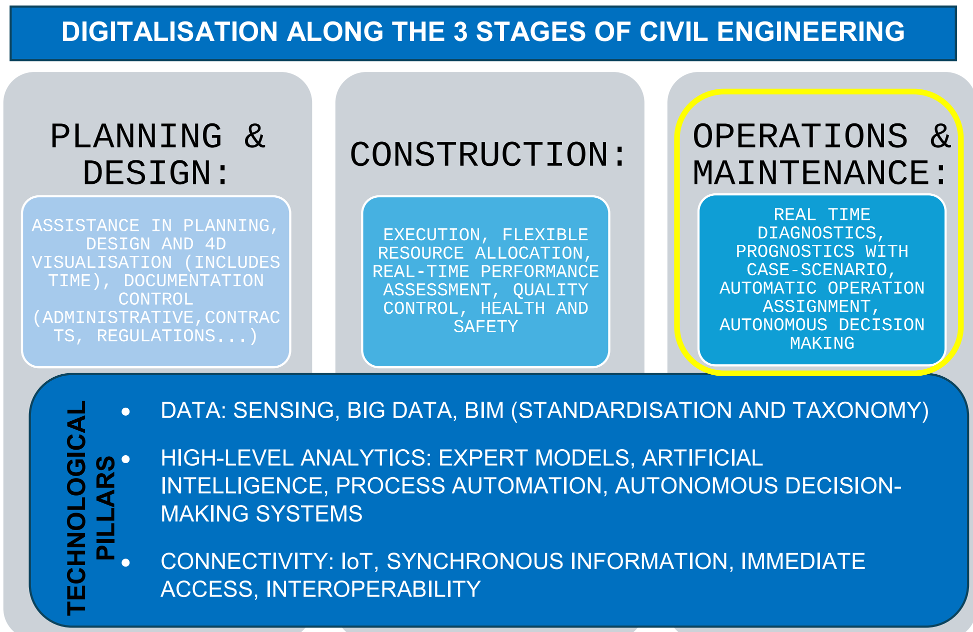


Figure 1.2: Digitalisation in civil engineering, highlighting the focus on the Operations & Maintenance stage.

The DT technology has been recognised as pivotal for industrial development since 2017, as acknowledged by technology consultancies such as Gartner[1] (refer to Figure 1.3 for further insights), and has received considerable attention in the scientific literature related to the industry [10–13]. In this context, the DT term has been sometimes embedded within the concept of *hyperautomation*, which is intended to give a fundamental role to AI in the industry and thus increase the trend towards AI-driven decision-making processes following the *Industry 4.0* technological trend [14]. This seems to be the future to which this technology is directed, although its full implementation in civil engineering and the AECO sector is still pending. This sector exhibits lower levels of digitalisation compared to industries such as aeronautics and manufacturing, where DT technology has been established for decades [15]. In the civil engineering domain, DT technology is still in its infancy as compared to other industries, with the earliest contributions found in the literature dating from 2018 [8]. The delay in the adoption of DT technology in civil engineering, as opposed to fields like mechanical engineering, can be attributed to several factors. Firstly, civil constructions typically have a much longer lifespan, averaging around 100 years, as opposed to mechanical engineering assets, which typically last about 15 years on average [16]. Additionally, civil engineering assets are

[1]https://www.gartner.com/en

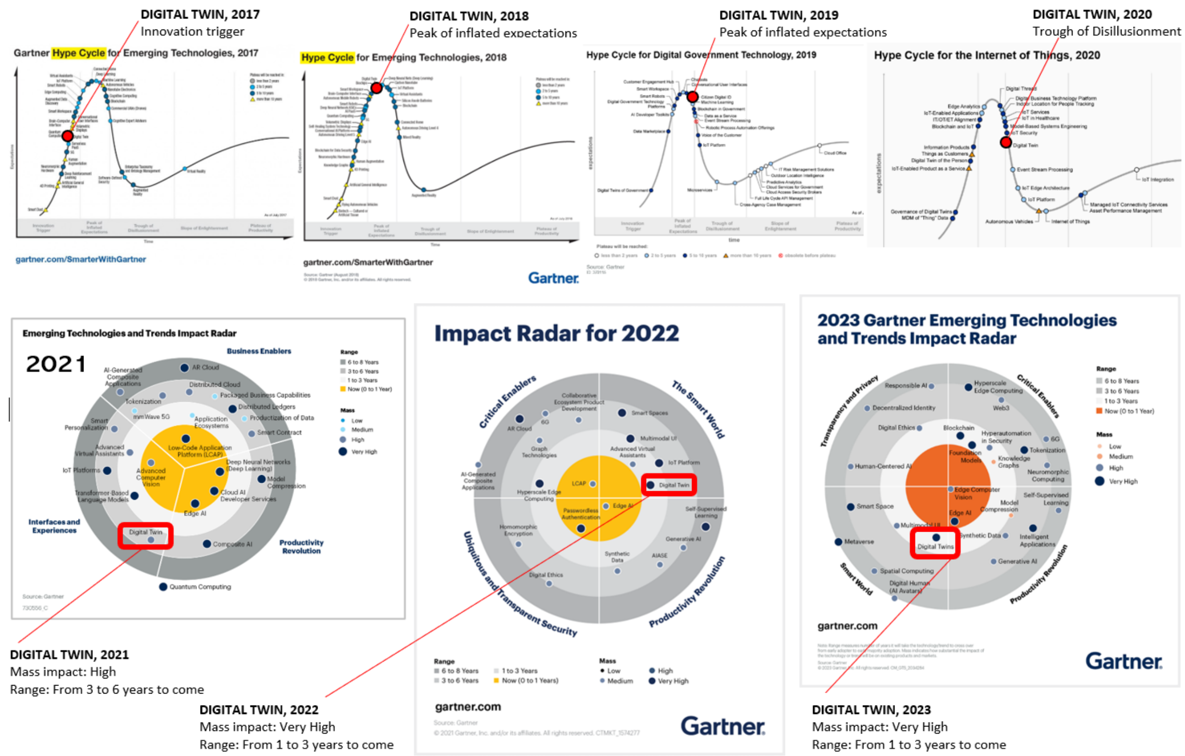


Figure 1.3: The DT as an emerging technology in the industry since 2017, according to Gartner.

individually designed for specific purposes and environments, as opposed to those being mass-produced for general purposes. Lastly, the application of modern sensing techniques to civil engineering has only attracted the attention of the industry in recent times, as compared to other sectors. Nowadays this topic is receiving increasing attention, possibly as a consequence of a technology–push by the irruption of the industrial IoT [17], and a demand–pull, due to modifications in the use of structures with increasing demands, climate change, and accumulated ageing [18].

To date, the majority of the scant representations of the DT in civil engineering have primarily focused on simulation, applying theoretical models that lack integration with real-time data or the incorporation of AI. Other representations encountered mainly involve BIM models designed to store static information regarding the geometric and material characteristics of the operated asset. However, these models lack analysis, decision-making capabilities, or connections to operations and maintenance (O&M) throughout the asset's useful life. The rationale behind this thesis is driven by the imperative to integrate these functionalities cohesively and complementarily under the term *Digital Twin*. This integration aims to amplify the synergy of the entire system, supporting the monitoring of infrastructures throughout their lifecycle. The goal is to enable intelligent and autonomous maintenance, prolonging service life, reducing costs, and ensuring safety.

Moreover, the motivation of this thesis also aligns with the 2030 Agenda for Sustainable Development [19], and particularly the Sustainable Development Goal 9 (SDG 9), which focuses on 'Industry, Innovation, and Infrastructure'. DTs contribute to infrastructure development by facilitating planning, design, construction, maintenance and management processes, leading to the creation of more resilient and sustainable infrastructure. This technology also promotes industrialisation in the construction sector by standardising processes and embracing interoperable methods. Furthermore, DTs support resource efficiency goals by enabling predictive modelling, optimisation, and real-time monitoring, thereby enhancing sustainable practices within construction projects, and afterwards, O&M. Goal 11 'Sustainable Cities and Communities' is also closely aligned, as there are several specific targets related to urban planning and management, improving road safety, and protecting cultural and natural heritage, among other aspects. The sustainability of the construction sector is essential to achieving this goal, as building construction and urban planning have significant impacts on the environment, society, and economy. Overall, the deployment of DTs in the civil engineering field serves as a practical application that embodies the targets of SDGs 9 and 11, fostering innovation, enhancing infrastructure development, and promoting sustainable practices.

In summary, this thesis aims to incorporate the state-of-the-art capabilities of the DT, contributing to the advancement of civil engineering in the ongoing trends of sustainability and digitalisation. The overall motivation is to improve the efficiency of managing, maintaining, and operating existing and future civil assets, aligned with the European Union (EU) and United Nations (UN) sustainability objectives [20], and ultimately working towards reducing environmental impact.

Chapter 2

Literature review

A literature review has been conducted to obtain a technical understanding of the evolving field of DT, the identification of emerging trends, and the evaluation of established practices. Its aim is not to be exhaustive in its coverage, but rather to focus on the principal and most relevant publications within the area of structural civil engineering. By synthesising a wide array of academic contributions, this literature review facilitates the identification of key components and their corresponding functionalities that are integral to the successful implementation of DTs in the field of structures within the civil engineering domain.

2.1 Introduction

The present survey has identified and analysed 65 articles published from 2018 to the present, focussing on the application of DT in structural assets. These assets include bridges, offshore structures, hydraulic infrastructure (such as dams), as well as contemporary and historical buildings, roads, and railway infrastructure. These publications have been selected for the review as practical implementations while excluding literature reviews and theoretical developments. The rationale behind this exclusion is the lack of tangible materialisation of the concept in real-world applications. While theoretical works are crucial for establishing foundational principles and exploring theoretical foundations, their absence of practical validation leaves uncertainties regarding their applicability and effectiveness in physical contexts.

As a result, this review concentrates on conceptualisations encompassing practical case studies of DT in structural civil engineering. This emphasis ensures that their insights and findings are directly relevant and provide tangible evidence of the feasibility and effectiveness of DT in real-world scenarios. It is worth noting that while literature reviews and theoretical formulations of DTs abound, practical applications in real-world contexts are comparatively limited in number.

2.2 Discussion

The vast majority of the publications reviewed predominantly focus on constructing the representation of the DT for pre-existing structural engineering assets, rather than for new designs. These studies present an initial survey and conceptualisation of the DT, which is characterised by an ambitious vision compared to its deployment in real-world applications. The numerous challenges faced in the practical implementation are discussed by the authors, showing that still further research is needed to fully implement the DT in the structural engineering domain. Grieves (2006) [21] introduced the capability hierarchy of DT as a progressive scale of sophistication. This scale includes five different levels of achievement: supervision (Level 1); operation (Level 2); simulation (Level 3); learning including simulation and prediction, aided by AI (Level 4); and finally, autonomous management (Level 5). Several authors, including [22, 23], have used this scale to categorise their deployments, performing their DT applications at levels up to 3 and 4. Therefore, further research and development are required to advance the implementation of higher-level DTs.

The review has been summarised in Table 2.1, providing a systematic overview of the contributions in DT structural engineering that have been examined. Given the recognised absence of a unified definition of DT within the academic community, including the civil and structural engineering field [24–26], this study aims to identify key components in each revised perspective of DT. Consequently, the reviewed articles are categorised based on criteria such as the integration of SHM techniques for data monitoring, adoption of IoT for communication, incorporation of AI, or the presence of uncertainty quantification, among other factors. The presence or absence of individual components within the DT offers unique insights into how the DT virtually mirrors real-world systems or processes, providing clarity on their practical implementation in real-world scenarios.

In the subsequent sections, each constituent of the DT identified in the literature review will be examined. Accompanying these discussions will be the main references related to each component, cited in the APA[1] style. Additionally, the general IEEE[2] style adopted throughout this work will be maintained to quote the other pertinent citations.

By thoroughly examining the perspectives presented in each article, valuable insights can be gleaned regarding the significance and role of each component identified within the DT framework and how they interact with one another. This process aids in understanding the importance of each component and its contribution to the DT functionality, thereby clarifying its significance and practical implementation in real-world applications. This comprehensive analysis helps identify best practices and potential areas for improvement in the implementation of DT, ultimately contributing to the advancement and refinement of DT technology in the civil engineering field.

[1] $https://apastyle.apa.org/instructional-aids/reference-guide.pdf$
[2] $https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf$

Table 2.1: Synoptic table of bibliography including practical applications related to DT in civil and building structural engineering.

Ref.	Year	Field	SHM	IoT	BIM	AI	MOD	INF	DIAG	PHM	UQ	WF	DS	ADM
[27]	2018	Bridge O&M	✓	-	✓	✓	✓	-	✓	-	-	-	-	-
[16]	2019	Offshore assessment	✓	✓	✓	-	✓	-	-	-	-	-	-	-
[28]	2019	Bridge assessment	✓	-	✓	-	✓	-	✓	-	-	✓	-	-
[29]	2019	Bridge assessment	✓	-	-	-	✓	-	✓	-	-	-	-	-
[30]	2019	Bridge assessment	✓	-	✓	-	✓	-	✓	-	-	-	-	-
[31]	2019	Bridge assessment	✓	-	✓	✓	✓	✓	-	-	✓	-	-	-
[32]	2019	Bridge O&M	-	-	✓	-	✓	-	✓	-	-	-	-	-
[33]	2019	Building monitoring	-	-	✓	-	✓	-	✓	✓	-	-	-	-
[34]	2019	Bridge monitoring	✓	-	✓	-	✓	-	✓	✓	-	-	-	-
[22]	2020	Structural assessment	✓	-	-	-	✓	✓	✓	-	✓	-	-	-
[24]	2020	Eng. Systems degradation	✓	✓	-	✓	✓	-	✓	✓	-	✓	✓	✓
[35]	2020	Structural assessment	✓	-	✓	✓	✓	-	-	-	-	-	-	-
[36]	2020	Structural assessment	✓	-	-	✓	✓	✓	✓	✓	✓	✓	-	✓
[37]	2020	Building assessment	-	-	✓	-	✓	-	✓	✓	-	-	-	-
[38]	2020	Bridge assessment	✓	-	✓	-	✓	✓	✓	-	-	-	-	-
[39]	2020	Building O&M	✓	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓
[25]	2021	Structural assessment	✓	✓	✓	-	✓	-	-	-	-	✓	✓	-
[26]	2021	Structural assessment	-	-	-	✓	✓	-	✓	-	✓	-	-	-
[40]	2021	Structural assessment	✓	-	-	-	✓	-	✓	✓	✓	-	-	-
[41]	2021	Bridge assessment	✓	-	✓	✓	✓	-	✓	-	-	-	✓	-
[42]	2021	Offshore assessment	✓	✓	-	✓	✓	-	✓	-	✓	-	✓	-
[43]	2021	Bridge assessment	✓	✓	✓	✓	✓	-	✓	-	-	-	✓	-
[44]	2021	Bridge assessment	✓	✓	✓	✓	✓	-	✓	-	-	-	✓	-
[45]	2021	Hydraulic eng. assessment	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
[23]	2022	Bridge O&M	✓	-	✓	-	✓	-	✓	-	-	✓	✓	-
[46]	2022	Structural assessment	✓	✓	-	✓	✓	✓	✓	-	✓	✓	✓	✓
[47]	2022	Structural assessment	✓	-	-	✓	✓	✓	✓	-	✓	-	-	-
[48]	2022	Bridge O&M	✓	-	-	-	✓	✓	✓	-	✓	-	-	-
[49]	2022	Structural assessment	-	-	-	-	✓	-	-	-	-	✓	-	-
[50]	2022	Structural O&M	✓	✓	✓	-	✓	-	✓	-	-	✓	✓	-
[51]	2022	Bridge assessment	✓	✓	✓	-	✓	-	✓	-	-	-	-	-
[52]	2022	Bridge assessment	✓	-	-	-	✓	-	✓	-	-	-	-	-
[53]	2022	Bridge assessment	✓	✓	✓	-	✓	-	✓	-	-	-	✓	-
[54]	2022	Construction & maint.	✓	-	✓	-	✓	-	-	-	-	✓	✓	-
[55]	2022	Bridge assessment	✓	-	-	-	✓	-	✓	-	-	-	-	-
[56]	2022	Structural assessment	✓	-	-	-	✓	✓	✓	-	✓	✓	-	-
[57]	2022	Bridge assessment	✓	-	-	-	✓	-	✓	✓	✓	✓	-	-
[58]	2022	Building assessment	✓	-	-	-	✓	-	✓	-	✓	-	-	-
[59]	2022	Road assessment	✓	✓	-	-	-	-	-	-	-	-	-	-
[60]	2023	Bridge assessment	✓	-	✓	-	✓	-	-	-	✓	-	-	-
[61]	2023	Bridge assessment	✓	-	✓	-	✓	-	✓	-	-	-	-	-
[62]	2023	Building assessment	✓	-	-	-	✓	✓	✓	-	✓	✓	-	-
[63]	2023	Building assessment	✓	✓	-	-	✓	-	✓	✓	✓	✓	✓	-
[64]	2023	Structural assessment	✓	-	✓	-	✓	-	✓	-	-	✓	-	-
[65]	2023	Structural assessment	✓	-	-	-	✓	✓	✓	✓	✓	✓	✓	-
[66]	2023	Building assessment	✓	✓	✓	-	✓	-	✓	-	✓	-	✓	-
[67]	2023	Structure assessment	✓	-	-	✓	✓	-	✓	-	-	✓	✓	-
[68]	2023	Bridge assessment	✓	-	-	✓	✓	-	✓	-	-	-	-	-
[69]	2023	Building assessment	✓	-	-	-	✓	-	✓	-	-	-	-	-
[70]	2023	Building assessment	✓	-	✓	-	-	-	✓	-	-	-	-	-
[71]	2023	Bridge assessment	✓	-	-	✓	✓	-	✓	-	-	-	✓	-
[72]	2023	Bridge assessment	✓	-	-	✓	✓	-	✓	-	-	-	-	-
[73]	2023	Building assessment	-	-	✓	-	✓	-	✓	-	-	✓	-	-
[74]	2023	Structural assessment	✓	-	-	-	✓	✓	✓	-	✓	-	-	-
[75]	2023	Structural assessment	✓	-	-	✓	✓	-	✓	-	✓	✓	-	-
[76]	2023	Structural assessment	✓	-	-	✓	✓	✓	-	-	✓	-	✓	-
[77]	2023	Offshore assessment	✓	-	-	-	✓	-	-	-	✓	-	-	-
[78]	2023	Structural assessment	✓	-	-	-	✓	✓	-	-	✓	✓	-	-
[79]	2023	Structural assessment	✓	-	-	✓	✓	-	-	-	-	-	-	-
[80]	2023	Structural assessment	✓	-	-	-	✓	✓	✓	-	✓	-	-	-
[81]	2023	Bridge assessment	✓	-	-	-	✓	-	✓	-	-	-	-	-
[82]	2024	Railway str. assessment	✓	-	-	✓	✓	-	-	-	-	-	-	-
[83]	2024	Structural assessment	✓	-	-	-	✓	✓	✓	-	✓	✓	-	-
[84]	2024	Building construction	✓	✓	✓	-	-	-	-	-	-	✓	-	-
[85]	2024	Structural assessment	✓	-	-	✓	✓	✓	✓	-	✓	✓	✓	-

Ref.: Reference; SHM: Data obtained by SHM techniques; IoT: Use IoT; BIM: Use Building Information Modelling; AI: Use Artificial Intelligence; MOD: Use models; INF: Makes Bayesian Inference;DIAG: Makes Diagnostics; PHM: Make Prognostics; UQ: Provides Uncertainty Quantification; WF: Includes a workflow model; DS: Provides decision support; ADM: Makes Autonomous decisions

Buiding Information Modeling, BIM

The review unveils the earliest references of DT applications in structural civil engineering dating back to 2018, with consideration given to computer-aided design (CAD) and BIM as precursors to the DT [16, 27, 28]. These methodologies enable the use of digital representations of a built asset to streamline design and construction processes [30, 32, 61, 64, 70, 73]. However, it is important to recognise that BIM models only represent a specific asset at a particular moment in time. Essentially, creating a BIM model of an asset is akin to capturing a snapshot of the asset at that precise moment. Consequently, dynamic updates are not possible without manual intervention, rendering this methodology alone unable to meet the requirements of developing a DT. Additionally, noteworthy attention has been given to the Industry Foundation Classes (IFC) standard by the authors [28, 32, 60]. Developed by the International Organization for Standardization (ISO), the IFC standard serves as a platform-neutral data format facilitating the exchange of BIM data among various software applications in the AECO sector.

Models

In the range of DT formulations explored, models emerge as a common feature for the DT implementation, diverging from those DT formulations that mainly consist of geometric descriptions supplemented with monitoring data [59, 84], or strictly BIM representations [70]. Models are key in the DT conceptualisation made by Gartner et al. (2020) [36] although it is emphasised that DTs encompass more than just validated models. Throughout the literature review, models are used as a means of processing and interpreting monitoring data. Ye et al. (2019) [31] describe a dual-perspective approach to modelling, involving both physics-based and data-driven methods. The physics-based approach relates sensor measurements with prior model predictions based on first principles, code formulas, or finite element (FE) models, and explains discrepancies by inferring real structural condition and executing model updating to minimise these discrepancies. Meanwhile, the data-driven approach formulates statistical models based solely on data, identifying trends, patterns, and correlations, and quantifying uncertainties of structural condition and performance, all over an often unsupervised approach and without including physical guidance.

As an evolution, hybrid approaches such as the 'data-centric engineering approach' proposed by Ye et al. (2020) [38], combine data-driven and physics-informed methods, integrating data with Gaussian processes (GP) derived from FE models validated with real monitoring data. Another combined procedure is presented by Li et al. (2021) [45], which have developed a deterministic model and simulation analysis serving as a rapid structural calculation method based on BIM and FE analysis. This approach involves transferring the physical characteristics of the asset directly from the BIM browsing module to the FE calculation module. Within the FE module, external load conditions of the model are altered, and the operational status of the system is evaluated using a deterministic model based on FE structural calculations. This methodology allows for the simulation of load conditions that have not occurred previously, thereby enabling anticipation of the asset's operational behaviour.

Models need updating for the DT to remain accurate and reliable over time. Ye et al. (2020) [38] employs FE model updating methods, which involve direct update and iterative approaches. Modal analysis is commonly employed when dealing with dynamic loads, such as those experienced by bridges, and allows for structural damage identification and scenario simulation. Additionally, discrete models are also utilised in DT applications, as noted by Ritto et al. (2021) [26].

Prognostics and Health Management, PHM

In structural engineering, models are also employed to simulate different stages. For example, Angjeliu et al. (2020) [37] focus on the modelling stage construction, structural evolution, current stage conditions, and future damage prediction for preventive maintenance. Damico et al. (2020) [24] explore models for formulating asset degradation over time, simulating phenomena such as creep and shrinkage, particularly relevant for historical buildings. They note that degradation of engineering structures typically occurs due to factors like wear, corrosion, and fracture, progressively leading to performance decay until system failure. Assessing degradation and forecasting the RUL of the asset are crucial aspects of DT in structural engineering, with only a limited number of publications exploring prognostics and health management (PHM) capabilities, such as in [33, 34, 40, 77].

The integration of a flexible array of models into the DT framework is enabled by its capability to host and couple diverse modelling approaches. Liu et al. (2020) [35] integrate up to four types of models—geometric, physical, behavioural, and rule models into the DT framework.

Internet of Things, IoT

In recent times, the rapid development of sensing technologies has allowed more data to be gathered from the built environment than ever before. In general, four types of monitoring data can be gathered [18]: response-based, such as strain, displacement and inclination; geometry-based, such as conventional surveying and laser scanning; vision-based, such as image and video; and loading, such as operational and environmental loadings. More recently, the technological developments of wireless sensor networks (WSNs) and the IoT have enabled advancements in integrated sensing and analytics. WSNs within the IoT have garnered growing attention due to their capability to be deployed across large infrastructure, facilitating real-time data connectivity for real-time analysis. These networks boast flexible and low-power consumption characteristics. Nevertheless, less than half of the reviewed articles currently place trust in this technology and acknowledge its potential for enabling real-time DT operations and facilitating edge computing. Moreover, remote locations, historic buildings, and existing complex infrastructures are particularly well-suited to this technology, as demonstrated in studies such as those by Grosse et al. (2019) [16], and others [45, 51, 53, 63, 66].

Structural Health Monitoring, SHM

Monitoring data is essential for SHM, which aims to assess structural performance during service and operation phases, control the margin of acceptable designed structural behaviour, and mitigate any defects by analysing their behavioural change with time [31]. When performed on line and autonomously, monitoring becomes SHM [86]. SHM is present in nearly every article reviewed, with authors unanimously recognising its crucial role in DTs, such as in [29, 52, 55, 69, 81]. They agree that SHM provides essential data necessary for creating digital replicas of physical assets and facilitates ongoing communication between the physical asset and its digital counterpart. This continuous exchange ensures mutual updates and synchronisation, underlining SHM's significance in the development and maintenance of DTs, as encountered at [25, 48, 55, 68].

The fusion of SHM data, BIM, FE analysis, and statistical modelling is envisioned by Ye et al. (2023) [65] as the next direction in research aimed at enhancing the monitoring and life-cycle management of structures such as bridges. The significant advantages of DT for bridges are highlighted by several authors [41, 43, 57], including streamlined data, integrated data processing capabilities, and a unified collaborative environment spanning their entire lifecycle. However, they concluded that further efforts are needed to effectively integrate data and models. From a practical standpoint, implementing a DT poses challenges: the complexity level is substantial, and a standardised process has yet to be established [65].

Diagnostics

Damage diagnostics in structural engineering helps in assessing the health and condition of the structure by identifying structural issues like cracks, excessive displacements, and deformations, providing insights into structural performance which are key for the decision making in safety and maintenance [47, 52, 67]. For this reason, diagnostics is prevalent within the research focused on O&M than those in construction.

Among the different approaches encountered in the review for damage diagnostics, a distinction can be made between unsupervised anomaly detection such as in Lu et al. (2020) [39] and supervised diagnostic that can be found in Torzoni et al. (2024) [85]. Anomaly detection aims to identify patterns that differ significantly from normal behaviour without the need for labelled examples [87]. While these methods can highlight abnormal conditions, they do not provide detailed information about the specific type or extent of damage [71, 72, 75]. On the contrary, supervised diagnostic approaches can classify and characterise damage more precisely by using labelled examples of known damage types, identifying different characteristics of the damage [85].

Uncertainty quantification and Inference

Uncertainty quantification is not always included in the articles reviewed. Several authors [22, 26] incorporate it as a measure of uncertainty from data and models, often jointly, in order to reflect the overall level of uncertainty inherent in both the model predictions and the gathered data. The methodology employed include techniques such as linear and non-linear regression, although Bayesian methods are preferred [56, 62, 74, 78, 80, 83]. A major constraint for uncertainty quantification is the need for operating in real time or almost real time, as indicated by Wag et al. (2020) [22], demanding the development of highly computationally efficient estimation techniques and the adoption of fast-running statistical surrogates that approximate the response of the underlying computational models within the DT.

The updating of the DT can be undertaken in two ways: by directly measuring the model parameters and consequently inserting them into the model; or indirectly updating the model parameters through an inference process by measuring the structural response with sensors and comparing it to the model prediction. Among the authors reviewed, Bayesian inference emerges as the preferred method for inference in the DT applications [22, 36, 39].

Workflow

The necessity of a workflow is introduced in the DT by Lu et al. (2019) [28], Damico et al. (2020) [24] and Gardner et al. 2020, [36] in order to coordinate its multiple tasks concurrently, a feature primarily advantageous during the asset management phase rather than the construction stage. Formalisations of network theory, such as knowledge graphs constructed from an initial ontology as in Wagg et al. (2020) [22], and Petri nets, as suggested by Chiachio et al. (2022) [46], emerge as relevant tools for workflow development. This raises questions about how workflows can adapt over time within the evolving DT. Additionally, Pregnolato et al. (2022) [23] underscore the necessity of incorporating a workflow into a DT, presenting a framework consisting of five actionable sequence steps: data acquisition, digital modelling, sensor data transmission, data and models integration, and operation. Other research endeavours incorporate graphical workflows [49, 54, 58], however, additional supportive methods are required to ensure their effective implementation in real-world applications.

Decision Support and Autonomous Management

Many of the explored research studies mention decision support (DS), yet only a few provide mechanisms to effectively enable it as in [42, 50, 85]. In the context reviewed, DS is often provided through a graphical user interface (GUI), also known as a dashboard. This dashboard serves as a visual representation of the data collected, analysed, and synthesised by the DT. It may present performance indicators, metrics, trends, and other pertinent information in a comprehensible format. Decision makers can interact with the dashboard to monitor system status, identify anomalies, explore various scenarios, optimise system performance, make informed decisions, and mitigate risks, among other functionalities.

The dashboard acts as the intermediary connecting machines and humans, while also serving as a graphical interface of the physical computing infrastructure supporting the DT. This infrastructure is manifested in the review as a platform powered by an API web with HTTPS data transfer and utilising structured query language (SQL) databases for storing collected data, as noted by Damico et al. (2020) [24] and other sources [46]. The significance of databases is further emphasised by Wagg et al. (2020) [22], who propose knowledge graphs and ontology-driven databases as promising storage mechanisms to facilitate easy access to the DT's knowledge base. Additionally, Lu et al. (2020) [39] highlight the necessity of a platform for visualisation purposes, with the ultimate goal of the DT being to provide intuitive information deployment and decision support to users.

Autonomous management represents the pinnacle of the capability hierarchy proposed by Grieves (2006) [21], with only a few research efforts incorporating it [24, 36, 39]. However, these implementations still involve limited and narrowly-focused actions, such as triggering alarms or sending warning messages. Further research in this direction is needed, undoubtedly aided by the AI.

Artificial Intelligence, AI

AI plays a pivotal role in leveraging past data to facilitate modelling, control, and prediction tasks using statistical techniques, without requiring explicit programming, as employed in [76, 79]. Additionally, AI-based computer vision techniques are utilised for generating three-dimensional representations, while automated planning and scheduling algorithms are employed for optimisation purposes such in [35, 44, 82]. Moreover, Knowledge-based systems (KBS), a subset of AI, focus on autonomous decision making by leveraging existing knowledge, among other methodologies. AI is also employed by Ritto et al. (2021) [26] where a physics-based model is combined with a ML classifier to construct a DT that would be connected to the physical counterpart and support decisions.

AI can be used for conducting the diagnostics function within the DT by creating fast-running surrogates from a data-driven approach by ML methods. Al-Hijazeen et al. (2023) [68] relies on SHM sensing techniques and structural analysis to detect damage or degradation, all forming the basis for its DT implementation, with support from AI methods. This framework involves collecting structural and environmental data from the structure, facilitating the model design and calibration of the structural DT. The integration of DT and ML revolves around an information management and control platform grounded in ML, which combines physics models with ML techniques. Moreover, direct transmission of sensor information to ML algorithms facilitates model calibration and structural evaluation. This strategy adopts a notably inclusive hybrid model approach. However, while the emphasis is given to capture and store historical asset data to predict future behaviour, it overlooks real-time condition monitoring, workflow management, and uncertainty quantification. Another comprehensive and promising approach is made by Liu et al. (2024) [83] with a DT deployment for structural integrity management, including model updating, real-time simulation and data-driven forecasting with inference of model parameters and uncertainty quantification, although AI implementation is missing so advanced predictive capabilities and automated decision-making processes are handicapped.

2.3 Conclusions

In the course of the literature review, three distinct categories of DT representations were identified. The first category revolves around geometric digital representations of physical assets, chiefly employing the BIM methodology. This approach emphasises the creation of precise digital replicas, focusing on geometric accuracy and detailed physical characteristics of the asset. In contrast, the second category is centered on monitoring and appears as an evolution of SHM technology. These representations prioritise data collection to monitor the condition and performance of physical assets in real time. The third category comprises offline simulators, which provide simulation capabilities but lack connectivity to the real environment. While useful for scenario testing and analysis, these representations do not offer real-time functionalities. In addition to this, most of the reviewed procedures lack a management model that serves as a connecting system between physical and digital counterparts, capable of autonomously coordinating tasks within the DT, such as data collection, model updating, or internal hardware-in-the-loop control of sensors and actuators, among others.

In the first category, some studies particularly those involving BIM models [23, 32, 33, 37, 38, 54, 61, 64], aim to provide detailed geometric representations of the physical asset [70, 73] while occasionally enriching them with SHM data [27, 28, 34]. Furthermore, most of them emphasise the design and construction phases over the O&M, with a specific focus on monitoring during construction rather than fostering decision-making capabilities [30, 35, 45, 49, 60, 84].

The second category explored for DT representations emphasises monitoring and closely resembles advanced SHM methods [25, 29, 50–53, 59, 66, 81]. While they offer additional capabilities beyond conventional SHM deployments, such as diagnostics [55, 69] or uncertainty quantification [40, 48, 56–58, 62, 63, 65, 74, 78, 80, 83], they still lack a comprehensive management structure.

To the thrid category belong most of the research papers, which incorporate models for reproducing the behaviour of the systems. They resemble more like simulators rather than comprehensive DT developments, as they lack one or more key capabilities, such as diagnostic, prognostic, and decision-making functionalities, as well as workflow integration [16, 39, 41–44, 47, 67, 68, 71, 72, 75–77, 79, 82]. Nevertheless, a few of these studies [22, 26] delve into procedures and discussions regarding the dynamic updating of models based on data from their physical counterparts [31], or provide automated decision making [24, 36, 46, 85]. However, not all these methods implement AI or make use of surrogates, resulting in operations primarily performed offline rather than in real time as required by the DT.

In summary, while the approaches encountered in the literature contribute individually to draw the required functionalities, they fail to encompass all the key components into a fully functional single DT representation. These components are oriented to provide the capabilities required for a comprehensive DT deployment, including simulation, learning, and management, as illustrated in Figure 2.1, adapted from [46].

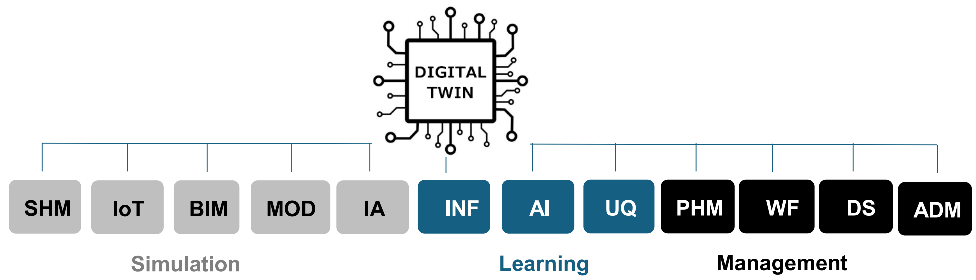


Figure 2.1: Main capabilities of a DT. Refer to Table 2.1 for acronyms definition.

As a conclusion, no single approach has emerged that integrates all the essential elements highlighted in the literature, namely: SHM, IoT, BIM, modelling, diagnostics, inference, AI, uncertainty quantification, prognostics, workflow models, decision support, and autonomous decision-making capabilities. Further research and efforts are needed to bridge this gap and develop comprehensive DT frameworks that effectively incorporate all these crucial components for practical application in structural civil engineering.

Chapter 3

Research objetives

The **general objective** of this research is to provide a conceptualisation of DT as a tool based on different components supported by state-of-the-art technologies, in order to effectively carry out the management of the O&M of civil infrastructures, as well as to develop the methodology that allows their integration and workflow. To that end, several hypotheses are formulated below, leading to more specific objectives:

1. The concept of a DT is well-established, but its definition and application can vary across industries. Generally, a DT refers to a virtual representation of a physical system or process, and it is used for monitoring, analysis, and simulation. While the fundamental notion is established, ongoing developments and advancements in technology may contribute to refining and expanding the definition. Different industries and researchers may also adapt the concept to their specific requirements. In the context of civil engineering, the technology is currently employed in silos and there is a pressing need for digitalisation to enhance efficiency and sustainability in this field. Moreover, the academic literature does not provide sufficiently detailed and comprehensive DT proofs of concept to clearly show the deployment of a DT for illustrative purposes (refer to the literature review in Chapter 2). While several studies focus on simulation and data assimilation, less emphasis has been placed on the coupling between monitoring and simulation together with decision making. In the face of this scenario, there is a crucial need for well-defined elucidation and practical examples to expedite the implementation of the DT paradigm in the civil engineering field. Clear definitions provide a shared understanding, reducing ambiguity and facilitating communication among stakeholders. Moreover, practical applications are of the foremost importance as they serve as tangible examples, demonstrating how the concept can be effectively utilised in real-world scenarios. This approach not only speeds up the learning curve but also fosters confidence and encourages support for the adoption of DT technology.

   Hypothesis 1: The DT paradigm requires a clear definition and support through illustrative practical applications to gain traction and achieve widespread adoption within the civil engineering field.

Research Objective 1: Provide a comprehensive conceptualisation of the DT paradigm in civil engineering, detailing its components. Implement various case studies that showcase substantial applications of the DT in practical scenarios. These case studies will serve as exemplars, promoting the widespread adoption of DT in civil engineering.

2. Models often involve simplifications or abstractions of the real-world system and data obtained from sensor measurements may be affected by various sources of inexactitude, including measurement errors, noise, and device inaccuracies, so a methodology to accurately quantify this uncertainty is needed. The Bayesian framework allows for the estimation of posterior probabilities, which represent the updated beliefs about the system given from models with the provision of the newly observed data. Moreover, it also enables the updating of model parameters in response to new evidence. The Bayesian framework appears essential in refining the DT in light of uncertain or imprecise information from data and models, providing a systematic way to update and enhance predictions and model parameters as new data becomes available.

   Hypothesis 2: The Bayesian approach can be incorporated into the DT framework to quantify uncertainty in both models and data for a more reliable risk assessment and informed decision-making process within the context of DT in civil engineering.

   Research Objective 2: Incorporate uncertainty quantification through the Bayesian approach into the DT workflow to make a risk-aware decision making.

3. The complex task of managing numerous interconnected components within DT developments for civil engineering systems, encompassing real-time monitoring data, supplementary data sources, analytics, and management, presents a significant challenge for orchestrating the DT workflow effectively. Ensuring seamless collaboration and information exchange among different DT components is crucial. Moreover, the dynamic nature of AECO requires the DT to swiftly adapt to changes, initiating updates in response to real-time data, adjustments in maintenance policies, or unforeseen events within dynamic event-based systems.

   Hypothesis 3: The implementation of a dedicated tool is crucial for orchestrating the workflow of DT implementations in civil engineering, guaranteeing efficient collaboration among diverse elements, streamlined management, and adaptability to changing conditions. Petri nets are a suitable method to manage the DT workflow in civil engineering applications.

   Research Objective 3: Develop a suitable frame based on Petri nets to manage the workflow of a DT, which accurately represents the dynamic and event-driven behaviour of the virtualised systems.

4. Implementing DT in civil engineering faces several challenges that span data, technology, and management aspects. Privacy and security concerns emerge due to the handling of monitoring data, and there are high associated costs linked to costly static data acquisition systems and overloaded traffic of large volumes of data. In addition, interoperability issues arise from integrating diverse data sources and technologies, exacerbated by a lack of standardisation. Furthermore, the scarcity of high-quality data for training DT models is also a substantial concern, especially relevant in the initial stages when monitoring data may be lacking. This issue can lead to reduced model accuracy, difficulties in capturing variability, and limited feature representation. Over time, as the DT accumulates monitoring data, the models can be continually refined to enhance their accuracy and effectiveness. However, the challenge associated with data persists, as there are high associated costs linked to data acquisition systems and overloaded traffic of large volumes of data, together with privacy and security concerns related to data handling.

   Hypothesis 4: The development of an AI-based methodology for generating data in quality and quantity for the effective training of the DT models can overcome the challenge of data scarcity, data privacy and security, interoperability and traffic overload.

   Research Objective 4: Elaborate an AI-based generative setting for supplying data in quality and quantity for the effective training of the DT models, thereby reducing costs and time, enhancing computing efficiency, complying with data format standards, and ensuring cybersecurity and privacy concerns. This setting should incorporate strategies such as synthetic data generation, data augmentation, and inclusion of domain expertise with physics guidance, enabling models to compute at the edge.

5. The complexity of civil engineering systems poses a challenge in developing models for the DT. Achieving a balance between accuracy and computational efficiency is essential, enabling real-time operation. However, despite providing a detailed depiction of reality, these models frequently exhibit high computational demands, require extensive and high-quality data, encounter scalability issues, lack interpretability, pose challenges in calibration and validation, are susceptible to overfitting, struggle with dynamic adaptation, entail high maintenance costs, and demand substantial resources and skills for upkeep and updates. In the structural domain of civil engineering, these models primarily concentrate on implementing a damage assessment strategy to forecast asset failure and enact effective maintenance policies. The damage assessment capability is paramount for a DT in civil engineering, as a proactive tool for identifying, evaluating, and managing potential structural damage or anomalies in infrastructure, reducing risk and costs, and increasing safety and efficiency.

   Hypothesis 5: The development of surrogate models, which are simplified representations of more complex models or systems, facilitates the accurate approximation of their behaviour in a computationally efficient manner. When designed to capture essential features and relationships within the data, these surrogate models can maintain efficiency while enabling real-time performance for the DT.

Research Objective 5: Create a damage assessment surrogate model deployment strategy suitable for integration into the DT.

Chapter 4

Completion of objectives

The main research objectives of this thesis have been presented previously in Chapter 3. The work undertaken towards the completion of such objectives, including the methodologies developed and the experiments carried out to demonstrate such methodologies, are briefly outlined in this chapter. Additionally, quotations indicating where detailed information can be located within this document are included in the text below. The majority of the research objectives have been already reviewed; some are published in paper journals and others are being revised or prepared to be sent, all listed in Appendix A.1.

- **Research Objective 1:** *Provide a comprehensive conceptualisation of the DT paradigm in civil engineering, detailing its components. Implement various case studies that showcase substantial applications of the DT in practical scenarios. These case studies will serve as exemplars, promoting the widespread adoption of DT in civil engineering.*

  In Chapter 6, a comprehensive conceptualisation of the DT is presented, encompassing both a mathematical expression and a computational formulation (Section 6.2). This chapter delves into various aspects, including the interdisciplinary nature of the DT (Section 6.1), its overarching purpose and objectives (Section 6.3), and the diverse data sources that contribute to its functioning (Section 6.5). Each component of the DT is meticulously discussed, accompanied by illustrative figures to facilitate comprehension and enhance clarity (Section 6.4).

  Furthermore, this thesis includes the development of two case studies that serve as practical applications of the DT in civil engineering. The first case study (Chapter 11) focuses on the DT of a 2D metal tower, highlighting the integration of technology supporting the DT and the utilisation of the Bayesian framework for solving the inverse problem, specifically inferring a magnitude of interest. The second case study (Chapter 12) implements the DT of a 3D tower of bigger scale and concentrates on the generative setting and the deployment of DT models for damage assessment. In both cases, meticulous attention has been given to implementing detailed DT workflow management.

- **Research Objective 2:** *Incorporate uncertainty quantification through the Bayesian approach into the DT workflow to make a risk-aware decision making.*

  To integrate a suitable tool for uncertainty quantification within the DT, a Bayesian framework has been adopted (Chapter 7). This framework has been applied in two key areas: first, in addressing the forward problem, which involves uncertainty propagation, as detailed in Section 7.1. Second, it has been utilised for tackling the inverse problem, which pertains to updating model parameters or inferring unknown quantities, as elaborated in Section 7.2.

- **Research Objective 3:** *Develop a suitable frame based on Petri nets to manage the workflow of a DT, which accurately represents the dynamic and event-driven behaviour of the virtualised systems.*

  In this thesis, the workflow management of the DT has been realised through the use of a Petri net, adopted for its suitability in handling complex and dynamic operation environments (Chapter 8). Section 8.1 elaborates on the description of the workflow of a DT, outlining its various stages and processes. Section 8.2 provides a comprehensive explanation of how this workflow is managed, with an illustrative example to enhance understanding. Through the implementation of a Petri net, this thesis ensures effective management of the DT's workflow, enabling efficient navigation through its intricate processes.

- **Research Objective 4:** *Elaborate an AI-based generative setting for supplying data in quality and quantity for the effective training of the DT models, thereby reducing costs and time, enhancing computing efficiency, complying with data format standards, and ensuring cybersecurity. This setting should incorporate strategies such as synthetic data generation, data augmentation, and the inclusion of domain expertise with physics guidance.*

  In Section 9 of this thesis, a generative setting aimed at effectively training DT models has been developed, employing state-of-the-art AI-based methodologies. The necessity for DT models to have access to high-quality and abundant data for their training is thoroughly explained in Section 9.2. Meanwhile, Section 9.3 provides a detailed description of the generative setting, outlining its components and processes. Furthermore, in Section 9.4, metrics for evaluating the performance of the models are introduced to ensure the efficacy of the proposed methods. Through these sections, this thesis aims to establish a robust framework for training DT models effectively, emphasising the importance of data quality and quantity, the intricacies of the generative setting, and the metrics to check its performance.

- **Research Objective 5:** *Create a damage assessment surrogate model deployment strategy suitable for integration into the DT.*

  One of the main competencies enabling the DT to facilitate decision making in civil engineering is the deployment of a damage assessment capability. This capability is realised in Chapter 10 of this thesis through a pipeline of surrogate models designed to conduct a four-level damage assessment in real time. The methodology for effectively implementing the models is elucidated in Section 10.1 whereas the functionalities of the models level by level are detailed in Section 10.2.

Subsequently, in alignment with the research objectives presented in this thesis, a comprehensive table is included below, summarising the attainment of each of them. This table serves to provide a clear overview of how each research goal has been achieved throughout the course of this study. By examining the completion status of each target, valuable insights can be gained into the scope and impact of the research conducted within this thesis.

Table 4.1: Summary of the completion of objectives in this PhD thesis

Hypothesis	**Research Objective**	**Theoretical Background**	**Contribution**	**Case Study**
1. The DT paradigm needs clear definition and practical examples to gain traction and widespread adoption in civil engineering, which currently lacks digitalisation and faces significant sustainability challenges.	1. Provide a clear definition and practical applications to accelerate the application of the DT in civil engineering.	Section 5.1: Mathematical foundation	Chapter 6: DT conceptualisation in Civil Engineering	Chapter 2: Literature Review; Chapter 11: DT of a 2D metal tower, section 11.2: Technology integration
2. The Bayesian approach can be incorporated into the DT to quantify uncertainty in both models and data for a more reliable risk assessment and informed decision making despite models simplifications and noisy sensor data.	2. Incorporate Bayesian uncertainty quantification into the DT workflow to make risk-aware decision making.	Section 5.2: Uncertainty quantification	Chapter 7: Incorporating UQ into the DT workflow	Chapter 11: DT of a 2D metal tower, section 11.3: Bayesian inference of unknown parameters
3. A dedicated tool is essential for orchestrating the DT workflow in civil engineering, coordinating real-time monitoring, data sources, analytics, and management, and adapting promptly to changes. Petri nets are well suited for representing and managing the dynamic, event-driven behaviour of such systems.	3. Formulate a suitable frame based on Petri nets to manage the workflow of the DT.	Section 5.3: Flow control with Petri nets	Chapter 8: DT workflow management	Chapter 11: DT of a 2D metal tower, section 11.2.3: Web-based integration platform, Workflow service; Chapter 12: DT of a 3D metal tower, section 12.5: Inclusion into the DT workflow
4. An AI-based approach for generating sufficient quality and quantity of data can address challenges like data scarcity, privacy, interoperability, and traffic overload, improving the training of the DT models.	4. Ellaborate an AI-based generative setting for supplying data in quality and quantity for the effective training of the DT models, tackling their data challenges.	Section 5.6: AI application via NNs	Chapter 9: Generative setting for training DT models	Chapter 12: DT of a 3D metal tower, section 12.3: Generative setting
5. Surrogate models simplify complex systems, enabling accurate approximation in a computationally efficient manner, ensuring real-time performance for the DT, despite model complexity, computational demands, data requirements, and scalability challenges.	5. Develop a damage assessment surrogate model deployment strategy tailored for seamless integration into the DT.	Section 5.4: Failure analysis, FTA and FMECA. Section 5.5: Damage Assessment	Chapter 10: Damage assessment models strategy for DTs	Chapter 12: DT of a 3D metal tower, section 12.4: Damage assessment pipeline of models

Chapter 5

Theorical fundamentals

This chapter aims to provide the theoretical background that supports the methodologies proposed in this thesis. Given the interdisciplinary nature of the DT, a significant range of topics are addressed. These include *Information* theory, which encompasses the *State Space* and the *Finite State Machines* theories sustaining the mathematical foundation of the DT. Furthermore, *Uncertainty Quantification* is discussed to enable risk-aware decision making within the DT framework. Control engineering principles, including the application of *Petri nets* for managing the DT workflow, are also examined. Additionally, the thesis delves into reliability engineering, focusing on *damage assessment* to prevent system failure. Methodologies like *Failure Tree Analysis* (FTA) and *Failure Modes, Effects, and Criticality Analysis* (FMECA) have been explored in this regard. Finally, the use of AI techniques, particularly DL methods, is considered to enable the creation of fast and efficient surrogate models for the DT.

5.1 Mathematical foundation

The mathematical foundation of a DT can be conceptualised as a hybrid integration of both the *State-space* theory [88], rooted in differential equations which capture the dynamics of the processes, and the theory of *Finite state machines* (FSMs) [89], which finds greater significance in safety-oriented applications [90]. This integration is based on the fundamental objective of achieving a rigorous representation of complex systems characterised by the coexistence of continuous and discrete behaviours.

By incorporating state-space models and FSMs, a DT captures the physics-based continuous dynamics of a system and its discrete control logic for decision making and event handling. This hybrid approach is highly valuable for modelling real-world complexity, ensuring fault handling and safety, enabling real-time control, ensuring adaptability, offering comprehensive system analysis, and promoting synchronisation between the digital model and the physical system. Furthermore, it is particularly beneficial for applications involving CPS and complex interactions between physical processes and digital control components.

The features resulting from the synergy between the state-space (continuous modelling) and the FSM (discrete modelling) theories within a DT context are multiple. Firstly, this hybrid approach enables adaptive control, where both continuous and discrete strategies can be employed based on real-time conditions. The state-space representation simplifies the FSM, as it allows for a more concise representation of continuous dynamics, thus enhancing the manageability of FSMs and reducing the number of finite states. Moreover, state-space improves accuracy by integrating realistic physics-based representations into the DT. On the other hand, the use of FSM allows for the explicit modelling of safety protocols, fault detection, and isolation mechanisms, enhancing the system's ability to respond to abnormal conditions and mitigate risks promptly.

Whitin the *State-space* theory, *system identification* can be regarded as a practical application aimed at constructing the state-space model that best fits the observed data, allowing to understand and predict the behaviour of a system. The system identification process involves collecting input-output data from a system and using techniques such as the *Time-domain* analysis [91], and estimating the model parameters. The outcome is a model that accurately represents the system's dynamics based on real-world measurements.

The mathematical formulation describes and models the states, behaviours, and evolution of a dynamic system in a digital environment, representing the virtual counterpart's state-variables, state-space, state-equations, and their relationships with inputs and outputs. This allows for the simulation and analysis of the physical system's behaviour in a digital environment.

Just like in FSMs, a DT can have *inputs and outputs* in this representation. Inputs represent external influences on the physical system, and outputs express measurements or observations of the system's behaviour. Besides, it is feasible to establish a set of *state-variables* that represent the state of the physical system at a given time. These state-variables can be continuous or discrete, and capture the system's behaviour, including factors like position, velocity, temperature, pressure, and other relevant attributes. In the context of a DT, the state-variables can represent the physical system's parameters. The state-variables collectively define the *state-space* of a system. The state-space represents all possible combinations of state-variable values that the system can assume. In a DT, the state-space captures the possible configurations and conditions of the physical system.

The *state-equations* describe how the state-variables change over time. These equations are typically differential equations that model the dynamics of the system. State-equations are fundamental for understanding how a system's state evolves in response to inputs, disturbances, and internal interactions, and in a DT context, state-equations can be used to simulate the behaviour of the physical system. In discrete systems addressed through FSMs, transitions can be defined as conditions or rules that dictate how state-variables change over time, rather than relying on differential equations or complex algorithms. These conditions can be thought of as transition rules similar to those employed in PNs. The *State-space* theory also considers the existence of *input-output relations*, which describe how external factors or control inputs affect the state variables and, in turn, influence the system's behaviour.

Consequently, a DT of a system described by a state-space representation can be expressed mathematically as follows:

$$\hat{s}(t) = m(u, e, \theta) \tag{5.1}$$

where u denotes the input variables, e is the environmental vector, θ represents the model parameters, $m()$ is the model or $state-equation$ and $\hat{s}(t)$ is the output or current state of the system captured by the DT.

As an illustrative example, the DT of a structure subjected to external forces (e.g. wind) is presented (Figure 5.1). Its dynamic response can be represented as the second-order differential equation of motion in its simplest form for a single-degree-of-freedom system:

$$m\ddot{\hat{d}} + c\dot{\hat{d}} + K\hat{d} = F(t) \tag{5.2}$$

where m is the mass, c the damping, and K the stiffness of the structure, $F(t)$ are the applied forces over time, and $\ddot{\hat{d}}$ the acceleration, $\dot{\hat{d}}$ the velocity, and $\hat{d}$ the displacement of the structure.

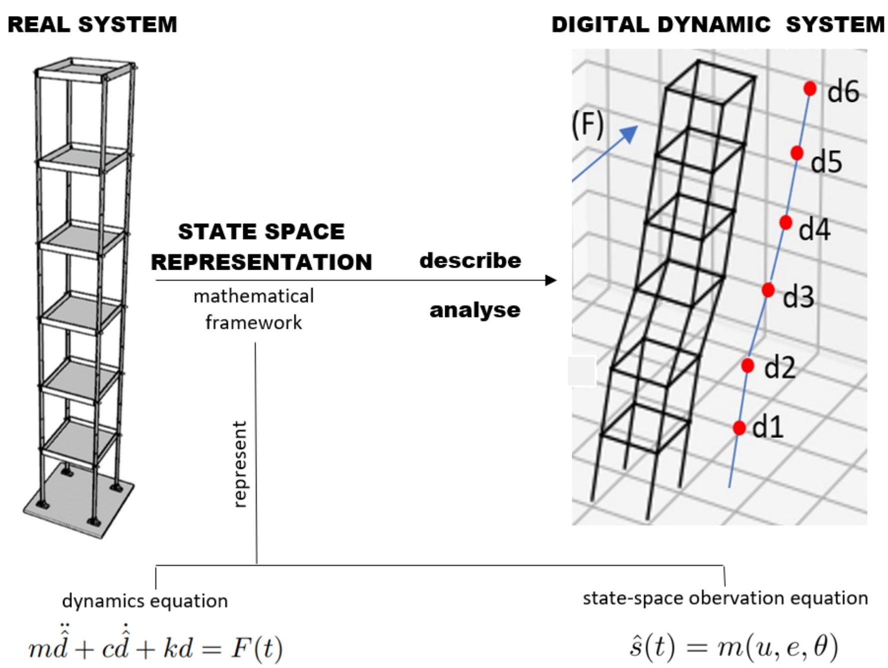


Figure 5.1: Illustrative example of a state-space representation.

The *output* refers to the response of interest, which in this case is the displacement of the structure denoted as $\hat{d}$. This variable provides information about how the tower responds to external forces and its dynamic behaviour over time. The *input* is the external force applied to the tower at time t and is represented as $F(t)$. The state-equation is then expressed as follows:

$$\hat{d}(t) = \frac{1}{K}F(t) - \frac{c}{K}\dot{\hat{d}}(t) - \frac{m}{K}\ddot{\hat{d}}(t) \tag{5.3}$$

System identification involves estimating the parameters m, c and K based on observed input-output data. This can be performed through different methods, such as least squares estimation, maximum likelihood, optimisation techniques or NNs, to cite any.

The *state-variables* in this example correspond to the acceleration, velocity, and displacement of the structure. These state-variables belong to the space of real numbers, defining the *state-space*. Assuming that direct measurements of the response of interest are accessible, the *input-output relation* can be expressed as $d(t) = \hat{d}(t)$, where at time t, the observed displacement $d(t)$ is equal to the modelled displacement $\hat{d}(t)$. This is contextualised in a deterministic approach, which assumes no errors. However, in real-world scenarios, discrepancies between observations and predictions are common and must be handled. This challenge is addressed through uncertainty quantification, as will be discussed in the following section.

5.2 Uncertainty quantification

Uncertainty quantification (UQ) involves the systematic evaluation and analysis of uncertainty within data, models, and predictions, encompassing factors related to variability and likelihood in inputs, outputs and model parameters. Its ultimate objective is to facilitate informed decision making by taking into account associated risks.

There are two primary classifications of uncertainty: aleatoric and epistemic [92]. Aleatoric uncertainty emerges from the inherent randomness in the system being modelled or measured, often called stochastic or random uncertainty. On the other hand, epistemic uncertainty stems from missing data or data scarcity, a lack of information regarding model parameters, or a limited understanding of the underlying physical processes of the system being modelled or measured. In contrast to aleatoric uncertainty, epistemic uncertainty can be reduced through additional data collection and improved modelling. Techniques like sensitivity analysis or Bayesian inference are commonly employed to quantify this form of uncertainty.

The essential elements of an uncertainty quantification are given next.

1. Uncertainty Sources: The process of addressing uncertainty starts with the identification of potential sources of uncertainty within a system or model. These sources may encompass input parameters, model assumptions, data inaccuracies, and external variables.

2. Uncertainty Representation: After identifying the sources, UQ proceeds by expressing uncertainty through means such as probability distributions, intervals, or other mathematical frameworks. Probability distributions, such as Gaussian (normal), uniform, or triangular distributions, are commonly used to model uncertainty.

3. Propagation of Uncertainty: UQ techniques aim to transfer uncertainty from uncertain inputs or model errors to the model outputs, and finally estimate the uncertainty in their predictions. Beyond input uncertainty, UQ also accounts for uncertainty related to model error, which may arise from

simplifications, assumptions, or approximations made in the modelling process. This process often involves mathematical techniques such as Monte Carlo simulations, polynomial chaos expansions, or sensitivity analysis.

4. Sensitivity Analysis: Evaluates how fluctuations in input parameters impact the variability of model outputs. It helps identify which input parameters contribute most to output uncertainty.

5. Model Selection and Validation: UQ encompasses the process of selecting the appropriate model structure and validating the selected model against empirical data, often considering the inherent uncertainty associated with the model itself.

6. Validation and Calibration: UQ considers the validation of models or simulations against observed data. Calibration involves the objective of fine-tuning model parameters in order to minimise the disparity between model predictions and actual observations, all while accounting for uncertainty in both the model and the data.

UQ is typically handled using probability distributions and statistical metrics to describe the behaviour and uncertainty associated with model's predictions, including the following main concepts given below.

- Probability Density Function (PDF): The PDF represents the probability distribution of a continuous random variable. For a variable x, the PDF is denoted as $p(x)$, and satisfies the following property: $\int p(x)\ dx = 1$ (where the integral is taken over the entire range of x)

- Expectation: The expectation ($\mathbb{E}$) or mean (μ), often denoted as $\mathbb{E}(X)$, represents the average value of a random variable X. For a continuous random variable, it is computed as: $\mathbb{E}(X) = \int xp(x)dx$

- Variance: Measures the spread or dispersion of a random variable X. It is denoted as σ^2 and is calculated as $\sigma(X) = \mathbb{E}[(X-\mu)^2]$. Covariance quantifies the extent to which two random variables X and Y vary together. It is denoted as $\Sigma(X, Y)$ and calculated as: $\Sigma(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$

- Standard Deviation: Denoted as σ, is the square root of the variance and provides a measure of how much the values of X deviate from its mean, expressed as: $\sigma = \sqrt{\sigma^2(X)}$

- Correlation coefficient: Often denoted as ρ, measures the linear relationship between two random variables X and Y. It is calculated as $\rho = \Sigma(X, Y)/(\sigma_X \cdot \sigma_Y)$.

- Conditional Probability: Expresses the probability $p(\cdot)$ of an event taking place after another event has already happened. It is denoted as $p(A|B)$, where A is the event of interest, and B signifies the conditioning event, and calculated as $p(A|B) = p(A \cap B)/p(B)$

- Bayes' Theorem: Bayes' theorem is a fundamental formula in Bayesian inference, connecting conditional probabilities $p(\cdot|\cdot)$. It is expressed as $p(A|B) = [p(B|A) \cdot p(A)]/p(B)$

5.3 Flow control with Petri nets

Petri nets (PN) are often considered an extension of FSMs, as they encompass functionalities enabling the representation of concurrency and synchronisation of events. This characteristic renders them highly effective in modelling the flow of complex systems characterised by concurrent behaviours. They are applied in safety and reliability engineering mainly for modelling workflows, information flow, or steps of batch processes.

A Petri net is defined as a graphical and mathematical modelling tool that allows the representation of systems, including their processes, events, and state transitions [93]. PNs are particularly effective in featuring discrete events and parallel processes that account for concurrency, which are commonly encountered in DT-managed scenarios. Thus, the dynamic nature of DTs reflecting changes over time based on online measurements is well-captured by the PNs. In addition to this, the flexibility to model transitions and state changes proves valuable in representing the evolving states that a DT undergoes. PNs also support formal analysis techniques and simulation. This capability facilitates the evaluation of the behaviour and performance of the system, aiding the calibration of the DT models towards the real-world asset. On top of this, stakeholders and other decision makers can comprehend the system's behaviour represented by a PN due to its expressiveness and intuitiveness. The resulting graphical representation is intuitive enough to make them accessible the asset's performance, fostering informed decision making based on these representations.

Places, *transitions*, and *arcs* are the main components of PNs, capturing the dynamic behaviour of the system and providing a structured representation of how abstract moving entities (tokens) move through that system. *Places* and *transitions* are connected by *arcs*, which indicate the direction of the connection. A *place*, graphically symbolised as a circle, represents a particular state of the system or activity being modelled (e.g. the current damage state of a structural component or an inspection activity in progress). *Places* are temporarily visited by *tokens*. The distribution of *tokens* over the PN at a specific execution time is referred to as *marking*, which is a vector whose components indicate the state of the PN. The *transitions*, represented by boxes, are responsible for the dynamic behaviour of the PN, and enable the system to move from one state to another [94]. *Arcs* are labelled with their corresponding *weights* (1 by default). Firing a *transition* according to a *firing rule* consumes tokens from input places and produces tokens in output places, representing changes in the marking.

Mathematically, a PN is defined as follows [95]:

$$\mathcal{PN} = (\mathrm{P}, \mathrm{T}, \mathrm{E}, \mathrm{M}_0, \mathrm{W}) \tag{5.4}$$

where:

$\mathrm{P} = \{p_1, p_2, \ldots, p_n\}$ (set of places),

$\mathrm{T} = \{t_1, t_2, \ldots, t_m\}$ (set of transitions), with $P \cap T = \varnothing$ (disjoint sets),

$\mathrm{E} \subseteq (P \times T) \cup (T \times P)$ (set of weighted arcs),

$M_0 : \mathrm{P} \to \mathbb{N}$ (initial marking function),

$\mathrm{W} : A \to \mathbb{N}$ (weight functions).

$\mathrm{A} \in \mathbb{N}^{n_t \times n_p}$ is the *incidence matrix* of the graph, whose elements are the result of subtracting the *backward incidence matrix* from the *forward incidence matrix*, thus $\mathrm{A} = \mathrm{A}^+ - \mathrm{A}^- = a_{ij}^+ - a_{ij}^-$, where $i = 1, \ldots, n_t$, $j = 1, \ldots, n_p$.

Given an initial marking of the net, namely M_0, the PN dynamics can be described through the change of the marking vector, which can be obtained by the following equation:

$$\mathrm{M}_{k+1} = \mathrm{M}_k + \mathrm{A}^T \mathrm{u}_k \tag{5.5}$$

where u_k is the *firing vector* at execution time k, a n_t-dimensional vector of Boolean values. If transition t_i is activated at state k, then $u_{i,k} = \mathbb{I}_i$ according to the *firing rule*, where $\mathbb{I}_i$ takes the value of 1 if $M_k(j) \geqslant a_{ij}^- \ \forall p_j \in {}^{\bullet}\mathrm{P}_{t_i}$, or 0 otherwise. In the last equation, $M_k(j) \in \mathbb{N}$ is the marking at state k for place p_j, and ${}^{\bullet}\mathrm{P}_{t_i}$ denotes the set of places that belong to the preset of transition $t_i, i = 1, \ldots, n_t$.

Different types of PN can be found in the literature, including the High-Level Petri nets (HLPN) [96], which incorporate logical and mathematical operators within the net elements (nodes, arcs, and rules for transition firing). The HLPN allow for higher complexity in the modelling of systems dynamics as well as the analyses of logic flows in a more versatile manner.

HLPNs make it possible to add an additional *transition condition* to the firing rule, integrating certain requirements based on DT variables. These variables can include the model's inputs or outputs, environmental parameters, etc. In mathematical terms, the HLPN can be described as:

$$\mathcal{HLPN} = (\mathrm{P}, \mathrm{T}, \mathrm{E}, \mathrm{M}_0, \mathrm{W}, \mathcal{C}) \tag{5.6}$$

where $\mathcal{C}$ is the set of conditions associated with transitions, and the resulting firing rule can be expressed as $u_{i,k} = \mathbb{I}_i \cdot \mathbb{I}_{\mathcal{C}_i}$, where:

$$\mathbb{I}_{\mathcal{C}_i} = \begin{cases} 1, & \text{if} \quad \mathcal{C}_i = \text{True} \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

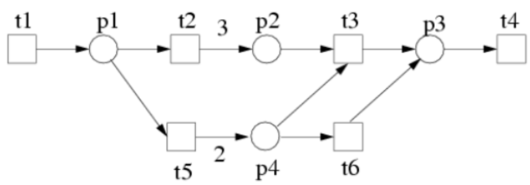
The variable $\mathcal{C}_i$ can be a boolean variable ($Bool \to \{\text{True}, \text{False}\}$) and, therefore, a transition t_i is fired only if it is enabled ($\mathbb{I}_i = 1$) and the transition condition $\mathcal{C}_i$ is true, as adopted in this work. When represented graphically in an HLPN, the transition conditions $\mathcal{C}_i$ are shown within the transition box. If there is no transition condition, then $\mathbb{I}_{\mathcal{C}_i} = 1$.

The following rules resume the algebra of the HLPN followed in the present development:

- Transitions always consume tokens from all the input arcs at the same time, and produce the same number of tokens to all out-coming arcs;

- Transition t_i is enabled if every input place p_j from its preset ${}^{\bullet}\mathrm{P}_{t_i}$ is marked with at least a_{ij}^{-} tokens;
- An enabled transition t_i will fire if the transition condition $\mathcal{C}_i$ is true;
- After firing, transition t_i removes a_{ij}^{-} tokens from p_j, and adds a_{ij}^{+} tokens to each j-th output place of t_i.

In summary, the dynamics of an HLPN are governed by the firing of the high-level transitions, along with the availability of the required number of tokens in their input places, as determined by the weight function. Firing a transition results in a new marking (token distribution) and the evolution of the marking can be expressed through the incidence matrix, as depicted in the following illustrative example (Figure 5.2) adapted from [97]:



incidence matrix A

	t1	t2	t3	t4	t5	t6
p1	+1	−1	0	0	−1	0
p2	0	+3	−1	0	0	0
p3	0	0	+1	−1	0	+1
p4	0	0	−1	0	+2	−1

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & -1 & 0 \\ 1 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & 2 & -1 \end{bmatrix}$$

Figure 5.2: Illustrative Example of a simple PN with 6 transitions ($t1$ to $t6$) and 4 places ($p1$ to $p4$). On the right is the incident matrix (A) related to transitions and places and its mathematical expression.

5.4 Failure analysis: FTA and FMECA

Fault Tree Analysis (FTA) and *Failure Modes, Effects, and Criticality Analysis* (FMECA) are two commonly implemented failure analysis methodologies which amplify the digital representation of the physical asset, identifying critical components, potential failure modes and their associated consequences, and allowing the DT to formulate effective maintenance strategies. Its integration into the DT also provides a comprehensive framework for risk mitigation and informed decision making in operations and maintenance strategies. Maintenance policies derived from FTA and FMECA include corrective, preventive and predictive measures tailored to address specific failure scenarios. The DT integrates this information with real-time data from sensors to continuously assess the condition of critical components. This dynamic monitoring enables the DT to adapt maintenance schedules and strategies based on the evolving health and performance of the physical system.

By providing information about potential failure modes and their criticality, FTA and FMECA empower the DT to optimise maintenance interventions. This ensures that resources are allocated efficiently, minimising downtime, reducing the likelihood of unexpected failures, and ultimately enhancing the overall reliability and performance of the system. The sequence is as follows: first, FTA is employed to identify potential failure modes and establish their logical relationships, providing a foundational understanding of the system vulnerabilities. Building on the insights gained from FTA, FMECA prioritises these failure modes according to their criticality, directing focus towards components that are more susceptible to damage.

In FTA, a logical tree-like structure is employed to model and analyse different events and their combinations that could lead to a particular undesired outcome. This dynamic approach allows the DT to simulate and assess the impact of changing conditions or events on the overall system reliability. The resulting fault tree serves as a blueprint for understanding the pathways leading to system failures.

FMECA, on the other hand, involves a comprehensive examination of failure modes, their effects, and criticality levels. FMECA builds upon FTA by evaluating the consequences of identified failure modes. Criticality assessments guide attention to components with heightened vulnerability, facilitating targeted monitoring efforts. The dynamic response in this context is manifested through continuous updates based on real-time monitoring. As the DT receives live data from sensors placed on critical components, FMECA dynamically adapts its analysis, identifying evolving failure modes and adjusting criticality assessments accordingly. In summary, FMECA informs the DT by mapping critical components, simulating failure modes in *what-if* scenarios, and optimising sensor placement based on its insights.

The Risk Priority Number (RPN) is a key component of the FMECA process, serving as a quantitative metric to prioritise and rank the identified failure modes [98]. The RPN is calculated by multiplying three factors:

$$RPN = S \cdot O \cdot D \tag{5.8}$$

- Severity (S): This factor assesses the potential consequences or impact of a specific failure mode. It assigns a numerical value to the severity of the failure, typically on a scale from 1 to 10, with higher values indicating more severe consequences. Severity takes into account the potential harm or damage that could result from the failure.
- Occurrence (O): Occurrence quantifies the likelihood or probability of a specific failure mode occurring. It also uses a numerical scale, typically ranging from 1 to 10, with higher values indicating a higher likelihood of occurrence. Occurrence considers the frequency or probability of the failure happening under normal operating conditions.
- Detection (D): Detection assesses the ability to detect the failure mode before it leads to a critical consequence. Like Severity and Occurrence, it is assigned a numerical value, usually from 1 to 10, with higher values indicating a lower likelihood of detection. Detection considers the effectiveness of existing detection methods, such as inspections or monitoring.

Following this, illustrations of a fundamental FTA and FMECA for a metal tower are provided (Figures 5.3 and 5.4). In a more comprehensive analysis, each of these elementary events could be extended to encompass precise categories of welding defects, corrosion mechanisms, load scenarios, etc. Moreover, probabilities and other variables can be integrated into the analysis to evaluate the overall reliability and safety of the structure.

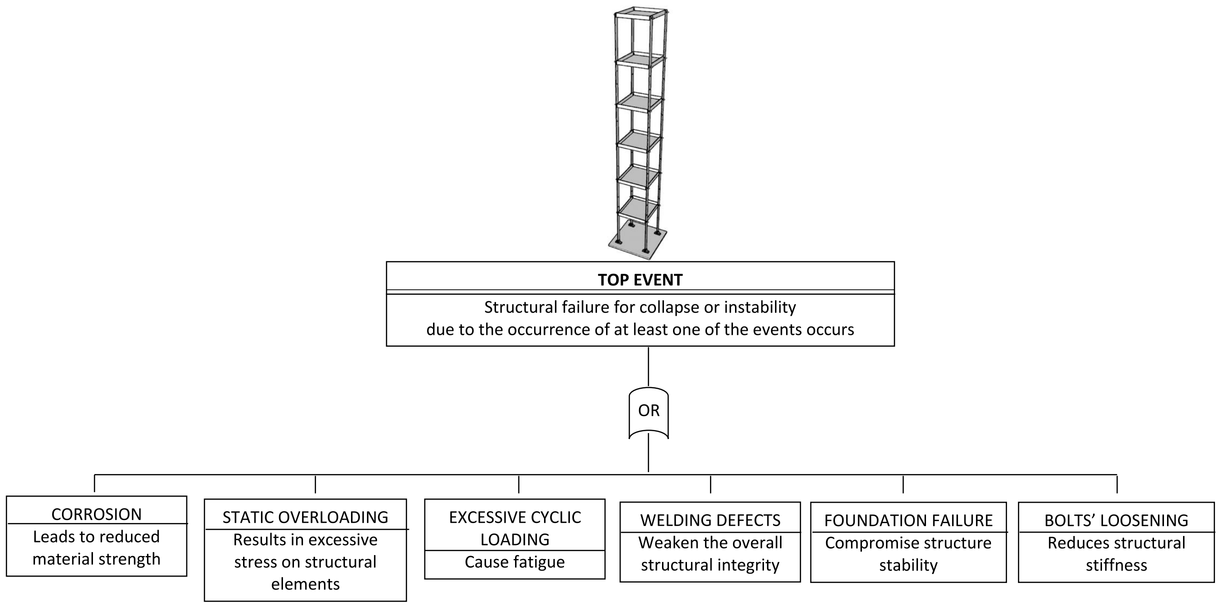


Figure 5.3: Example of a basic Fault Tree Analysis (FTA) of a metal tower. Note the bolt's loosening mode of failure (the element depicted at the extreme right side of the figure), relevant for the case studies in Chapters 11 and 12.

FMECA (Failure Modes, Effects, and Criticality Analysis)									
ITEM	FAILURE MODE	EFFECT	CAUSE	SEVERITY (S)	OCCURRENCE (O)	DETECTION (D)	RPN	BASIC MAINTENANCE MEASURES	STRUCTURAL HEALTH MONITORING THROUGH SENSORS
Steel structural elements	Collapse	Loss of Sectional Area	Corrosion	5	4	3	60	Application of protective coatings	Corrosion sensors (electrochemical, acoustic, electromagnetic …)
Steel structural elements	Collapse	Insufficient strength	Static overloading from traffic, weather, impact	4	3	7	84	Load testing, weight restrictions on traffic, impact-resistant barriers	Strain gauges, load cells, pressure sensors, displacement sensors …
Steel structural elements	Collapse	Fatigue	Vibration, cycling loading	4	2	8	64	Conduct dynamic load assessment and vibration analysis to avoid resonance	Accelerometers, acoustic emission sensors, ultrasonic sensors…
Welded joints	Collapse	Insufficient strength	Welding defects	5	3	6	90	Regular inspections	Radiographic testing, Magnetic Particle testing, Infrared thermography,...
Foundation	Instability	Tilt of the structure	Foundation settlement	4	3	7	84	Settlement monitoring systems and geotechnical measures such as stabilisation, drainage…	Inclinometers, settlement Plates, GNSS (Global Navigation Satellite System) …
Bolted joints	Instability	Loss of stiffness	Bolt's loosening	6	2	8	96	Regular Inspections and tightening and torquing	Displacement sensors, accelerometers, ultrasonic bolt load monitoring…

Figure 5.4: Example of a basic Failure Modes, Effects, and Criticality Analysis (FMECA) of a metal tower. Note the bolted joints component (the last file depicted on the table) relevant for the case studies in Chapters 11 and 12, which scores the highest in the RPN.

5.5 Damage assessment

Damage is often a precursor or leading indicator of potential failure. In many engineering and structural contexts, the monitoring and assessment of damage is essential for predicting and preventing failure, enabling maintenance or repair actions to mitigate the risk of failure. Therefore, while failure represents the ultimate undesirable outcome, damage assessment is a key process for managing and mitigating the risk of failure in systems and structures. FTA and FMECA set the foundation by identifying potential failure modes and critical components within a system, while damage assessment validates these predictions in a real-world context.

Within the civil engineering domain, damage assessment stands out as one of the main capabilities of a DT, serving as a valuable tool for accurate safety and risk planning. This methodical approach involves evaluating the condition and integrity of a structure and identifying any damage or deterioration, which in turn empowers the development of a well-tailored maintenance plan to ensure the long-term health and efficiency of the systems involved. As a result, scheduling asset operations for infrastructures is improved. Furthermore, the damage assessment capability empowers proactive measures through real-time informed decisions whenever it is performed online and automatically, leading to enhanced security and productivity. For these reasons, a robust damage assessment is essential to guarantee the reliability, functionality, and longevity of structures such as buildings, bridges, dams, and other infrastructures. Frameworks such as SHM and DT support the continuous monitoring, early detection, and prediction of damage, allowing for timely interventions and effective maintenance strategies, thus contributing to state-of-the-art asset management practices in civil engineering.

5.5.1 Damage definition

Damage is typically defined as any change to the material or geometry that can alter the structural properties or the response of the structure, thus adversely affecting the current or future physical integrity, functionality, or performance of the system [99]. This can encompass a wide range of adverse conditions that may compromise the structural safety, stability, and overall system operational effectiveness.

Within a structure, damage can manifest in various ways, including material degradation leading to weakened structural components, cracking and fracturing which can propagate over time decreasing the load-bearing capacity, and displacements or shifts in the structure's components affecting its stability and alignment. Different types of sensors and monitoring techniques are used to detect and quantify the various manifestations of damage, many of them being integrable with IoT to enable remote real-time monitoring for data transmission.

5.5.2 Damage assessment approach

Damage assessment is the comprehensive evaluation of the damage in a structure at a particular point in time or state. In the past, damage was only assessed when occurred (*corrective maintenance*) or during periodical inspections (*predefined maintenance*) either carried out using non-destructive evaluation (NDE) or visual observations. These techniques were susceptible to subjectivity and human errors, suffering from prolonged labour time, technical knowledge requested and difficulties in reaching distant or dangerous zones. *Condition-based maintenance* performed by techniques such as SHM emerged to face these challenges making use of recent developments like remote IoT-based sensing and AI to cite any, enabling pro-active maintenance. This means that damage can be predicted before it occurs (*predictive maintenance*) and actions are undertaken to be in place for the avoidance, mitigation or full correction of the failure (Figure 5.5). *Prescriptive maintenance* extends beyond predictive maintenance by not only forecasting failures but also prescribing specific actions to prevent those failures. This can include recommendations for maintenance tasks, repair procedures, and even automated responses designed to address issues proactively, preventing any disruption to operations.

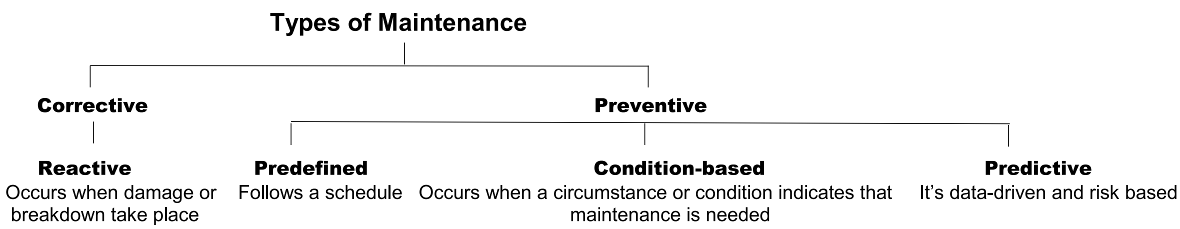


Figure 5.5: Types of maintenance policies for engineering assets

For the design of an accurate damage assessment strategy, an in-depth observation of the system is of foremost importance. An analysis combining FTA and FMECA is crucial for efficiently planning the monitoring, data collection and model provision for the deployment of the damage assessment plan of action [100]. Both techniques are highly valuable tools in risk assessment and management, helping to determine the damage typologies that may impact the system, identify the most vulnerable structural components or locations, and allowing to formulate the monitoring strategy for measuring the parameters implied, considering actions (e.g. temperature, humidity, loads, traffic, etc.) and reactions (e.g. vibrations, stress, strain, displacements).

In classical approaches, also known as *model-driven*, to assess damage it was necessary to previously build full physical models to accurately represent the behaviour of the structure. The common practice was to have a FE model of the structure as a baseline and compare it with new measurements for noticing changes in the pattern. Model updating is performed afterwards, replacing the initial parameters with the measured values. Consequently, further updating of the model can identify the damage by considering the structural changes [101]. One drawback of this approach is that full physical models can be complex and computationally intensive, requiring significant expertise to develop and operate.

Lately, with advancements in SHM and data-driven techniques, the focus has shifted towards more efficient and data-centric approaches. One such approach is known as model-free or *data-driven* damage detection, where the emphasis is on using measured sensor data to assess damage without relying on detailed structural models, making the process faster without compromising accuracy in controlled environments. This procedure, built over AI-based machine learning (ML) and recently more advanced DL techniques, has several advantages, as reduces the computational burden, can be applied to structures where modelling is challenging or not feasible, and easily adapts to changes in the structure geometry or behaviour over time. ML approaches for damage detection can be classified as unsupervised or supervised.

Unsupervised approaches are the most employed for damage diagnosis, including methods such as anomaly detection, clustering, and DL autoencoders for feature learning and dimensionality reduction, necessary when dealing with complex data [102]. Unsupervised ML for damage detection is only trained on healthy data and no labels are required.

On the contrary, supervised ML approaches are trained in both healthy and unhealthy labelled data and unlike the basic unsupervised approach, they are able to not only automatically detect if the system is healthy or damaged but also to determine other characteristics of the damage, like the damage type, status and position [101]. The supervised approach is based on a supervised classification, which can be binary, multiclass or multilabel, in order of increasing complexity and information provision.

In addition to both *model-driven* and *data-driven* methods, *hybrid approaches* combine experimental data with physics models to leverage the strengths of both methodologies. The data-driven techniques can capture complex and nonlinear relationships that may be challenging to model using traditional physics models. On the other hand, the physics-model approaches provide valuable constraints and insights, making it possible to extrapolate solutions to unexplored domains and enhance the interpretability of the predictions. The synergy of both approaches results in more robust, accurate, and versatile solutions, particularly in scenarios where either one approach alone might be insufficient or challenging to implement. Examples of hybrid models include physics-informed neural networks (PINNs) where NNs are combined with physical equations to improve predictions, and generative models that blend outputs from numerical simulations with observational data to improve model accuracy.

The degree of knowledge about a damaged state is commonly evaluated using Rytter's Hierarchy [103]. This hierarchical framing provides a structured approach to comprehensively assess damage within a system, encompassing stages from initial detection to informed prediction, consisting of the following four levels:

1) First Level: Damage detection,

2) Second Level: Damage location,

3) Third Level: Damage extent,

4) Fourth Level: Damage prediction.

The list above can be enlarged with an intermediate step between points 2) and 3), including the definition of the type of damage for the cases in which several kinds of damage modes can concur [104].

Hence through such means, damage assessment incorporates both diagnostics (damage detection, identification, extent and type) and prognostics (prediction of the future behaviour of the damage and the remaining useful life (RUL) of the asset) in a comprehensive conception that enables the DT to make informed decisions concerning its structural integrity, operations and maintenance.

Damage detection (Level 1)

To detect the damage, the undamaged state of the structure must be identified. This undamaged state serves as a reference line for comparison, enabling the identification of deviations or anomalies, and signalling the presence of damage. In this context, damage detection can be regarded as a *pattern recognition* problem, the most straightforward of the four Levels [105].

In SHM, damage detection methods can be broadly categorised into two main groups: *dynamic* (vibration-based) methods and *static* (non-vibration-based) methods. Each category presents its own set of techniques, the most popular one the *vibration analysis*, which includes methods like modal analysis, frequency response, and wavelet analysis. Static methods are less commonly employed compared to dynamic ones, as they are said to be less sensitive to small damages and specific to a particular structure and damage type. These disadvantages can be overcome with a good SHM strategy after the corresponding FTA-FMECA analysis. Besides, the static methods present valuable advantages such as the use of simpler instrumentation, the absence of preprocessing or complex analysis to identify abnormal behaviour, they are less affected by environmental influences and their operative mode makes possible the computation in the edge, enabling real-time responsiveness.

Damage detection can be undertaken by ML models, along with statistical and numerical methods [106], the former being less complex, faster and more suitable for the DT real-time requirements. Within the ML scope, unsupervised approaches are the most employed to damage diagnostics [102] such as anomaly detection, clustering, and DL autoencoders for feature learning and dimensionality reduction, necessary when dealing with complex data. Nevertheless, when the level of complexity increases towards the prognostics of the damage (Figure 5.6, taken from [101]), supervised approaches become necessary to overcome the challenge.

Damage location (level 2)

Once damage is detected, it is important to precisely localise which point or component of the system is affected in order to make the best-informed decision concerning the safety of the structure. The damage at one location could be affecting the monitoring measurements at other locations, even when those areas

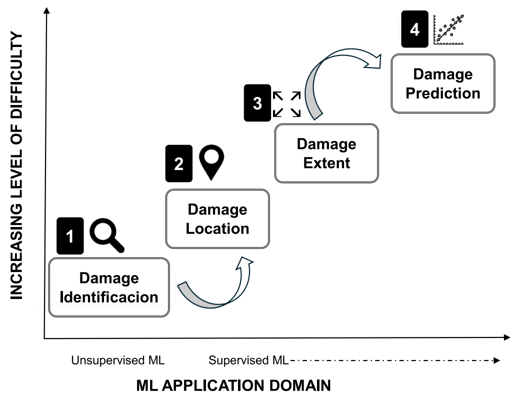


Figure 5.6: Four-level hierarchical damage assessment scheme.

remain pristine. Consequently, the complexity of the task at hand has increased from Level 1. Again, a good SHM strategy in designing the allocation of the sensors along with a high-fidelity characterisation of the behaviour of the structure at these critical locations will be crucial for the successful development of the Level 2 in damage assessment.

Unsupervised and supervised ML methods are employed for damage location, the choice depending on several factors, the principal the availability of labelled data. Unsupervised methods do not rely on prior knowledge of damage locations and can discover novel or unexpected damage patterns that may not have been observed during training, although interpretability can be challenging and often results are not sufficiently precise. On the other side, supervised methods provide accurate and definite damage localisation resulting in better-informed decisions when trained with high-quality data, which is not always available but can be generated. It is also well noted that regular updating is needed because the model's performance may degrade when faced with damage patterns that significantly differ from the training data.

Damage extent (level 3)

Generally, data-driven methods can only reach Level 1 (damage detection) and Level 2 (damage location). In contrast, hybrid methods combining sensor outputs and models enable an expanded scope of damage assessment until meeting Levels 3 (damage extent) and 4 (damage prediction) [101]. As previously mentioned, the usual starting point of hybrid approaches is to consider a physical model (generally solved by FE) to represent the behaviour of the structure. Model parameters associated with damage phenomena such as stiffness, are sought to be monitored while ensuring the compatibility between the model and the measurements. This approach also allows for the quantification of the structural damage and the prognosis of its RUL if a time-dependent damage equation is involved.

Damage severity quantification can be treated as a classification problem when categories covering the damage states are detailed, or as a regression task when calculating numerical indexes. Thus, discrete damage severity categories can range from 'low' to 'moderate' and 'severe', for example, whereas a generic damage severity parameter θ can vary between 0 (healthy) and 1 (fully damaged). The latter approach is commonly used in SHM to assess the extent of damage in structures [107]; a severity damage index (SDI) is calculated (5.9) as a percentage or a ratio of the loss in a structural value parameter affected by the damage, such as stiffness:

$$SDI = (I_v - C_v)/I_v \tag{5.9}$$

where I_v is the value of the parameter before any damage occurs and C_v is the value of the parameter after damage has been detected. The SDI provides a quantifiable measurement of damage severity that can be converted into a damage severity classification by categorising the extent of damage into different severity levels, like *Healthy* if SDI < Threshold 1, *Moderate Damage* if Threshold 1 ≤ SDI < Threshold 2 and *Severe Damage* if SDI ≥ Threshold 2.

Supervised ML techniques automate both classification and regression tasks enabling the real-time assessment of damage severity within the DT framework. This automation allows an immediate response once damage is detected, located and quantified, resulting from informed decision making.

Damage prediction (level 4)

Damage prediction entails forecasting the future condition or severity of damage by assessing the current state of a system, anticipating the forthcoming environments and estimating the remaining useful life (RUL) of that system through simulation and/or past experience. In short, predicting the damage aims to estimate how much deterioration a structure will accumulate over a specified time frame or under certain conditions [86]. The *Damage prediction* and the *severity of damage* are related concepts when damage is time-dependent and the damage accumulation rate remains constant over time. In this case (5.10):

$$SDI(t) = I_v - D_r \cdot RUL(t) \tag{5.10}$$

where $SDI(t)$ is the time-dependent structural damage index, I_v comes from (5.9) and D_r expresses the decrease of the parameter value with time. The RUL is often expressed as a continuous variable such as the number of operating hours, days, years or cycles before the end of life.

In practice, the relationship between the SDI and the RUL can be more complex since the real-world structural damage may be influenced by several factors, and models need to be constantly updated to reflect the dynamics of the changing conditions.

Predicting the RUL is typically framed as a supervised ML regression task where data including historical records of a component or system with known RUL values are available or generated. Besides, if criteria or conditions indicating the end of the structure's useful life are established, it is possible to simulate the behaviour of the structure under different scenarios and relate the responses with the RUL, relying on factors such as the critical level of damage and the time-dependent performance degradation [108].

5.6 Artificial intelligence application via neural networks

In the context of a DT, AI encompasses a broad spectrum of functionalities, most of which are implemented through the use of NNs. NNs play a pivotal role in the realm of DT providing a robust foundation for data-driven modeling and decision-making processes. Their proficiency in pattern recognition, capacity for learning from dynamic datasets, and scalability, contribute significantly to the accurate representation of complex real-world systems within a digital framework. Their real-time processing capabilities make them invaluable for applications requiring timely responses as surrogate models. NNs excel in predictive analytics, enabling the anticipation of future events based on historical data. Additionally, they contribute to anomaly detection, optimisation of processes, and decision support, enhancing the overall efficacy of DT.

5.6.1 Neural networks

An artificial neural network (NN) is a mathematical model that simulates the structure and functionalities of biological NNs [109]. A basic building block of a NN is an artificial neuron, that is, a simple function with three simple sets of rules: multiplication, summation and activation. At the entrance of the neuron, the inputs are weighted, which means that every input value is multiplied by an individual weight. In the middle section, it is the sum function that adds all weighted inputs and biases. At the exit, the sum of the previously weighted inputs and bias is passed trough an activation function.

For the NN model to meet its objective, it is necessary to generate the parameters w_i for each layer and b_j for each neuron. The process to obtain these parameters is referred to as *training*, and it can be broadly categorised as either supervised or unsupervised. The loss function, also referred to as objective function, is a crucial component that guides the learning process. It measures how well the model's predictions align with the actual target values, quantifying this difference and serving as a measure of the model's performance. The goal of the training is to minimise this loss by adjusting the internal parameters, mainly weights and biases, through a process such as backpropagation [110], aiming to find the parameter values that result in the smallest possible loss. The choice of a specific loss function depends on the nature of the task, such as mean squared error for regression or categorical cross-entropy for classification. In summary, an artificial NN consists of a number of artificial neurons (i.e., nonlinear processing units) which are connected to each other via weights, and "learn" a task by adjusting its parameters.

Mathematically, the behaviour of a neuron can be represented by Figure 5.7 and the following equation (5.11):

$$\hat{y}_i = \sigma\left(\sum_j W_j x_j + b_i\right) \tag{5.11}$$

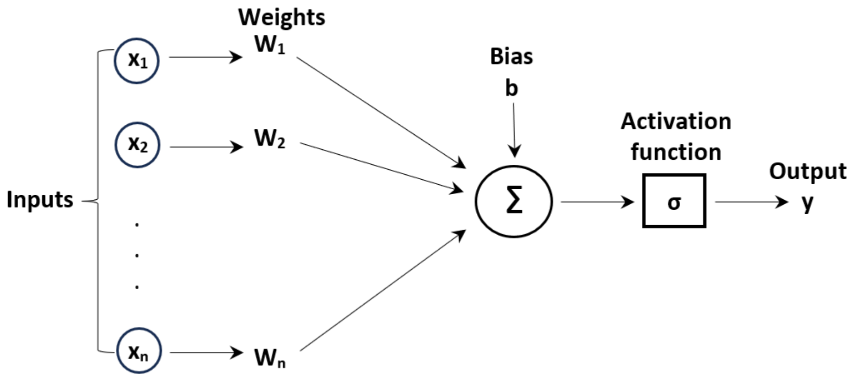


Figure 5.7: Architecture of an artificial neuron.

where $\hat{y}_i$ is the output of the model, x_j are each of the j inputs of the neuron with their corresponding weights W_j, b_i are the bias associated with the neuron i, and $\sigma()$ is the activation function of that neuron. The activation function is chosen on the basis of the problem to solve, such as sigmoid for binary classification, softmax for multiclass classification, and linear for regression.

Various types of NN are available and their classification depends on factors such as the architecture, data flow, types of neurons employed, density of neurons, layers and their depth, activation functions, etc. Exemplary types include Dense or Feed Forward (FNN) Neural Networks [111], Convolutional Neural Networks (CNN) [112], Recurrent Neural Networks (RNN) [113], Transformers [114], Autoencoders (AE) [115], and Generative Adversarial Networks (GAN) [116].

5.6.2 Physics informed neural networks

Physics-Informed Neural Networks (PINNs) is a designation for a category of ML models that leverage NNs to incorporate physical laws, constraints or knowledge in the form of data into the learning process [117]. This hybrid approach emphasises the synergy between data-driven learning and physics-based insights, as it not only captures patterns from data but also incorporates essential principles from physics into the NN's structure and training.

There are different methods for embedding the physics of a system in a NN [118]. For systems with simple and well-defined physics, embedding governing equations in the loss function ensures adherence to physical principles during training. In this way, when the physics equations are integrated into the loss function of the NN, the model is ensured not only to fit the training data but also to respect the governing physics of the system. Other ways of integrating physics-based information involve incorporating additional inputs that encode relevant physical parameters or constraints into the input layer of the NN. In addition to this, physics-based information can be also integrated into the output layer. In this case, the NN is guided during training to generate outputs that align with the expected physical behaviour.

Physics information through the loss function

In straightforward problems or systems with well-defined relationships between inputs, outputs, and the underlying physics, it is feasible to incorporate domain-specific knowledge and physical principles by embedding the *learning of physics through the loss function.* In this case, the expression of the physics-informed loss function is given by (5.12):

$$L_{\text{PINN}} = \gamma \cdot L(\hat{y}, y) + \mu \cdot L_{\text{physics}}(\hat{y}, y) + \lambda \cdot L_{\text{reg}} \tag{5.12}$$

where:

L represents the loss obtained from the data,

L_{physics} denotes the loss obtained from the physics equations or constraints,

L_{reg} corresponds to the loss arising from the regularisation term, which prevents overfitting and enforces specific smoothness requirements within the L_{PINN}, such as continuity and derivability,

$\hat{y}$ is the model output or model prediction,

y is the real-data output or target output,

γ is a hyperparameter for balancing the *data* component of the loss,

μ is a hyperparameter for balancing the *physics* component of the loss,

λ is a hyperparameter for balancing the *regularisation* component of the loss.

Physics information through the inputs

Conversely, complex systems are characterised by their intricate, often nonlinear interactions, which can involve numerous variables and parameters. Trying to capture the full complexity of the underlying physics through a loss function in such cases can be both challenging and error-prone. This difficulty arises from the complexity of accurately encoding all the parameters and interdependencies that govern the system's behaviour within the constraints of a loss function. A more practical and effective approach, particularly in the context of these intricate systems, is to convey the relevant physics to the NN indirectly, using the input and output data. This approach capitalises on the NN's innate ability to learn complex patterns and relationships directly from data.

By providing the network with input data that contains essential physical parameters or properties and encouraging it to generate output data that aligns with established physical principles, the NNs are able to understand and represent intricate physical behaviours without the need for explicit specification. In this case, the results generated by physics-based equations or models are introduced as additional inputs to the NN, which becomes physics-informed, contributing to the learning process. Furthermore, during the training phase, the NN's weights and biases adapt based on the physics-encoded inputs. This adjustment ensures that the NN's predictions align with the observed data. As a result, the forward pass of the network incorporates the principles of *physics through the input neurons*, as illustrated by the following equation (5.13):

$$\hat{y}_i = \sigma\left(\sum_j W_j(x_j + \sum_p W_{pj} x_{physics\ p} + b_j) + b_i\right) \tag{5.13}$$

where:

$\hat{y}$ is the model output or model prediction,

y is the real-data output or target output,

x_j are each of the j inputs of the neuron with their corresponding weights W_j,

b_i are the bias associated with the neuron i

$x_{physics\ p}$ are each of the p inputs corresponding to the physics information with their corresponding weights W_{pj},

b_j are the bias associated with the j physics-informed inputs,

$\sigma()$ is the activation function

Physics information through the outputs

Alternatively, the outcomes of the physics-based equations or models can be included in the output layer of the NN, just like another bias parameter. This modified architecture compels the NN's weights and biases to adjust in order to account for the physics-derived information. This adaptation enables the network to grasp complex patterns that may be absent in the physics-based model, such as environmental influences. Additionally, these complexities do not necessitate prior definition, as the NN presents significant adaptability to accommodate to different patterns and capture the inherent variability within the entirety of the observed data, regardless of their nature. Consequently, the *physics learnt through the output neurons* become an integral part of the network's forward pass as reflected in equation (5.14), enhancing the network's ability to extrapolate effectively.

$$\hat{y}_i = \sigma\left((\sum_j W_j x_j + b_i) + y_{physics\ p} + b_p\right) \tag{5.14}$$

where:

$\hat{y}$ is the model output or model prediction,

y is the real-data output or target output,

x_j are each of the j inputs of the neuron with their corresponding weights W_j,

b_i are the bias associated with the neuron i

$y_{physics\ p}$ are each of the p outputs corresponding to the physics information,

b_j are the bias associated with the p physics-informed outputs,

$\sigma()$ is the activation function

5.6.3 Convolutional neural networks

A convolutional neural network (CNN) is a multistage NN generally used for spatial pattern recognition [119], consisting of a filter phase followed by a classification or prediction phase. The architecture is framed on a number of CNN layers followed by dense (fully connected) layers. This structure enables hierarchical learning and information extraction in subsequent layers. The filter stage involves convolutional, batch normalisation, activation, and pooling layers. The classification/prediction phase uses fully connected dense layers to establish pattern relationships. Regarding architecture, a small multilayer CNN is enough to train a full model of regular size with a good representation of the input signals, improving the overall performance of the network [120].

The order of the CNN is set according to data dimensionality. For the case in which data primarily exhibit variations along a single dimension, such as time or a single spatial axis, it is a 1-D problem. For 1-D CNNs, the convolutional layer applies a sliding time window along the feature series axis to obtain subsequences (Figure 5.8). Each subsequence is element-wise multiplied with the kernel to obtain the convolution result [121]. The computation of each unit in the convolutional feature signal can be expressed as:

$$C_i = \sigma_i \left(W_i \otimes S_{i-1} + b_i \right) \tag{5.15}$$

Here, S_{i-1} represents the $(i-1)$-th input feature signal or subsequence, W_i is the weight matrix connecting the $(i-1)$-th input feature signal to the i-th output convolutional feature signal and b_i is the bias term for layer i. The sign $\otimes$ denotes the convolution operation, $\sigma_i()$ is i-th the activation function, and finally C_i represents the output, namely the featured map or convolutional featured signal [122].

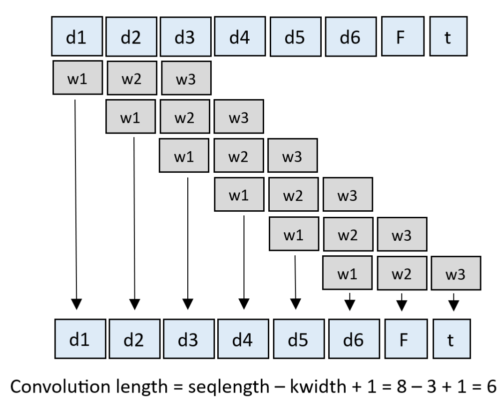


Figure 5.8: Example of a 1D CNN with kernel width = 3.

CNNs are typically trained in a supervised manner (i.e., using a labelled dataset where the input data is paired with corresponding output labels) using stochastic gradient descent and backpropagation algorithm [123]. During each iteration, the gradient magnitude of the network parameters, including the weights of convolutional and fully connected layers, is computed. These parameter sensitivities are then used to update the CNN parameters iteratively until a stopping criterion is met. The backpropagation method is a well-known procedure in the literature [122] and leads to a CNN architecture that efficiently captures spatial invariance, identifying relevant patterns in the input data while maintaining parameter efficiency.

5.6.4 Generative neural networks

Generative neural networks are designed to generate new data that is similar to the training data they were exposed to. Unlike discriminative models that focus on classifying input data into predefined categories, generative models learn the underlying patterns and distribution of the training data to generate new samples.

Within the umbrella of generative models, six types of algorithms are considered [124]: Autoregressive Models (AMs), Flow-based Models (FBMs), Energy-based Models (EBMs), Variational Autoencoders (VAEs), Diffusion Models (DMs), and lastly, Generative Adversarial Networks (GANs) (Figure 5.9).

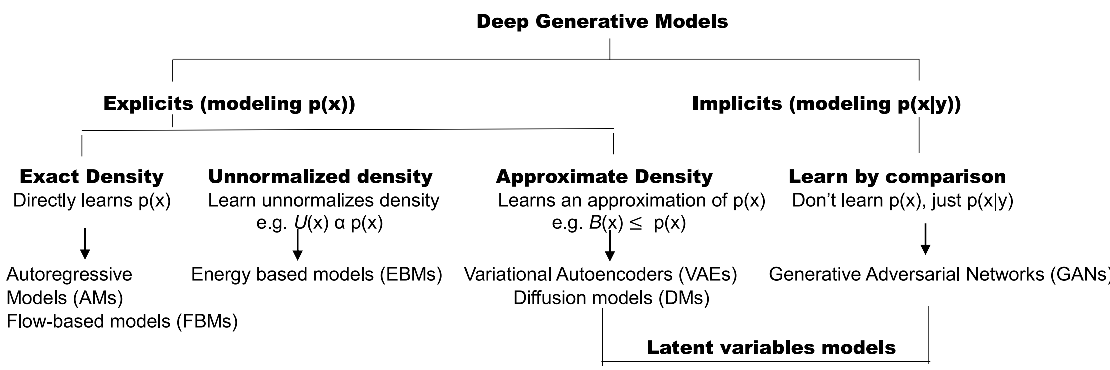


Figure 5.9: Types of generative models.

Explicit models specifically provide a detailed representation of the data likelihood based on parameters, leading to improved interpretability at the cost of reduced flexibility. Conversely, *implicit models* introduce hidden variables to capture complex data distributions, offering greater versatility but potentially reduced interpretability. In addition, while likelihood-based models such as AMs, FBMs, EBMs, VAEs and DMs can be trained stably, implicit models like GANs can be unstable to train. To cope with this difficulty, mathematical implementations such as the use of Wasserstein distance as a loss function [125] along with the incorporation of the gradient penalty technique [126], make the training process more stable and avoid other training issues in GANs, such as the model collapse and the vanishing gradients.

The use of implicit models is preferred in complex systems as it eliminates the need to deal with explicit likelihood expressions, simplifying the modelling process and making it more tractable. Besides, the use of latent variables exhibits robustness when dealing with noisy or incomplete data and demonstrates a remarkable ability to adapt to evolving data, excelling particularly in capturing intricate, non-Gaussian data distributions.

Among the latent variable models, GANs are considered more flexible and less constrained than VAEs and DMs in terms of modelling data distributions. This flexibility is a notable characteristic of GANs and is attributed to their implicit generative approach: GANs create data without explicitly modelling the data distribution, unlike VAEs and DMs. The focus of GANs is to capture data features and structures through a competitive learning process (named adversarial) between a discriminator and a generator, which continuously improves the quality of generated data without being bound to a predefined probability distribution. That fact, along with the introduction of latent variables which offer a high degree of freedom in data generation, enables GANs to produce diverse and realistic data instances, completely original.

In civil engineering, GANs are the preferred generative method due to their probed capability of encapsulating damage characteristics using a combination of categorical and continuous variables [127]. This means that the integration of GANs into the existing SHM processes such as damage assessment provides significant potential benefits [101].

Generative Adversarial Neural Networks (GAN)

A standard GAN consists of two NNs: 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 aligns with the distribution of the real data [128]. To illustrate the main structure of a GAN, Figure 5.10 presents a schematic representation.

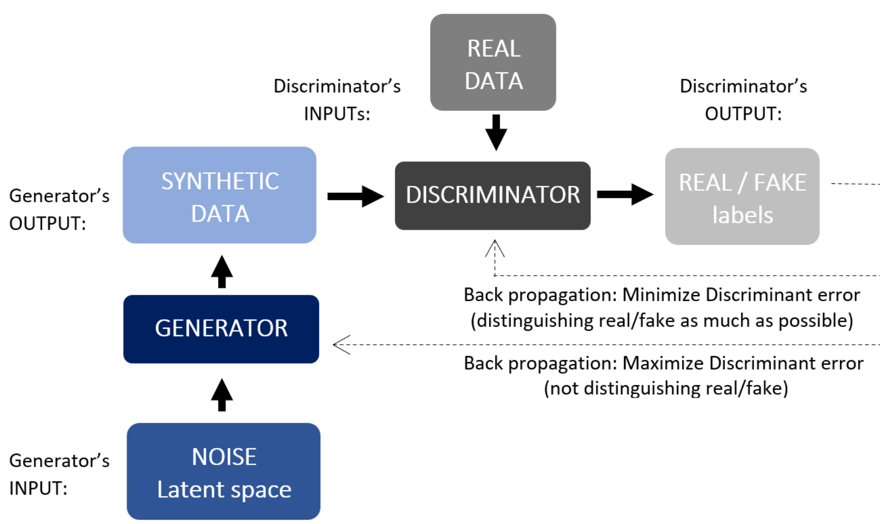


Figure 5.10: Structure of a GAN.

Let $x \in \mathcal{X} \subseteq \mathbb{R}^d$ be the vector representing the real data, and $\pi_r : \mathcal{X} \to \mathbb{R}^+$ a probability density function (PDF) associated with that data. Similarly, we denote $\tilde{x} \in \tilde{\mathcal{X}} \subseteq \mathbb{R}^d$ as the latent noise vector, which follows the PDF $\pi_g : \tilde{\mathcal{X}} \to \mathbb{R}^+$ (typically a multi-dimensional 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 real data for training x and the generated vector $G(\tilde{x})$ as inputs and results as output the probability of the generated sample $G(\tilde{x})$ belonging to π_r. The training of D involves maximising the probability of correctly assigning labels ('real' or 'fake') to the generated samples. Conversely, the training of the G aims to minimise the probability of the discriminator classifying the generated samples as fake. The training, based on a min-max approach, adheres to the following equation (5.16) for the objective function L:

$$\underset{G}{min}\, \underset{D}{max}\, L_{\text{GAN}} = \mathbb{E}_{\pi_r}\left[\log D(x)\right] + \mathbb{E}_{\pi_g}\left[\log\left(1 - D\left(G(\tilde{x})\right)\right)\right] \tag{5.16}$$

Wasserstein Generative Adversarial Neural Networks with Gradient Penalty (WGAN-GP)

To enhance the stability of GANs, various improvements have been suggested to optimise the objective function. One recent advancement in this domain is the Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) [126, 129]. 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 vanishing gradients (extremely small gradients) and facilitating more consistent training [130]. In WGAN-GP, the discriminator is often referred to as 'critic' since its focus is not on classifying real or fake samples, but on determining a degree of belief, providing confidence or reliability estimates for the generator's predictions [131]. By designing the loss function based on the Wasserstein-1 distance, the WGAN-GP improves training stability with minimal hyperparameter tuning, as compared to the original GAN. The Wasserstein-1 distance is defined as follows:

$$W(\pi_r, \pi_g) = \inf_{\prod(\pi_r,\pi_g)} \mathbb{E}_{\gamma\sim\prod(\pi_r,\pi_g)}\left[\|x - \tilde{x}\|\right] \tag{5.17}$$

where $\prod(\pi_r, \pi_g)$ denotes the set of all joint distributions over π_r and π_g, and the function $\prod(\cdot)$ can be interpreted as the measure of mass required to be transported from x to $\tilde{x}$ in order to transform π_r into π_g. $\mathbb{E}$ is the expectation and $\|\cdot\|$ is the norm operation. 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 achieves Lipschitz continuity constraint in the critic, making it suitable for computing the Wasserstein distance. Thus, the objective function of the WGAN-GP is defined as follows:

$$L_{\text{WGAN-GP}} = \mathbb{E}_{\pi_g}\left[D\left(\tilde{x}\right)\right] - \mathbb{E}_{\pi_r}\left[D\left(x\right)\right] + \lambda\mathbb{E}_{\pi_g}\left[\left(||\nabla_{\tilde{x}} D\left(\tilde{x}\right)||_2 - 1\right)^2\right] \tag{5.18}$$

where the term λ is the penalty coefficient used to weigh the gradient penalty term and ∇ represents the gradient operator.

Conditional Generative Adversarial Neural Networks (CWGAN-GP)

In a standard WGAN-GP, the generator produces data from randomness, and the critic's goal is to evaluate the quality of the generated samples, all completely controlled by random noise. However, in a conditional WGAN-GP, both the generator and the critic receive conditional information that guides the data generation process (Figure 5.11). This is a way to ensure class balance and context aware in the generated output [132].

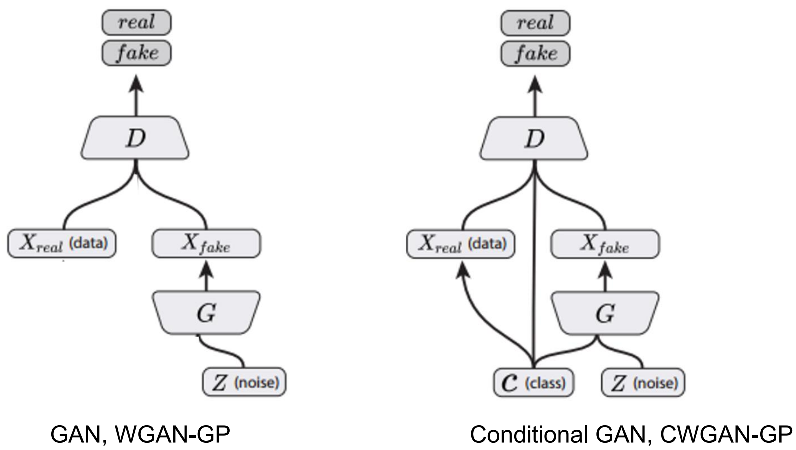


Figure 5.11: Differences between GANs and conditional GANs.

Let c the *code* vector containing the condition information to be given to the generator and the critic. This *code* may be a continuous variable or be discrete/categorical. According to this approach, the critic creates a different decision boundary depending on the condition c, resulting in a generator that learns to produce samples corresponding to specific variables or belonging to different categories [133]. This way, the equation governing the training of the CWGAN-GP follows the expression:

$$L_{\text{CWGAN-GP}} = \mathbb{E}_{\pi_g}\left[D\left(\tilde{x}|\, c\right)\right] - \mathbb{E}_{\pi_r}\left[D\left(x|\, c\right)\right] + \lambda \mathbb{E}_{\pi_g}\left[\left(||\nabla_{\tilde{x}} D\left(\tilde{x}|\, c\right)||_2 - 1\right)^2\right] \tag{5.19}$$

with c being the code vector containing the condition information, fed in both discriminator and generator as an additional input layer.

Part II

Contributions

Chapter 6

Digital Twin conceptualisation in civil engineering

Defining a concept like a DT can be challenging due to its interdisciplinary nature and the need to encompass multiple aspects. While it may not be described using a single mathematical equation, computational algorithm, chart, or graph, it can be precisely defined through a structured textual definition that covers its key components and characteristics. This comprehensive definition includes the following elements:

- Interdisciplinary Nature: Acknowledgement that the DT spans multiple disciplines and technologies, essential for capturing the complexity and dynamics of real-world systems.
- Description of the Concept: Considering that the concept of a DT is still constantly evolving and should be adaptable to various contexts, a comprehensive but flexible definition is necessary.
- Key Attributes or components: A list of the essential DT attributes or capabilities is essential to establish a common understanding, define its scope, guide development, and ensure alignment with specific requirements and contexts.
- Data Sources: Sources from which the DT derives its information, such as sensors, simulations, historical data, and external sources.
- Purpose and objectives: Due to its versatility, while there are general purposes and objectives for the DT, it can be adapted to meet the specific needs, goals, and challenges of different industries, applications, and contexts. The primary goals of a DT are to achieve enhanced understanding which leads to improved decision making, performance optimisation, predictive capabilities, and reduced risk and operational cost.

Further elaboration of these elements is provided in the following sections of the present chapter.

6.1 Interdisciplinary nature of the concept of Digital Twin

The interdisciplinary nature of the DT acknowledges that the concept transcends the boundaries of individual disciplines and seamlessly integrates various technologies and domains of expertise. Without this collaborative integration, the concept would not be feasible.

Addressing complex systems demands insights and cooperation from diverse disciplines such as engineering, computer science, and physics, to construct comprehensive representations of physical assets. Moreover, the DT's strength lies in its fusion of technologies like IoT, AI, edge/fog computing and simulation, all of which capture the dynamics of the real-world systems. It should be emphasised that when referring to *simulation* within the context of DT, specifically it is discussed the design of *models* to replicate the behaviour of physical systems or processes.

From this perspective, the DT promotes cross-functional understanding, enabling professionals from different backgrounds to comprehend how their decisions impact the entire system. The DT encourages problem-solving beyond disciplinary boundaries and catalyses innovation through diverse perspectives. It is adaptable to various industries, from manufacturing to civil engineering, addressing complex challenges in each domain. In essence, the DT embodies the synergy of interdisciplinary collaboration and technology integration, rendering it a powerful tool for understanding, optimising, and innovating across complex structures.

6.2 Definition of the concept of Digital Twin

Given the continuous evolution of the DT concept in synchrony with the technologies it encompasses, as well as its need for adaptability across various contexts, the definition should be both comprehensive and flexible, remaining open to changes and developments. It is widely accepted that a DT can be defined as a dynamic and evolving digital representation of a physical entity, system, or process. It leverages a combination of real-time and historical data, advanced modelling for simulation, AI, and multidisciplinary knowledge to provide a holistic and actionable understanding of the physical counterpart. DTs serve as powerful tools for monitoring, analysing, optimising, and even autonomously managing complex systems and assets across diverse domains. They adapt to specific use cases, technologies, and industries, making them versatile solutions for enhancing decision making, predictive capabilities, and operational efficiency. This definition acknowledges the flexibility and adaptability of the DT concept while highlighting its core elements, including data integration, modelling, AI, and its role in improving various aspects of physical systems. It leaves room for customization and specialisation depending on the specific application, technology integration and context in which DTs are employed.

For a concise description of the DT in the context of civil engineering, it is presented in the following sections both a mathematical formulation and a computing algorithm expression. The mathematical formulation serves as theoretical foundation, enabling analytical capabilities for comprehending and simulating the DT's behaviour. Conversely, the algorithmic expression is indispensable for practical implementation, real-time functionality, and seamless integration with existing systems. These dual expressions work in tandem, guaranteeing the DT's effectiveness in representing, analysing, and optimising physical systems or processes across diverse domains.

6.2.1 Mathematical formulation

Before delving into a detailed explanation of the mathematical formulation, it is essential to establish the time framework, defined by non-dimensional discrete time steps as $t \,\epsilon\, \{0, ...T\}$. In this way, $t = 0$ indicates the beginning of the time period considered, such as the instant in which the physical asset is built and starts its Useful Life (UL). Similarly, $t = T$ corresponds to the end of its UL. It has to be noticed that the time duration between time steps may vary and will correspond to the DT monitoring frequency, so the DT will be updated once per monitoring time step.

With the timeframe defined, now it is under consideration the structural performance of the asset, represented by a n-dimensional state vector. The *physical state* is a parameterisation of the properties of the physical asset at time t and denoted as $\mathrm{s}(t) \in \mathcal{D} \subset \mathbb{R}^{\mathcal{D}}$, with $\mathcal{D}$ being the *observation space*. It is also assumed that the states $\mathrm{s}(t)$ can be measured during operation and that, at a certain time t, these states can be manifested through sensors' measurements $\mathrm{s}(t) = \mathrm{q}(\mathrm{w}, \mathrm{e})$, where $\mathrm{q} : \mathbb{R}^{n_w} \times \mathbb{R}^{n_e} \to \mathcal{D}$ is a measurement equation, $\mathrm{w(t)}$ is a n_w-dimensional measurement input vector which accounts for the measurement error, and $\mathrm{e}(t) \in \mathbb{R}^{n_e}$ is a vector of environmental variables. Note that the physical asset cannot be directly observable, but only indirectly and partially, so that the error is present.

The *digital state* is described by a parametrised model or set of coupled models representing the asset in the digital world and referred to as $\hat{\mathrm{s}}(t) \in \mathcal{M} \subset \mathbb{R}^{\mathcal{M}}$, with $\mathcal{M}$ the space of the models class. The expression of the digital state is made by means of a model m so that $\hat{\mathrm{s}} = \mathrm{m}(u, \mathrm{e}, \theta)$ and $\mathrm{m} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_e} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{\mathcal{M}}$, which depends on a set of n_θ uncertain model parameters $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ along with a vector of model input variables $u(t) \in \mathbb{R}^{n_u}$ and the environmental vector $\mathrm{e(t)}$. The aforementioned quantities of interest have been summarised in Table 6.1 .

Table 6.1: Quantities of interest in the mathematical formulation of the DT.

Component	**Notation**	**Description**
s(t)	$s \in \mathbb{R}^{n_w} \times \mathbb{R}^{n_e} \to \mathcal{D}$	Physical state manifested through sensors' measurements
$\hat{s}(t)$	$\hat{\mathrm{s}} \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_e} \times \mathbb{R}^{n_\theta} \to \mathcal{M}$	Digital state configured as a set of coupled models
e(t)	$\mathrm{e} \in \mathbb{R}^{n_e}$	Vector of environmental variables
w(t)	$\mathrm{w} \in \mathbb{R}^{n_w}$	Measurements' input vector
$\theta(t)$	$\theta \in \mathbb{R}^{n_\theta}$	Models' parameters
u(t)	$\mathrm{u} \in \mathbb{R}^{n_u}$	Vector of models' input variables

Following this approach, the DT can be mathematically described in a specific context $\mathfrak{C}$ defined as $\mathfrak{C} \subset \mathcal{D} \times \mathcal{M}$, as follows [46]:

$$\underbrace{\mathrm{q}(\mathrm{w}, \mathrm{e})}_{\mathrm{s}} \xleftrightarrow[\mathfrak{C}]{\mathrm{CS}} \underbrace{\mathrm{m}(u, \mathrm{e}, \theta)}_{\hat{\mathrm{s}}} \tag{6.1}$$

where the double-arrow indicates that the correspondence between the physical and the digital representation of the structural states $\mathrm{s}, \hat{\mathrm{s}} \in \mathfrak{C}$ is materialised through the communication system (CS), which connects the sensors, the actuators, and the DT.

6.2.2 Computational expression

A computational expression of a DT aims to represent its processes, interactions, and functionalities in a structured and algorithmic manner that can be implemented using computational tools. While it may not be possible to provide a generic algorithm for a DT, as it highly depends on the specific application, it can be created a conceptual algorithm that outlines the key steps involved in the DT's functioning. This would include processes related to communication with the physical counterpart, data acquisition, data processing when needed, model integration, and decision making.

The Algorithm 1 shows the process followed by a generic DT of an asset. The process begins with the monitoring of data acquired from sensors, including environmental factors that may impact the asset. These data are then integrated into the DT models, with a focus on updating model parameters to enhance accuracy and applying Bayesian inference for uncertainty quantification. The algorithm proceeds to perform analytics, which involves evaluating the health of the system through SHM and predicting its remaining useful life using PHM techniques. Subsequently, it engages in decision making, assessing optimal maintenance strategies, including repair (corrective), maintenance based on condition (condition-based), or further monitoring (predictive).

Visualisation tools and dashboards are fed to offer relevant information to human operators, facilitating their decision-making processes. Additionally, the algorithm accommodates autonomous/automated decision making, employing either expert-knowledge predefined logic or AI-driven responses. It also activates actuators to execute decisions generated by the automated process.

Ensuring real-time communication through IoT is a crucial aspect, facilitating data transmission between sensors, actuators, controllers, operators, and stakeholders. Furthermore, the Algorithm 1 emphasises the synchronisation of processes between the physical and the digital components of the DT.

Scheduled operations are categorised into *almost-real-time* and *offline* operations. The former triggers specific analytics or actions based on predefined conditions or scheduled times, addressing tasks that require near-real-time attention. The latter involves tasks that must be performed offline, such as data curation, model updating, and deeper analysis during scheduled downtimes.

Algorithm 1 DT algorithm

Initialise Digital Twin components
While system is operational:

1. **Monitor data:**
   Acquire real world data from sensors
   Obtain environmental data (temperature, humidity...) impacting the asset
   Update the DT models with thedata
2. **Perform fusion of data and models:**
   Integrate data and models
   Update model parameters to enhance accuracy
   Apply Bayesian inference for uncertainty quantification
3. **Execute analytics:**
   Evaluate system health (Structural Health Monitoring - SHM)
   Predict remaining useful life (Prognosis and Health Management - PHM)
4. **Perform Decision Making:**
   Assess optimal maintenance strategies: Determine if repair (corrective), maintenance (condition-based), or further monitoring (predictive) is required
5. **Provide visualisation and human decision support:**
   Provide relevant information for human operators in the loop through visualisation tools and dashboards
6. **Accomplish autonomous/automated decision:**
   Employ expert-knowledge predefined logic or AI-driven response for automated decision-making
   Activate actuators to execute the orders generated by the decision-making process
7. **Ensure real time communication (IoT):**
   Transmit data to/from sensors, actuators, controller, operators and Stakeholders in real time
8. **Synchronize processes:**
   Ensure coordination among twin components: the physical and the digital
9. **Scheduled Almost-Real-Time operations (at Time 1 or Condition 1):**
   Trigger specific analytics or actions based on predefined conditions or scheduled times
   Execute tasks that require near-real-time attention
10. **Scheduled Offline operations (at Time 2 or Condition 2):**
   Trigger specific analytics or actions that can be performed offline and do not require immediate attention
   Conduct data curation processes, data generation, model updating, data & knowledge base storage and maintenance, or deeper analysis during scheduled downtimes

End While

6.3 Purpose and objectives

The concept of a DT exhibits remarkable versatility, allowing for its adaptation to cater to the particular requirements, objectives, and intricacies of diverse industries, applications, and contextual settings. This adaptability stems from the inherent flexibility of the DT framework, making it a valuable tool for addressing industry-specific challenges and realising targeted goals.

At its core, the primary objectives of a DT remain consistent, irrespective of the industry or application. These overarching goals serve as guiding principles for the deployment of DTs across various domains. The fundamental aims of a DT encompass:

- Enhanced Understanding: One of the main purposes of a DT is to provide a comprehensive and dynamic digital representation of a physical system or asset. This representation enables stakeholders to gain a deeper understanding of the asset's behaviour, condition, and performance. This enhanced understanding extends to various aspects, including structural integrity, operational efficiency, and environmental interactions.

- Predictive Capabilities: DTs achieve predictive capabilities by combining data analysis, modelling, AI techniques, and continuous updating. These capabilities empower organisations to anticipate issues, failures, or performance deviations before they occur. Predictive maintenance, for instance, allows for proactive interventions, reducing downtime, and preventing costly breakdowns. DTs also employ probabilistic methods to account for uncertainties and variability in the data. This provides decision makers with a range of possible outcomes and their associated probabilities.

- Improved Decision Making: With a richer understanding of the physical system, DTs empower decision makers to make more informed choices, or get better data-driven solutions. The real-time insights, predictive analytics, and data derived from simulations equip decision makers with precise information to assess the potential outcomes of different courses of action, resulting in well-informed and highly effective decisions.

- Performance Optimisation: DTs can significantly help in optimising the performance of physical assets and systems. Through continuous monitoring, analysis, and feedback loops, DTs enable stakeholders to identify opportunities for efficiency improvements, cost reductions, and enhanced operational performance. This optimisation extends to areas such as energy efficiency, maintenance scheduling, and resource allocation.

- Risk Mitigation: By continuously assessing the condition and behaviour of physical assets, DTs proactively contribute to risk reduction. Early detection of anomalies or deterioration enables timely risk mitigation strategies, safeguarding both the asset's integrity and the safety of stakeholders.

- Operational Cost Reduction: Through a combination of improved decision making, performance optimisation, and risk mitigation, DTs contribute to the reduction of operational costs. By minimising unexpected expenses associated with downtime, emergency repairs, or inefficient resource allocation, organisations can achieve significant cost savings.

In summary, while the general purposes and objectives of a DT remain consistent, its adaptability allows it to be tailored to specific requirements and contextual nuances. Whether applied in manufacturing, infrastructure, or any other domain, the overarching objectives of a DT revolve around enhancing understanding, facilitating informed decision making, optimising performance, predicting future outcomes, mitigating risks, and ultimately reducing operational costs. The versatility of the DT framework ensures its relevance and applicability across diverse implementations, making it a valuable asset in the era of digital transformation.

6.4 Components of the Digital Twin

Certain components of the DT concept are consensually regarded as essential, requiring their presence irrespective of the specific domain or industry in which they are employed. These foundational constituents provide the core functionality of a DT, endowing it with fundamental capabilities that remain indispensable across diverse application contexts. These general elements include real-time data collection, integration with behaviour models, analytics for decision making, visualisation dashboards for decision support, and bidirectional IoT communication. Their widespread acceptance in the scientific community underscores their significance in enabling the DT to fulfil its objectives in various domains and industries.

A literature review was conducted in Chapter 2, focused on the specific components required for the application of the DT concept in the field of civil engineering, particularly in the context of structures. This review, while not exhaustive, was concentrated on the examination of pertinent publications that primarily featured practical applications. Literature reviews themselves were omitted from this investigation. In this context, a comprehensive analysis was performed on existing publications employing terms such as *Digital Twin*, *Structures/Structural (...)*, and/or *Civil Engineering* within practical contexts.

The noteworthy outcomes reveal the existence of various significant components scattered across these publications, without a unified integration into a cohesive DT application. This suggests a lack of a comprehensive conceptualisation of the DT in civil engineering that unites all the crucial elements identified separately in these publications. Among these identified elements, certain components align with the overarching elements of a universal DT concept, including real-time data, models, bidirectional IoT communication, and more. However, there are also domain-specific constituents, such as the diagnostics and prognostics modules, along with uncertainty quantification. Additionally, some components represent state-of-the-art advances in DT development, exemplified by the incorporation of AI and autonomous decision-making capabilities.

The summary of the aforementioned components, depicted in Figures 6.1 and 6.2, is developed in the following sections.

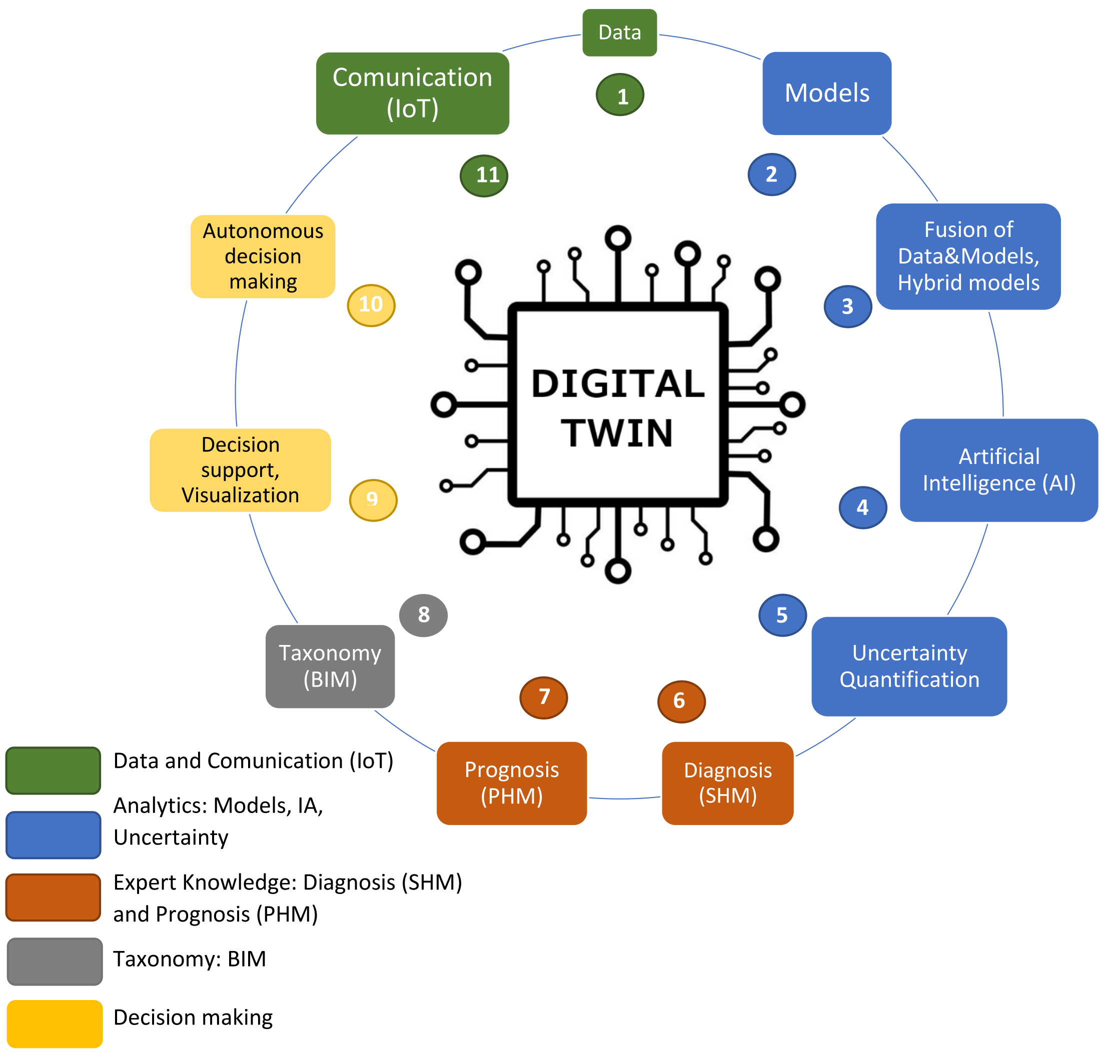


Figure 6.1: DT components

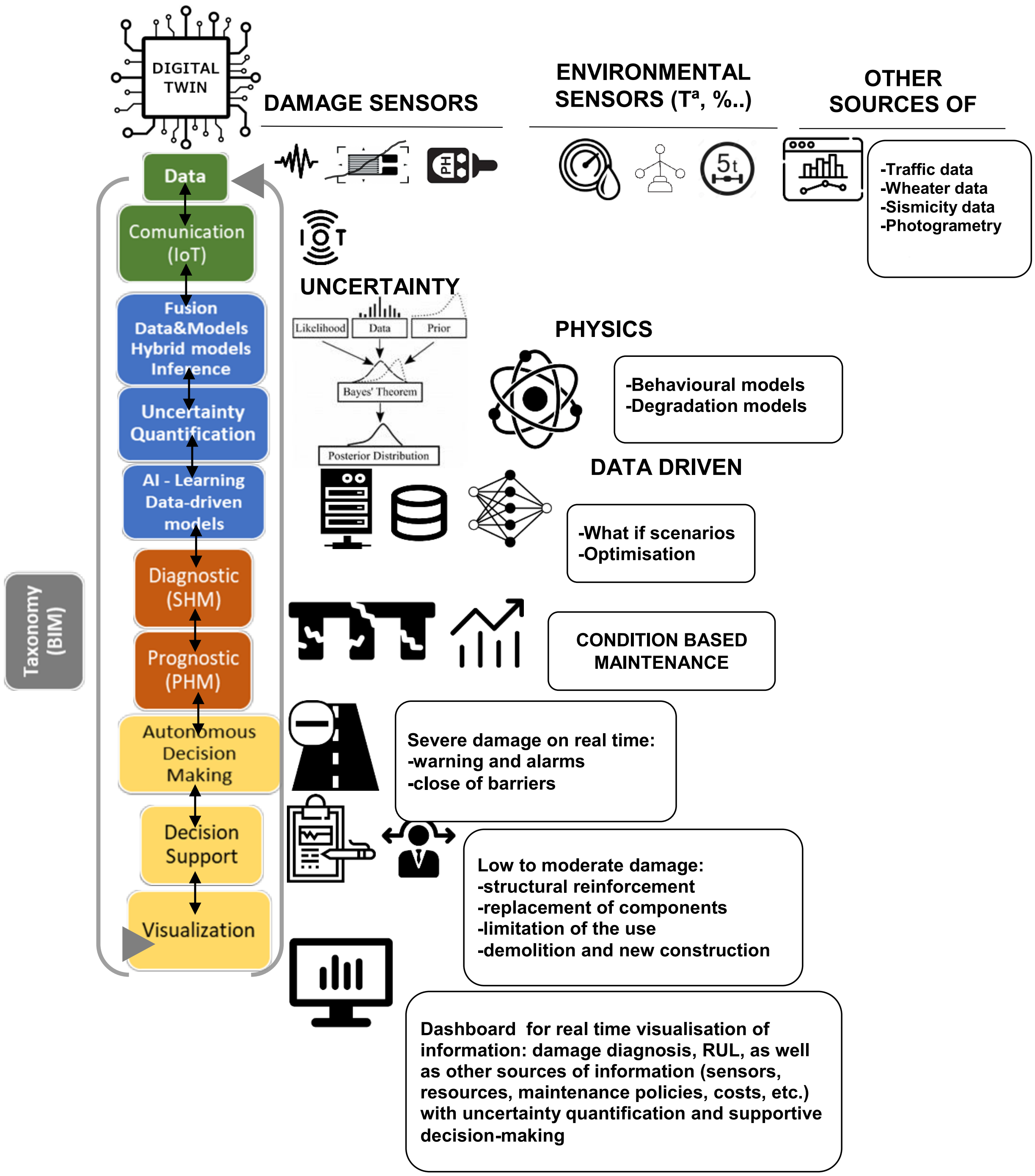


Figure 6.2: Workflow between the DT components

6.4.1 Data

The data that feed the DT comes from diverse sources, depending on the specific application and domain. What is essential, however, is the real-time integration of monitoring data from the system. These data pertain to the system's main physical parameters such as displacements, accelerations, external forces, and more, as well as environmental factors critical for system decision making, including loads, temperature, humidity, wind forces, and others. Furthermore, historical system data can be included, along with other informative sources.

Whenever possible, working with discrete measurements retaining meaningful physical significance is a preferred choice over continuous signals for several compelling reasons. Firstly, it leads to a reduction in data storage requirements and computational load, making it particularly well-suited for edge and fog computing environments characterised by limited processing capabilities and the need for low-latency processing. Moreover, discrete data is readily interpretable without extensive preprocessing. This enables quick real-time analysis, facilitating timely decision making. At the same time, discrete data presents enhanced comprehensibility, being easily understandable and analysable for both human operators and ML algorithms. Lastly, discrete data tends to exhibit greater stability and is less susceptible to noise compared to continuous data, resulting in more robust models and decision-making procedures.

In addition to these considerations, it is recommended to choose directly measurable input and output magnitudes for feeding the DT rather than relying on indirect estimation or inference. The direct measurement of these magnitudes simplifies the DT's architecture by eliminating the need for complex observer models or estimation methods, which can be computationally intensive and introduce additional uncertainty. While there may be situations where direct measurements are not feasible due to technical constraints or budget limitations, in such cases, the use of observer models or filtering techniques such as the Kalman filter [134] becomes necessary to estimate unmeasured quantities.

The expression of discrete data should incorporate a timestamp, a crucial element that enables synchronisation across measurements, decisions, and actions within the DT. Synchronisation is crucial to provide a coherent view of the system's behaviour without data discrepancies and make informed decisions based on the most up-to-date information. It also helps establish a chronological sequence ensuring proper alignment in time, making the DT able to exchange information with external systems seamlessly and consistently.

It is worth noting that apart from the inclusion of those timestamps, in discrete data representations there is no need for time dependencies or constraints, in contrast to continuous signals. Discrete data represents specific, isolated measurements or observations at distinct points in time and are not inherently tied to a continuous time domain, simplifying data processing and analysis without the need to consider continuous time dependencies.

6.4.2 Models

Models play a central role in a DT as they serve as the digital representation and simulation of the physical counterpart. Models enable real-time analysis, prediction, and optimisation of the physical system or asset. They provide the foundation for decision making, diagnostics, and prognostics within the DT, contributing to improved performance and maintenance strategies. Essentially, models are the core intelligence that empowers the DT to achieve its goals across diverse domains and industries.

Reality is captured by the models in a simplified yet accurate manner, focusing on critical aspects of infrastructure security while considering any limitations or constraints. Models serve as versatile representations of real-world systems, enabling analysis, prediction, and experimentation in a virtual environment. Simulation is a crucial component of DTs, allowing for testing and optimisation without direct modifications to the physical asset, which can be expensive or impractical in real time. Furthermore, simulation facilitates *what-if analysis*, preparing the DT to handle unforeseen scenarios, including rare or unexpected events (*black swan* scenarios). Simulation also supports optimisations in the DT by considering the model's response to different input conditions and conducting sensitivity analyses to identify the most influential parameters affecting the DT's performance.

Models in a DT need to be dynamic and adaptable to accurately mirror the behaviour of the physical asset or system. Different types of models can be implemented in the DT, including physics-based and data-driven models. Physics-based models are derived from the fundamental physical principles governing a system's behaviour. They are based on a deep understanding of the system's dynamics, using mathematical equations to describe how inputs affect outputs. These models are employed in well-understood and highly predictable systems where the underlying physics is clear and well-established. However, when a system's physics is complex or not well-understood, data-driven models come into play. These models rely on data collected from the system to learn its behaviour, without explicit knowledge of the underlying physical equations. ML techniques, such as NNs, are often used to develop data-driven models. They excel in handling complex and non-linear systems by learning patterns and relationships directly from the data. Together with ML, statistical and probabilistic methods fall under the category of data-driven models.

In practice, a hybrid approach that combines both physics-based and data-driven methods is often employed. For instance, physics-based models can be improved or fine-tuned using data-driven techniques, or data-driven models can incorporate physics principles through various strategies to adapt to real-world variations and uncertainties.

Surrogate models serve as simplified and computationally efficient representations of more complex and resource-intensive models. Creating surrogate models typically involves techniques such as regression analysis and DL algorithms, particularly NNs. The primary goal is to capture the essential features of the underlying system while minimising computational demands, achieving a balance between accuracy and efficiency. In essence, surrogate models play a pivotal role in ensuring the effective operation of DTs, acting as intermediaries that bridge the gap between intricate physical systems and the imperative need for real-time analysis and decision support.

6.4.3 Fusion of data and models

The fusion of information from data and models within the context of a DT involves combining and integrating these two sources of knowledge to create a comprehensive and accurate representation of the physical asset or system.

Continuously, real-time data and models are integrated within the DT environment. This process involves feeding the data into the models, aligning the timestamps, and ensuring that they are fully synchronised. The models may be continuously compared against the real-time data to validate their accuracy and any discrepancies can trigger updates to the models. The integrated information is used to assess the health of the physical asset, detect anomalies, predictive analytics and, in the context of decision making, give recommendations for maintenance, operational adjustments, or other actions to optimise performance and prevent issues.

The DT operates in a continuous feedback loop, where real-time data continuously update the models, and the models, in turn, provide insights and recommendations that guide actions in the physical world. In summary, the fusion of information from data and models is a dynamic and iterative process within a DT, where the virtual and physical worlds are closely intertwined to provide real-time insights, predictive capabilities, and support decision making for optimising the performance of the physical asset or system.

Various techniques, including Bayesian inference and fuzzy logic, can be employed to seamlessly blend data and models within a DT. Fuzzy logic incorporates linguistic inputs alongside predefined rules, combining them to generate actionable outputs. This method ensures that qualitative information is integrated with quantitative data, aiding in more comprehensive decision making. Other methods adopt similar strategies to fuse data and models. These techniques facilitate a comprehensive representation of the system's behaviour by combining real-world observations with model-based insights, enabling more informed decisions within the DT.

In Figure 6.3, different methods of combining data and models are depicted using an example of a metal tower system.

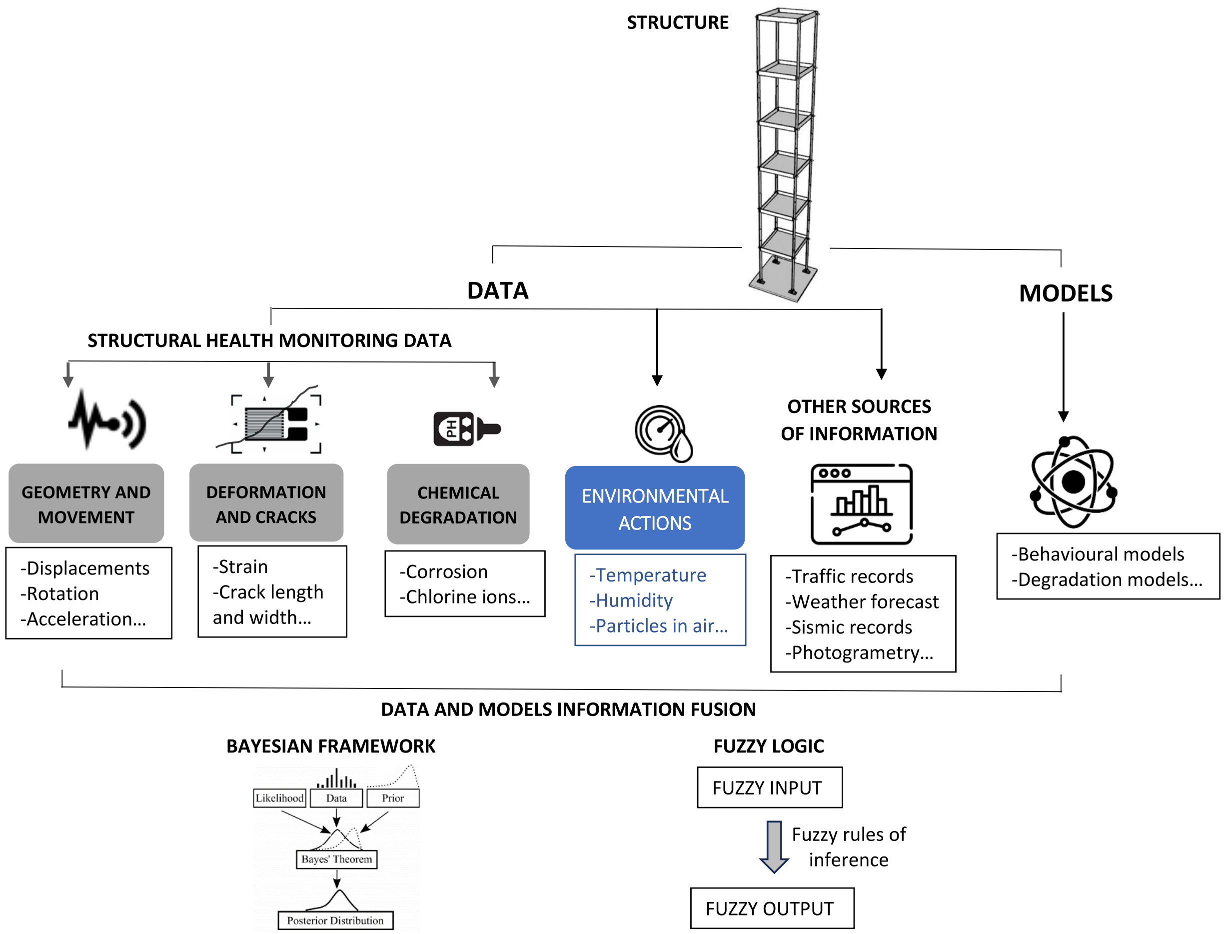


Figure 6.3: Illustrative example of data and models fusion in a structure.

6.4.4 Artificial intelligence

Artificial Intelligence (AI) plays a crucial role in DT by enhancing its capabilities in multiple key areas. AI facilitates data analysis and interpretation, enabling the identification of patterns and anomalies in real-time sensor data. It also supports predictive maintenance by forecasting system failures through predictive models. ML models, including DL NNs, are employed to create surrogate data-driven models within the DT, improving efficiency and adaptability.

AI assists in decision making by providing recommendations derived from reinforced learning and can even enable autonomous control of systems. Additionally, AI enhances human-machine interaction through Natural Language Processing (NLP) and cognitive computing techniques, simplifying user engagement. DTs are helped by AI to adapt to changing conditions as ML models can retrain with new data to improve accuracy and relevance, respond to unexpected events through generative methods, and optimise system performance using genetic algorithms, ultimately making DT more intelligent, responsive, and versatile across diverse domains and industries.

6.4.5 Uncertainty quantification

The data collected from sensors plays a crucial role in quantifying uncertainty within the DT. This uncertainty encompasses aspects related to the model, including simplifying assumptions and modelling errors, as well as uncertainties in the data due to measurement errors. Understanding and quantifying this uncertainty forms the foundation for informed decision making, with a focus on managing risks effectively.

Risk management in the context of the DT involves assessing and mitigating potential risks associated with the physical system or asset represented. It is common that in decision-making processes, first uncertainty needs to be understood and quantified before calculating the risk. The sequence typically begins with clarifying uncertainty. This involves gathering data and analysing information to better understand the sources and factors contributing to uncertainty. Once the sources of uncertainty are clarified, the next step is to quantitatively assess the associated risks. This includes assigning probabilities to different outcomes and evaluating the potential consequences of those outcomes. The quantification of risks enables decision makers to estimate the likelihood of various scenarios and their potential impacts. Subsequently, risk management strategies can be developed, incorporating risk mitigation and contingency plans, and risk transfer mechanisms like insurance. Decision makers can then make informed choices, considering the trade-offs between potential gains and losses, taking into account their risk tolerance.

It is important to note that not all uncertainty can be completely eliminated, especially in complex systems. However, the objective is to reduce uncertainty to a tolerance level where risks can be quantified, analysed, and effectively managed to support decision-making processes.

6.4.6 Structural health monitoring and diagnostics

SHM is a key capability of a DT within the civil engineering and infrastructure domains. It involves the continuous and real-time monitoring of a structure's condition and performance using sensors and non-destructive testing (NDT) techniques. These sensors can include accelerometers, strain gauges, displacement sensors, and more, and are intended to capture any anomaly that may indicate potential issues in the structure.

The focus in SHM lies in the continuous monitoring of critical components of a structure. These critical elements have a significant impact on the overall safety and functionality of the entire system. Techniques like FTA and FMECA aid in identifying these pivotal components, assisting in the strategic placement of monitoring sensors on these specific elements.

SHM is inherently multi-physics and multi-scale. This means that it considers various physical phenomena and operates at different scales. For example, in a large building, the SHM system may need to monitor both the macro-scale behaviour of the entire structure (e.g., overall vibrations) and micro-scale behaviour at specific critical points (e.g., stress concentrations in a particular beam). It also considers multiple physical factors that can affect the structure's health, such as temperature, humidity, and external forces like wind or seismic activity.

The data collected through SHM is not just used to monitor the current state of the structure but also to extrapolate and predict its future behaviour. By analyzing trends and deviations from expected behaviour, the overall safety of the system can be assessed.

In the context of a damage detection strategy, SHM contributes to detect, localise and quantify damage within the structure, as it will be further elaborated in this thesis, specifically in Section 12.4.

6.4.7 Prognostics and health management

The PHM process goes beyond merely monitoring the current health of the structure; it delves into predicting how much longer the structure can effectively serve its intended purpose. Within the DT framework, prognostics is focused on estimating the RUL of a given infrastructure or asset. To achieve this, it employs a comprehensive approach that includes simulating various load scenarios and environmental conditions. These scenarios may deviate from the original design specifications, allowing for a more holistic understanding of the structure's behaviour.

Unlike traditional monitoring, which often relies on sensors placed at specific locations, prognosis in the DT considers the entire structure. It takes into account the potential impacts of changes in loads, environmental factors, and other variables on the structure's overall health and longevity. One key feature of DT-based prognostics is its ability to continuously update predictions. This is made possible by integrating real-time measurements with deterioration models. As new data is collected and analysed, the prognosis model can adapt and provide ongoing estimates of the structure's RUL. In summary, prognostics in the DT context is a dynamic process that predicts how long it will remain functional under various conditions. This approach enhances maintenance and decision making by providing valuable insights into when and how to take action to ensure the continued reliability and safety of the asset.

6.4.8 Taxonomy

In the context of BIM, a taxonomy serves as a hierarchical classification system used to organise and categorise information related to elements, materials, and other relevant entities within the system or asset being represented. BIM collaborative modeling relies on open standards such as Industry Foundation Classes (IFC) to ensure interoperability across different phases of planning, design, construction, and asset management (O&M). This standardised approach provides essential information for decision making throughout the asset's lifecycle. Within the BIM framework, the IFC standard functions as a taxonomy.

Even in a DT perspective primarily centred on the asset's O&M stage, BIM and IFC play a significant role. BIM data seamlessly integrates with other O&M systems, including sensors and IoT devices. Moreover, BIM models incorporate historical data like construction and maintenance records. This historical context can be very valuable for gaining insights into the asset's performance and refining models through training processes. Furthermore, IFC ensures interoperability by providing a standardised data format, allowing different software tools to exchange BIM data seamlessly.

6.4.9 Support for human decision and visualisation

In the realm of a DT, there is also a need for human intervention in decision making within the O&M phase, and here is when Decision Support Systems (DSS) play a pivotal role. These systems are instrumental in assisting human-in-the-loop by identifying the most appropriate course of action. They take into account a multitude of factors, including the cause of a failure, its immediate impact, available logistics, downtime costs, maintenance expenses, and various others. Operators and stakeholders benefit greatly from DSS by gaining access to intelligence, quantifying risks, and ultimately being equipped to make well-informed decisions. To present this wealth of information in an easily comprehensible manner, graphical visualisation is implemented within a user-friendly environment in the form of a dashboard, facilitating efficient decision making.

The transition from data to action invariably entails an analytical process that may encompass human intervention or operate in complete automation. The evolution of analytics progresses from descriptive and diagnostic to predictive and prescriptive stages, ultimately culminating in autonomous decision making and action execution (Figure 6.4, reproduced from [135]).

Within the structural civil engineering domain, the extent of human involvement versus full automation in decision making is influenced by multiple factors. These include the complexity of the system, the nature of the decision, considerations of risk and safety, compliance with regulatory requirements, organisational culture, and the level of integration of the DT technology itself. Striking the right balance between human intervention and automation depends on a combination of these elements, with each scenario potentially requiring a unique approach.

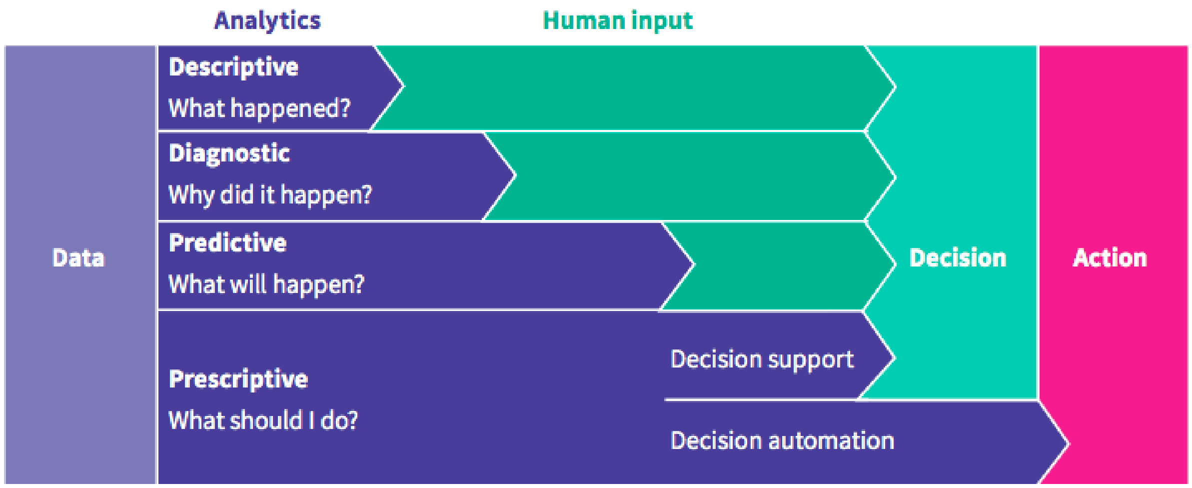


Figure 6.4: Analytics from Data to Action.

6.4.10 Autonomous decision

The level of autonomy and decision-making authority of a DT can vary widely depending on the specific application, industry standards, and safety regulations. Human oversight and intervention are typically integrated to ensure the reliability and safety of autonomous decisions made by the DT.

The autonomous decision-making process can be approached through various methodologies which encompass classic techniques such as fuzzy logic, graphical methods founded on expert knowledge including decision trees, finite state machines, Markov decision processes and Bayesian networks, or Petri nets, and AI-based analytics.

AI plays a substantial role in enhancing the DT's autonomous decision-making capabilities. It can be harnessed to create offline improvements in maintenance policies through techniques like reinforced learning or optimisation algorithms. These AI-driven approaches enable the DT to continuously refine and optimise its decision-making processes, ultimately contributing to more efficient and effective operations within a wide range of domains and industries.

In structural civil engineering, a DT can make autonomous decisions in various situations regarding monitoring, maintenance, emergency response, optimisation of performance, energy efficiency, risk mitigation, and adaptive structures. These decisions are driven by real-time data, models, and predefined rules. For instance, the DT can autonomously adapt to changing conditions, trigger maintenance, optimise energy use and respond to emergencies.

6.4.11 Communication

The heart of a DT is the ability to receive, process, and act upon real-time data so efficient communication is paramount for the seamless transfer of information, and this is facilitated through IoT technology.

IoT encompasses a network of communication components involving sensors, actuators, controllers, microprocessors, routers, gateways, and servers which can be edge, fog, and cloud located. Edge servers handle data locally, providing real-time responses and reducing latency and safety issues. Cloud servers offer scalability and centralised data storage, enabling advanced analytics and long-term data retention. In between, there is edge computing, introducing a distributed network of fog nodes (intermediary servers) placed at different points in the network between edge devices and the centralised cloud. Additionally, the choice of communication protocols is essential, as they need to adapt to various factors, including the volume of data to transmit, transmission distance, energy requirements, and available bandwidth. Examples of communication protocols commonly used in DTs include MQTT (Message Queuing Telemetry Transport) and HTTPS (Hypertext Transfer Protocol Secure) for web-based communication, among others.

IoT devices are instrumental in the collection of real-time data from different components within a system, being strategically placed throughout the physical infrastructure to monitor critical parameters. They continuously gather data and transmit it to the DT, where it can be analysed and used to create the virtual representation of the physical asset. However, it is important to distinguish between sensors and IoT devices. A sensor is a specific component within an IoT device. Sensors are responsible for gathering data from the physical environment, measuring parameters such as temperature, pressure, or humidity. They serve as the initial point of data acquisition. On the other hand, an IoT device represents a more comprehensive concept. It is a physical device that connects to the internet or another network and possesses the capability to both receive and transmit data. While IoT devices incorporate sensors to collect data, they go beyond this role. IoT devices are equipped with processing power, memory, and communication interfaces that allow them not only to collect data but also to transmit it to other devices or systems over a network. They often serve as data hubs, aggregating information, and facilitating its exchange with other servers or cloud platforms.

Communication in a DT is not limited to machine-to-machine interactions, as human-in-the-loop also plays a crucial role. DSS can present data and insights in a comprehensible manner to human operators, helping them make informed decisions.

6.5 Data sources

In the conceptualisation of a DT, it is important to have a clear understanding of the data needed to accurately represent the physical system and the sources from which this data will be obtained.

The data ingested by the DT is diverse and comes from various sources. The primary source of data is the real-time sensor data which provides essential insights into the current state of the physical system or asset. Other data is provided in almost real-time or offline modes to the DT. This data encompasses historical data, simulations, and complementary data from external sources.

Historical data includes records of maintenance logs, inspections, repairs, and any structural changes or modifications. This information is valuable for understanding the asset's maintenance history and tracking changes over time with a historical perspective on the asset's performance. External sources of data can include external databases, APIs, or web services that provide relevant information, helping assess how external factors may impact the structure. Examples of such data sources include weather forecasts, seismic activity reports, traffic reports, and more.

Additionally, simulations play a critical role in DTs by creating virtual models of the physical structure and enabling 'what if' analysis. These simulations generate data that allows for predicting how the structure will respond to different loads, environmental conditions, or potential damage scenarios. Several simulation methods are available, each tailored to specific application domains and objectives. These methods range from numerical techniques like FE analysis for structural simulations to statistical methods for probabilistic analysis and AI-based methods, including generative AI, capable of creating synthetic data and mimicking complex scenarios.

It is important that the data follows standards such as the IFC for interoperability and recognition by the BIM taxonomies. Furthermore, data format standardisation is vital to ensure compatibility and exchangeability. Common data formats include CSV (Comma-Separated Values), XML (Extensible Markup Language), and JSON (JavaScript Object Notation). Maintaining data quality standards is equally important throughout the data lifecycle, which includes data collection, validation, cleansing, and storage. This diligence ensures the accuracy and reliability of the information utilised by the DT.

Chapter 7

Incorporating uncertainty quantification into the Digital Twin workflow

Within the context of DTs, uncertainty quantification (UQ) assumes paramount importance, as it entails the rigorous characterisation and assessment of uncertainties permeating the various facets of the digital replica. Uncertainties can emanate from diverse sources, including measurement errors, model approximations, and environmental variability. Bayesian inference serves as a potent tool for the management of these uncertainties within the DT framework.

By applying Bayesian principles, the DT can systematically incorporate new observational data into existing models, facilitating the continuous adaptation and refinement of these models. This application extends to various crucial aspects, including uncertainty propagation and quantification, dynamic model updating and calibration, sensitivity analysis, and risk-based decision making (Figure 7.1). In essence, Bayesian inference plays a key role in the probabilistic treatment of information integration within the DT, substantially contributing to its reliability in predictive and decision-making processes.

Bayesian inference comprises two fundamental categories that capture its core aspects: the *forward probrem*, which involves leveraging existing knowledge to make predictions or infer outcomes, and the *inverse probrem*, which focuses on updating beliefs about model parameters based on observed data. By incorporating both problems into its operational framework, a DT facilitates informed decision making with risk management by enabling the consideration of two fundamental aspects: the prediction of anticipated system behaviour with quantified uncertainty (forward problem) and the continuous updating of the system's characteristics through the assimilation of real-time data (inverse problem).

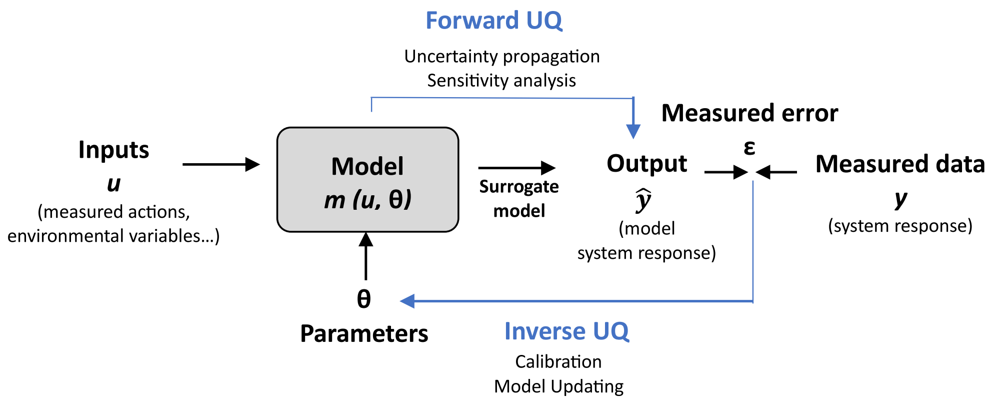


Figure 7.1: Uncertainty quantification, forward and inverse problems

The following table highlights the distinct objectives of the forward and inverse problems in the Bayesian framework, along with their respective inputs and outputs:

Table 7.1: Comparison between forward and inverse problems

	Forward Problem	**Inverse Problem**
Objective	Predict system output $\hat{\mathrm{y}}$ quantifying the uncertainty by $\mathrm{p}(\hat{\mathrm{y}}\vert\mathrm{u},\theta)$	Estimate parameters $\hat{\theta}$ quantifying the uncertainty by $\mathrm{p}(\hat{\theta}\vert\mathrm{y},\mathrm{u})$
Inputs	System inputs u, model parameters θ	System inputs u, Observed data y, model class $\mathcal{M}$
Outputs	System response predictions $\hat{y} = \mathrm{m}(u, \theta)$ and likelihood distribution $\mathrm{p}(\hat{\mathrm{y}}\vert\mathrm{u},\theta)$	Inferred model parameters $\hat{\theta}$ and posterior distribution $\mathrm{p}(\hat{\theta}\vert\mathrm{y},\mathrm{u})$

It has to be noted that Bayesian inference often involves iterative processes and sampling techniques, such as Markov Chain Monte Carlo (MCMC) or Variational Inference (VI), which aim to approximate posterior probability distributions. The challenge in applying Bayesian inference within a DT context lies in the computational demands of these methods, which may necessitate time-consuming computations. These operations are typically not conducive to real-time execution within the DT but can be performed in almost real-time, as the inherent computational demands of Bayesian methods may introduce latency in obtaining updated posterior distributions and refined model parameters.

7.1 The forward problem: uncertainty propagation

The forward problem encompasses the integration of measurement data with model-derived information to yield an output while providing the quantification of the inherent uncertainty. The inputs and model parameters could be imprecisely known, and this imprecision could be represented using probability distributions, making the forward problem stochastic in nature [136]. Various methods can be employed to address forward problems, among which *stochastic embedding* [137] emerges as one of the simplest yet rigorously structured approaches.

In accordance with the mathematical notation introduced in Section 5.1 and by incorporating the vector of environmental variables e into the input vector u, the discrepancy between the observed system output (previously referred to as the *physical state* and denoted as s) and the model output (previously referred to as the *digital state*, $\hat{s}$) can be expressed as an uncertain error term ε as follows:

$$\underbrace{\mathrm{y}}_{\text{system output}} = \underbrace{\mathrm{m}(u,\theta)}_{\text{model output}} + \underbrace{\varepsilon}_{\text{error}} \tag{7.1}$$

The error represented as ε can be described using a probability distribution, such as a zero-mean Gaussian distribution. This choice of distribution establishes the probability model for the system output y, which takes the form of a Gaussian distribution $\mathcal{N}$ characterised by a mean μ_ε equal to $\mathrm{m}(u,\theta)$ and a covariance Σ_ε, as presented below:

$$\varepsilon = (y - m(u,\theta)) \sim \mathcal{N}(0, \sigma_\varepsilon) \to y \sim \mathcal{N}(m(u,\theta), \Sigma_\varepsilon) \tag{7.2}$$

According to the Principle of Maximum Information Entropy (PMIE) [138] and the specifics detailed in [136], the expression for the output of the probabilistic forward model can then be derived from the deterministic model as follows:

$$p(y|u,\theta) = ((2\pi)^{N/2}|\Sigma_\varepsilon|^{1/2})^{-1} \cdot exp\left(-\frac{1}{2} \cdot (y - m(u,\theta))^T \cdot \Sigma_\varepsilon^{-1} \cdot (y - m(u,\theta)\right) \tag{7.3}$$

with N denoting the size of the observed system output y, and $m(u,\theta)$ corresponding to the output of the deterministic forward model. In the context of the problem, the variable y corresponds to the observed data, which is equivalent to the model output, and u represents the input vector.

An algorithmic description of the forward problem solved using stochastic embedding is provided below:

Algorithm 2 Bayesian Forward Problem with Stochastic Embedding and PMIE

Input: Model $m(u,\theta)$, Covariance matrix Σ_ε, Observed data y

Stochastic Embedding: Introduce uncertain model error term ε as discrepancy between actual and model output: $y = m(u,\theta) + \varepsilon$

Principle of Maximum Information Entropy (PMIE): Establish maximum-entropy probability model for error term ε, with $\mu_\varepsilon = m(u,\theta)$ and Σ_ε:

If $\varepsilon \sim \mathcal{N}(\mu_\varepsilon, \Sigma_\varepsilon)$, then $p(\varepsilon) \propto \exp\left(-\frac{1}{2}(\varepsilon - \mu_\varepsilon)^T \Sigma_\varepsilon^{-1}(\varepsilon - \mu_\varepsilon)\right)$ and $y \sim \mathcal{N}(m(u,\theta), \Sigma_\varepsilon)$

Probabilistic Forward Model: Obtain the probabilistic forward model using deterministic model and error term distribution. Perform forward predictions while quantifying uncertainties, considering uncertainties in parameters and input data.

$p(y|u,\theta) = \frac{1}{(2\pi)^{N/2}|\Sigma_\varepsilon|^{1/2}} \exp\left(-\frac{1}{2}(y - m(u,\theta))^T \Sigma_\varepsilon^{-1}(y - m(u,\theta))\right)$

Output: Likelihood distribution $p(y|u,\theta)$

Implementing the forward problem (prediction with uncertainty quantification) can be developed as follows:

- Model Development: First of all, the DT requires well-defined mathematical or computational models that capture the relationships between the system's inputs and outputs. Importantly, these models incorporate not only the deterministic aspects but also quantify the uncertainties associated with both models parameters and inputs.

- Input Data: The DT receives input data representing the current state of the physical system. This data may include sensor measurements, environmental conditions, user inputs, or any relevant variables affecting the system. The DT acknowledges that these input data also have associated uncertainties.

- Forward Prediction with Uncertainty: Using the model and the current input data, the DT performs forward predictions while quantifying the uncertainties. This involves considering the uncertainties in both the model parameters and the input data to estimate future states or behaviours of the system.

- Output Visualisation: The DT presents the predicted outcomes along with quantified uncertainties to users in a meaningful and accessible way. The output visualisations include not only the expected behaviour of the system but also the degree of uncertainty associated with those predictions, employing means such as confidence intervals, uncertainty bands, PDFs, or alternative representations of uncertainty to depict the spectrum of potential outcomes (Figure 7.2, reproduced from [136]).

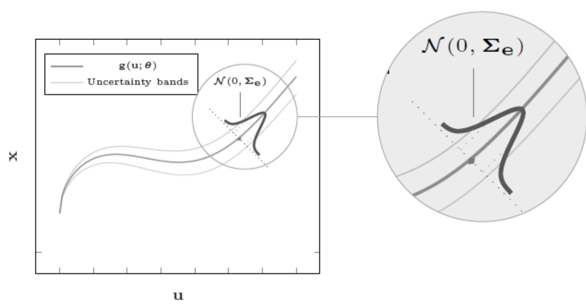


Figure 7.2: Visual representation of uncertainty in the forward problem.

7.2 The inverse problem: model updating

In contrast, in the model updating problem, the focus is to update the *prior* information about the value of a set of uncertain model parameters $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ from a parameterised model $m(\theta, u, e) \in \mathcal{M} \subset \mathbb{R}^{\mathcal{M}}$ where $\mathcal{M}$ is the model class, based on the information given by the observed data y [139]. The updating of a model using information gathered from sensors can be understood as an *inverse problem* [136] and expressed as in Equation (7.4):

$$p(\theta|y,\mathcal{M}) = \frac{p(y|\theta,\mathcal{M}) \cdot p(\theta|\mathcal{M})}{p(y|\mathcal{M})} \tag{7.4}$$

where $p(\theta|y,\mathcal{M})$ is the *posterior probability density function* (PDF) of the uncertain parameters θ, $p(y|\theta,\mathcal{M})$ is the *likelihood function*, $p(\theta|\mathcal{M})$ is the *prior*, and $p(y|\mathcal{M})$ is known as the *evidence*. In the proposed framework, the model class $\mathcal{M}$ is given by $\mathrm{m}(\theta, u, e)$ and the likelihood function is defined by the probability model chosen for the error ε.

Following the Bayesian formulation, the solution is not a single value of θ; on the contrary, Bayes' theorem (see Equation (7.4)) takes the initial quantification of the plausibility of θ, which is expressed by the *prior* PDF $p(\theta|\mathcal{M})$, and updates this plausibility using the information in the data y through the *likelihood function* $p(y|\theta,\mathcal{M})$ to obtain the *posterior* PDF of the model parameters $p(\theta|y,\mathcal{M})$.

In Bayesian inference, the posterior probability is conventionally proportional to the product of the prior probability and the likelihood function, when the evidence (denominator term) is infeasible, challenging to compute or unknown, so Equation (7.4) can be reduced to:

$$p(\theta|y,\mathcal{M}) \propto p(y|\theta,\mathcal{M}) \cdot p(\theta|\mathcal{M}) \tag{7.5}$$

Several methods, including Markov Chain Montecarlo (MCMC) [140] Metropolis-Hastings [141] method, have gained popularity for their effectiveness in estimating the PDF when the likelihood function is known, without the need to compute the evidence.

When incorporating the inverse problem for parameter calibration and updating into the workflow of a DT, the steps for its realisation can be outlined as follows:

- Initial parameterisation: When a DT is initially deployed, it often has limited or no measurements from the physical system it represents. Without real-world data, the DT's models cannot be calibrated or trained to accurately reflect the behaviour of the system. For this reason, in many cases the DT starts with an initial set of model parameters. These parameters may be based on prior knowledge, default values, or historical data. Another approach to setting up a DT is to create an initial training dataset through statistical procedures or AI-generated data. This artificial dataset serves as a foundation for training the DT's models parameters when real-world data is scarce or unavailable initially.

- Observation data: The DT collects observation data from the physical system in real time or through periodic measurements. This data represents the actual behaviour or state of the system as it evolves.

- Inverse problem solving: Bayesian inference is applied to solve the inverse problem. The DT uses the observed data and the model to update and refine its beliefs about the model parameters. This involves finding the parameter values that are most consistent with the observed data.

- Parameter update: As new data becomes available, the DT continually updates its parameter estimates. This dynamic adjustment allows the DT to adapt to changing conditions and improve the accuracy of its model.

- Feedback loop: The DT can incorporate a feedback loop that continually refines its model and parameter estimates based on real-world observations. This iterative process helps the DT maintain a high level of accuracy in representing the physical system.

An algorithmic description of the inverse problem solved using Metropolis-Hastings is provided below:

Algorithm 3 Bayesian Inverse Problem with Metropolis-Hastings

---

Input: Observed data y, Model $m(u, \theta)$ of the model class $\mathcal{M}$, Prior distribution $p(\theta)$
Sample initial parameter $\theta^{(0)}$ from prior distribution: $\theta^{(0)} \sim p(\theta)$
for $k = 1$ to N_{samples} **do**
  Propose new parameter value θ^* from proposal distribution $q(\theta^*|\theta^{(k-1)})$
  Compute acceptance probability $\alpha = \min\left(1, \frac{p(y|\theta^*)}{p(y|\theta^{(k-1)})} \frac{p(\theta^*)}{p(\theta^{(k-1)})} \frac{q(\theta^{(k-1)}|\theta^*)}{q(\theta^*|\theta^{(k-1)})}\right)$
  Generate random number r from uniform distribution on $\mathcal{U}[0, 1]$
  if $r \leq \alpha$ **then**
    Accept proposed parameter: $\theta^{(k)} = \theta^*$
  else
    Reject proposed parameter: $\theta^{(k)} = \theta^{(k-1)}$
  end if
end for
Posterior Calculation: Calculate posterior distribution using Bayes' theorem:
  $p(\theta|y) = (p(y|u, \theta) \cdot p(\theta))/p(y)$
  $p(\theta|y) \propto p(y|u, \theta) \cdot p(\theta)$ ▷ Bayesian inference up to proportionality
Compute posterior distribution: $p(\theta|y) \approx \frac{1}{N_{\text{samples}}} \sum_{k=1}^{N_{\text{samples}}} \delta(\theta - \theta^{(k)})$ ▷ With δ being the Dirac function centered at the current parameter sample
Output: Posterior distribution $p(\theta|y)$

Chapter 8

Digital Twin workflow management

The incorporation of a workflow in a DT is indispensable for simplifying the complexity inherent in mirroring real-world systems and processes. By systematically organising how data is collected, processed, and analysed, the workflow management ensures that a DT can accurately simulate and predict the behaviour of its physical counterpart in real time. This structured approach is crucial for handling the vast and varied data streams from sensors and other sources, enabling the DT to react promptly and effectively to changes or anomalies detected in the physical system. Without a well-defined workflow, the potential of a DT to optimise operational processes, reduce costs, and enhance decision making would be significantly diminished.

8.1 DT workflow description

In the context of a DT, a workflow refers to a predefined number of tasks, actions, or processes that are designed to be executed and timed to achieve a particular objective or outcome related to the monitoring, operation, maintenance, or optimisation of the physical system or asset represented by the DT.

The DT needs to operate through a carefully designed workflow that coordinates the activities of its constituent components. The system's autonomous decision making is geared towards optimising these components, progressively enhancing the precision and efficiency of the DT in mirroring its physical counterpart. In this context, the design of the workflow should emphasise the collaborative interactions among the DT components, enabling autonomy and the capacity to offer/request services. This ensures that the system remains adaptable, flexible, and configurable for different scenarios.

The DT workflow is often automated to ensure that the decisions and actions involved are carried out efficiently. This way, the orchestration of a DT workflow becomes a systematic process that encompasses several essential steps to ensure efficient decision making and action execution. It typically involves continuous monitoring, data ingestion, storage, integration, modelling, real-time data analysis, predictive analytics, decision support, autonomous control, visualisation, feedback, security, and scalability. It begins with the collection and preprocessing of data from various sources, followed by the calibration of mathematical models and simulations that represent the physical system. Real-time data analysis and predictive analytics provide valuable insights that assist decision-support systems in making well-informed choices. In some cases, DTs can autonomously control aspects of the physical system, leveraging expert-knowledge predefined rules or AI-optimised strategies. Visualisation tools facilitate understanding, and continuous monitoring and updating ensure that the DT remains accurate and aligned with real-world conditions. Finally, security, compliance, and error-handling mechanisms are integrated seamlessly into the workflow to safeguard the system's integrity and reliability.

The design of the workflow may vary depending on the specific application and objectives, encompassing linear sequences, feedback loops, iterative processes, and adaptive behaviours. In simple cases, DT workflows follow a linear sequence of tasks or actions. For example, in a predictive maintenance workflow, the DT may monitor sensor data continuously. When certain conditions are met, it triggers maintenance tasks and once these tasks are completed, the workflow may return to monitoring mode, waiting for the next set of conditions to trigger further actions. This process can be sequential and linear. In other cases, workflows within a DT can be iterative, meaning they involve repetitive cycles of data collection, analysis, decision making, and action, such as in SHM operations. However, DTs often incorporate feedback loops to continuously adapt and optimise to rapidly changing conditions. One step further is the adaptive workflow, where the DT can adapt its workflow by modifying the sequence of tasks or actions to respond to these changes effectively.

The workflow of a DT may not need a structured approach and in simpler cases, batch processing with periodic updates may be sufficient. However, in most cases, the level of complexity demands the automation of processes within the DT workflow to ensure efficient and accurate multitasking. Several methodologies and frameworks are adopted to manage workflows in DTs and can be categorised into different groups. Mathematical models and logical algorithms encompass agent-based modelling [142], graphical godels such as Bayesian networks [85], and Petri nets [46], while computing methodologies include workflow management systems [143] and service-oriented architectures [10], among others.

8.2 DT Workflow management by Petri nets

Among the various methods available for managing the workflow of a DT in the context of the civil engineering domain, Petri nets stand out as a valuable choice. They provide flexibility for modelling complex processes, offer clear visual representations, handle concurrency, and manage resources efficiently. Their analysis capabilities ensure correctness and reliability, which are paramount in managing critical engineering assets. Furthermore, Petri nets are adaptable to changing requirements and seamlessly integrate with other tools, as their standardised foundations promote consistency and interoperability. The sequence for implementing the management of a DT workflow through Petri nets can be outlined as follows:

1. Creation of places (nodes): Each workflow step is represented as a place (circle) in the Petri net diagram. For example, there can be a place for 'Data Collection', 'Analysis', 'Decision Making', etc.

2. Definition of transitions: The transitions (squares or bars) are located between places. Transitions constitute the actions or events that move the workflow forward by changing the distribution of tokens across places. In the context of a DT, transitions could be triggered by events such as 'data availability', 'thresholds being reached', or 'decision points'.

3. Conducting the flow with arcs: Arcs (arrows) connect places to transitions and transitions to places, indicating the flow of tokens and how transitions affect the system states. Arcs can also carry weights that determine the dynamics of token movement. For each arc connecting a place to a transition, the weight of an arc is equal to the number of tokens that are required for a transition to fire. Similarly, for each arc connecting a transition to a place, the weight of the arc decides how many tokens will be produced and added to the next place when the transition fires.

4. Addition of Tokens (Marking): The tokens (dots) are used to indicate the current state of each place and represent progress or completion of a step. For example, if data has been collected, a token is located in the 'Data Collection' place. The initial distribution of tokens across the places is known as the initial marking. If an arc does not specify a weight, it is assumed to have a weight equal to 1, which means that one token is consumed or produced.

These concepts are depicted in the following algorithm, which illustrates the simulation of a workflow managed by a Petri net:

Algorithm 4 Petri net Simulation for workflow management

```
Input:
  P                                    ▷ Set of places p
  T                                    ▷ Set of transitions t
  E ⊆ (P × T) ∪ (T × P)                ▷ Flow relation or set of weighted arcs
  M_0 : P → ℕ                          ▷ Initial marking of tokens
Output:
  M : P → ℕ                            ▷ Marking after simulation
  Initialize marking with the initial marking
  M ← M_0
  while there exists an enabled transition t in T do
      choose an enabled transition t
      for p ∈ P such that (p, t) ∈ E do
          M(p) ← M(p) − 1              ▷ Consume a token from each input place
      end for
      for p ∈ P such that (t, p) ∈ E do
          M(p) ← M(p) + 1              ▷ Produce a token for each output place
      end for
  end while
  return M                             ▷ Return the marking after simulation
```

Figure 8.1 illustrates an example of a Petri net that manages the workflow of a maintenance and operations policy of a DT structural system, with the aforementioned sequence implemented. The workflow is initiated with the arrival of data from real-time monitoring, representing the beginning of the process. The workflow progresses to the data collection node where raw data is prepared for further processing. The subsequent transition, labelled as 'data and models' fusion' is a critical juncture where collected data is integrated with existing models. This fusion process updates the data and the models to reflect the current state of the structural system accurately, ensuring that the analysis is based on the most recent and comprehensive information.

Subsequently, a critical transition occurs with the damage diagnostic analysis which determines the structural health, specifically assessing if there is any damage and the extent of it. The outcome of this analysis leads to one of three exclusive paths, each represented by an arc leading to a different node based on the condition of the structure. If no damage is detected, the workflow transitions to a node representing a healthy state, indicating that no immediate action is required and the system can continue its operation unaltered. However, if the analysis identifies moderate damage, the workflow shifts towards a node indicating a moderate damage state. This state activates the DSS through the DT dashboard, signalling the need for intervention and possibly corrective actions to address the identified damage. In cases where the damage is assessed as severe, the workflow moves to a node representing a severe damage state. This condition necessitates autonomous actions for immediate response to ensure safety and structural integrity, in addition to triggering the DSS on the dashboard for further decision making and action planning.

The final stage in the workflow involves a prognostics analysis, aiming to forecast the future condition of the structure based on the current data. This analysis updates the maintenance and operations policy of the DT system, incorporating all relevant data into the system's database. With the prognosis complete, the system is rearmed, ready to begin a new cycle of data monitoring and analysis. This loop ensures continuous monitoring and updating, allowing for real-time responses to changes in the structural system's condition.

By modelling the workflow in a Petri net, the dynamic and complex process of maintaining and operating a DT structural system is effectively captured. This model facilitates a structured approach to decision making, leveraging real-time data and advanced analysis to optimise the structural system's functioning, ensuring efficiency and reliability. It has been illustrated how Petri nets are able to manage concurrency, synchronisation and parallelism. Moreover, Petri nets also allow for dynamic workflow changes, with arcs, nodes and transitions removed or added as needed to adapt to evolving DT requirements.

Petri nets provide clear visibility into the workflow's progress and can be used for performance analysis and optimisation. Regarding performance analysis, they help identify bottlenecks, delays, and resource limitations in a system. Additionally, they can quantify system performance metrics, including throughput, response times, and resource utilisation including logistics in the number of tokens and weight of the arcs. Additionally, by exploring different strategies, configurations, or resource allocations, they facilitate the

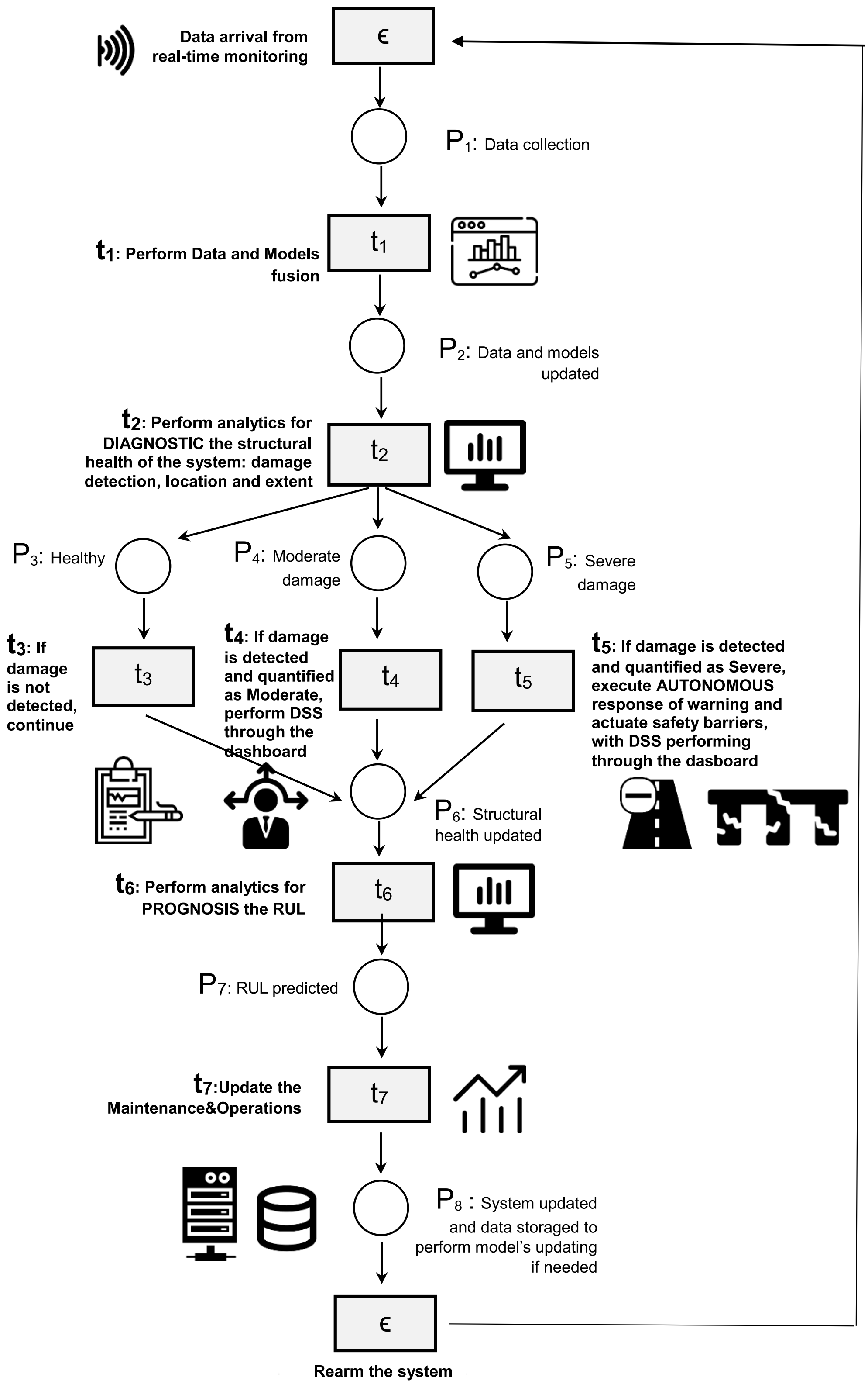


Figure 8.1: Illustrative example of a basic PN managing the workflow of a DT structural system

optimisation of the system behaviour.

While Petri nets are often represented and analysed mathematically using matrix multiplications, they can also be implemented in a more object-oriented and computationally efficient manner, particularly in computer programming contexts. This approach involves representing Petri nets as objects and using object-oriented programming (OOP) principles for their efficient operation. Within this approach, there is no explicit performance of matrix multiplications, which can be highly inefficient and computationally resource-consuming. Instead, the behaviour of the Petri net is simulated by executing methods and updating object states based on rules and conditions. Following the OOP methodology, classes and objects represent places, transitions, and tokens, following these steps:

1. Definition of classes: A *Place class* represents places in the Petri net and each place object contains information about its state (number of tokens) and properties. A *Transition class* symbolises transitions, where transition objects may contain information about their conditions and effects on places and a *Token class* represents the tokens, with token objects associated with places.

2. Creation of objects: The objects of the *Place*, *Transition*, and *Token* classes are instantiated based on the Petri net's structure.

3. Definition of methods: Methods within classes are defined to simulate transitions, token movements, and changes in place states. These methods may include "fire" methods for transitions, "addToken" and "removeToken" methods for places, and other repetitive tasks to be performed within the workflow.

4. Control logic implementation: Object-oriented programming principles are followed to control the execution flow, including the firing of transitions based on conditions and token availability.

To ensure the optimal operation of the OOP implementation, it is essential to employ appropriate data structures that enable an efficient organisation and handling of the Petri net components, including lists, dictionaries, or various other collection types. Moreover, continuous optimisation of both data structures and algorithms is crucial for maintaining efficient performance, particularly when dealing with large and complex Petri nets.

Concerning the execution time within the DT framework, it is important to note that not all processes necessarily run simultaneously. The synchronisation of processes does not imply that every component operates in real time or concurrently. Instead, different processes within a DT can have varying execution times and dependencies based on their nature and requirements. The alignment of processes with respect to time can be *real time*, *near real time* or *almost real time*, and *off line* (Figure 8.2).

In real time are the activities often associated with monitoring, data acquisition, and immediate responses to changes in the physical system. These processes operate in sync with real-time events. Near-real-time are the processes operating with a slight delay, but still within an acceptable timeframe to support timely decision making and without the need for immediate interaction with the physical system. It allows for a deeper analysis of the incoming data, enabling complex decision-making processes that take into account broader system states or trends. Finally, offline processes do not necessitate immediate interaction with the physical system either. They involve more complex analytics, such as updating models with historical data analysis, long-term planning, or running simulations without time constraints. In the offline mode, the analytics modules can incorporate extensive datasets to simulate multiple scenarios, evaluate the outcomes of potential improvements, or conduct thorough analyses of system performance over time. This can lead to strategic adjustments and optimisations, allowing for optimum maintenance scheduling during non-critical operation times.

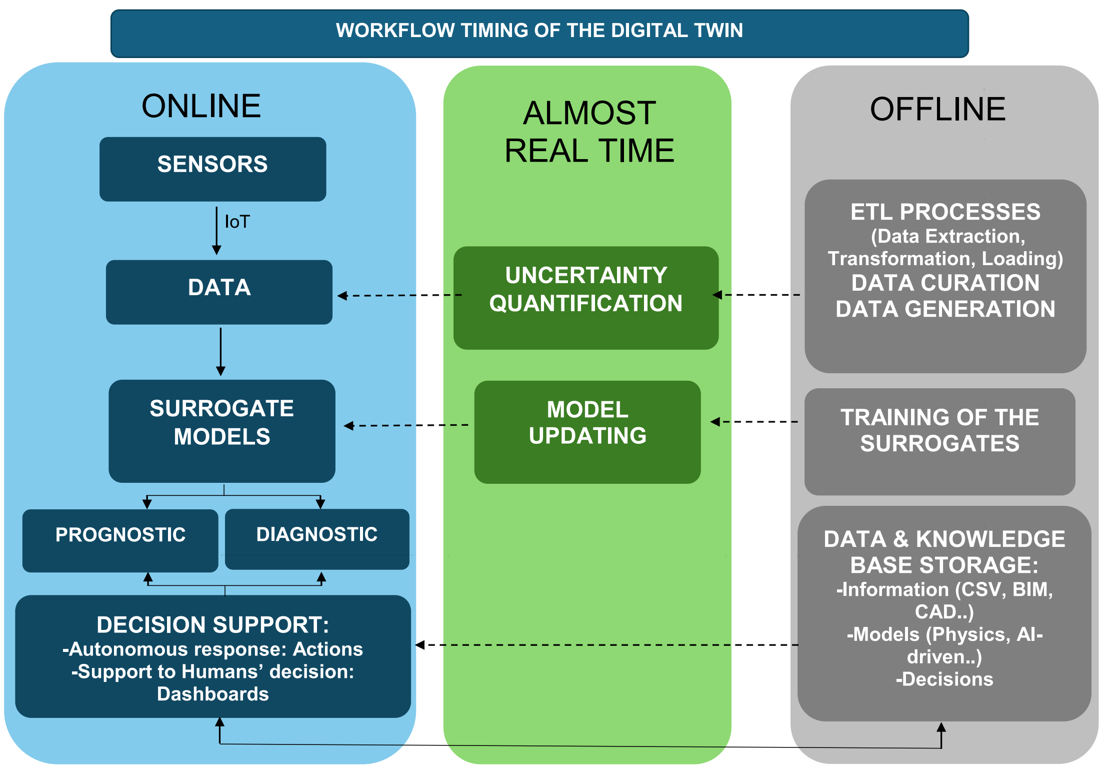


Figure 8.2: Timing of the DT

Incorporating these timing aspects into a DT workflow managed by a Petri net allows for a tailored approach to system monitoring, analysis, and decision making. By choosing the appropriate timing for different aspects of the system's operation, developers can optimise the performance and responsiveness of the DT, ensuring that it meets the specific needs and constraints of the physical system it mirrors. This flexibility in timing ensures that the DT can effectively manage a wide range of applications, from those requiring immediate action to those benefiting from careful, long-term planning.

Timed Petri nets extend the standard Petri net model by integrating time into the framework, allowing for the representation of DT's operations in real time, near real time, and offline modes. In this modality, timed transitions and tokens can be used to model the timing constraints and dynamics inherent to the DT's interaction with its physical counterpart. Real time operations are characterised by minimal firing durations and immediate token processing to ensure instantaneous response to critical system changes. Near real time scenarios slightly extend these durations to accommodate short delays in data processing and decision making, allowing for timely adjustments. For offline activities, longer firing durations reflect the extended analysis periods for maintenance planning and system optimisation. This nuanced approach to timing within Petri nets allows a DT to accurately reflect the state of the physical system it mirrors, ensuring optimal performance and enhanced decision-making capabilities across different time-based operational modes.

Chapter 9

Generative setting for training Digital Twin models

In this chapter, a generative framework has been developed to overcome the data shortage problem that affects the implementation of DT in the civil engineering field, especially in the initial stages, when real-world data is scarce or unavailable to train the DT models. The contribution consists of a comprehensive procedure for data generation based on conditional generative adversarial models (CWGAN-GP) which not only relies on data but also integrates information from physics. To assess the efficacy of the synthetic dataset generated, a range of metrics is proposed to evaluate the generated results both qualitatively and quantitatively.

9.1 Synthetic data and generative techniques

Synthetic data is data generated by simulation, based upon and mirroring properties of an original dataset [144]. This definition allows for the physics-informed character of the data (the simulation source) and the randomness inherent to original processes and how the environment affects the measurements (noise). It should not be overlooked that real data is measured in the actual world, whereas synthetic data is generated in digital environments. The errors present in the data are of two types [145]: the systematic error (that can be the one coming from the measurement equipment and it is considered as bias) and the random error (also called noise, which adds variability to the data but does not affect the average performance of the distribution) [145]. The systematic error can be corrected by calibrating the model and preprocessing the data, whereas the noise is almost unavoidable and confers to the dataset the innate character of the real world.

Statistical procedures facilitate the creation of datasets that adhere to predefined statistical distributions and patterns reflective of the physical system's behaviour. However, they often fall short in capturing the complexity of real-world systems and struggle to generalise effectively. Models trained exclusively on such data tend to underperform in practical applications, as they cannot encompass the full range of variations inherent in the physical system.

On the other hand, AI generative methods employ ML algorithms to synthetically produce data capturing intricate nuances and relationships within the system that might be challenging to capture through traditional statistical procedures. These AI-generated datasets not only provide an initial training ground for the DT's models but also offer the potential for superior model calibration. AI-driven datasets often exhibit a higher degree of fidelity to the real-world system's behaviour, as they allow for a more accurate emulation of complex dynamics. As the DT commences its journey with these initial parameters obtained through AI-generated foundational datasets, it embarks on a path of continuous learning and adaptation. Real-world data gradually replaces the synthetic datasets, and the DT's models evolve to align more closely with the actual system. Through this process (Figure 9.1), the uncertainty associated with the DT's predictions diminishes, and its capacity for accurate representation amplifies.

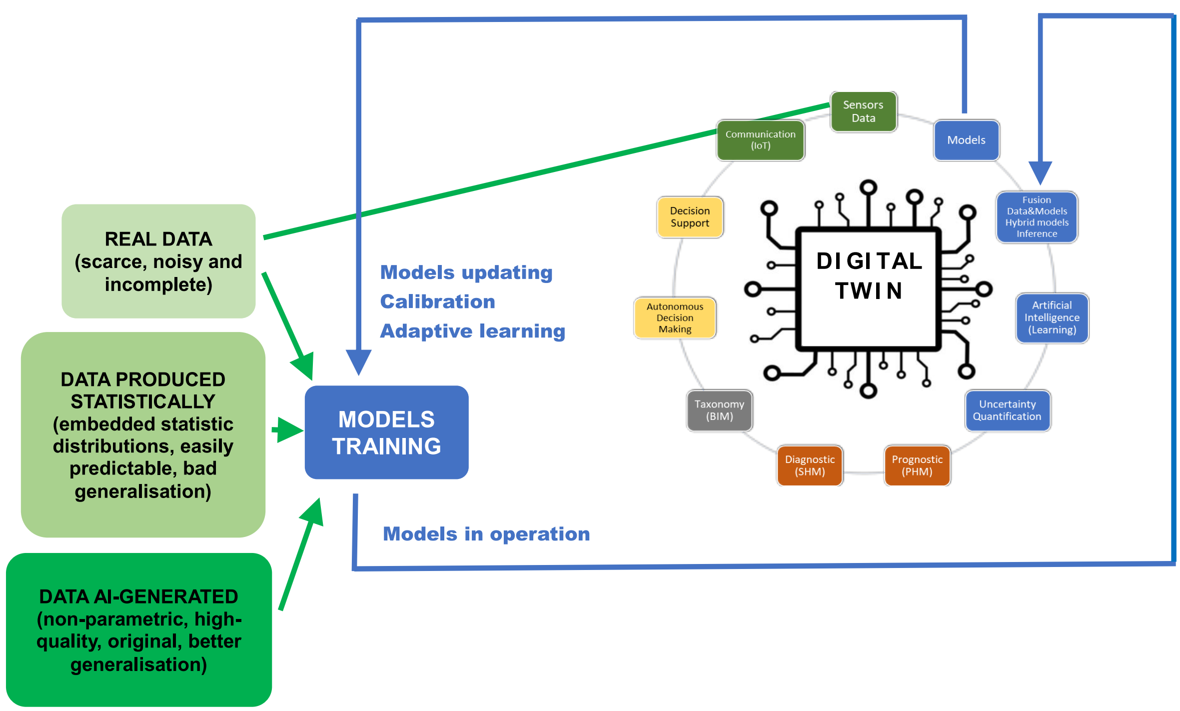


Figure 9.1: Iterative process of data in the DT.

9.2 Initialisation of the DT models

The inception of a DT marks the origin of a complex process aimed at replicating the real-world behaviour of a physical system within a digital environment. Central to this process is the establishment of reliable models with accurate parameters that govern the dynamics and characteristics of the digital representation. However, in the initial stages, real-world data is scarce or unavailable, so the DT needs to rely on initialisation model parameters to commence its operation. This limiting lack of data has been largely claimed by the engineering community, [101, 146–149]. The critical juncture highlighted prompts the question of how these initial values are determined and what impact they bear on the subsequent performance of the DT.

It is a fact that the lack of data in quality and quantity corresponding to the healthy state and, especially, to the damaged state of a structure, is an obstacle in the development of the DT. To enable the utilisation of pre-trained surrogate models for real-time adaptation to various structural scenarios, a substantial amount of data is required. The quality and quantity of this data significantly impact the efficacy of these models.

At the heart of this challenge lies the selection of the model's parameters, which can be done by either fixing them using prior domain knowledge or relying on conventional default values that are representative of the system category being modelled. While this method offers a straightforward starting point, it may lack precision and may not adequately capture the nuances of the particular system modelled. Alternatively, a more reliable and dynamic approach is to initialise the models using foundational datasets produced through statistical procedures or generated employing generative AI techniques. This way, by leveraging statistical procedures grounded in numerical models such as FE, or by harnessing generative AI methods, a foundational dataset is created for the purpose of training models and obtaining parameters. This strategy provides the DT with a strong initial foundation, as the initial parameters are informed by a more diverse dataset that is closer to the actual behaviour of the system.

AI generative methods empower the DT to initiate its operation and pave the way for more robust and accurate model updating. The synergy between AI-driven data generation and the evolving nature of the DT's learning process underscores the pivotal role of data-driven techniques in shaping the ultimate efficacy and reliability of the DT. State-of-the-art techniques such as generative adversarial networks (GANs) belonging to the field of AI and DL have come into place to aid the DT technologies in multiple ways [148], especially in tasks such as modelling [150], data augmentation [151], and reconstruction [152], and have been considered a paradigm shift in the generation of synthetic data [153].

The following steps resume the generative approach for initial model deploying, which is also illustrated in Figure 9.2:

1. Data collection from available sources: Gather all available data sources, including real-time data, historical records, numerical simulations, or any other relevant data. If such data is limited, explore

alternative sources that might provide additional insights into the system's behaviour.

2. Creation of a synthetic foundational dataset through a generative setting: When real-time data is insufficient, it becomes necessary to generate an artificial dataset for models training through synthetic means, employing the available data sources such as historical records, simulations, statistical methodologies, and AI techniques. Simulations entail the use of mathematical equations or numerical methods such as FE, whereas statistical procedures are based on predefined statistical distributions. Additionally, AI generative methods can be harnessed to produce data that simulates the physical system's behaviour, faithfully replicating patterns and variations of real-world systems, including unforeseen *black-swan* scenarios. This generative approach contributes to enhancing the model's generalisation capabilities through improved training.

3. DT Models training: The foundational dataset generated by the generative setting is used to train the DT's models. These models can be physics-based, data-driven, or a combination of both in a hybrid approach, depending on the system and its complexity.

4. Initial predictions: With the models trained on the foundational dataset, the DT can start making initial predictions about the system's behaviour. These predictions provide a baseline understanding of how the system is expected to perform.

5. Calibration and adaptive learning: As the DT operates and collects real-world data, it calibrates its models and adapt them based on this new information. The foundational dataset served as a starting point meanwhile the DT's models are continuously updating and improving through adaptive learning.

6. Reducing uncertainty: Over time, as the DT incorporates real-world data, the uncertainty associated with its predictions decreases, and the DT models become more accurate and representative of the actual system.

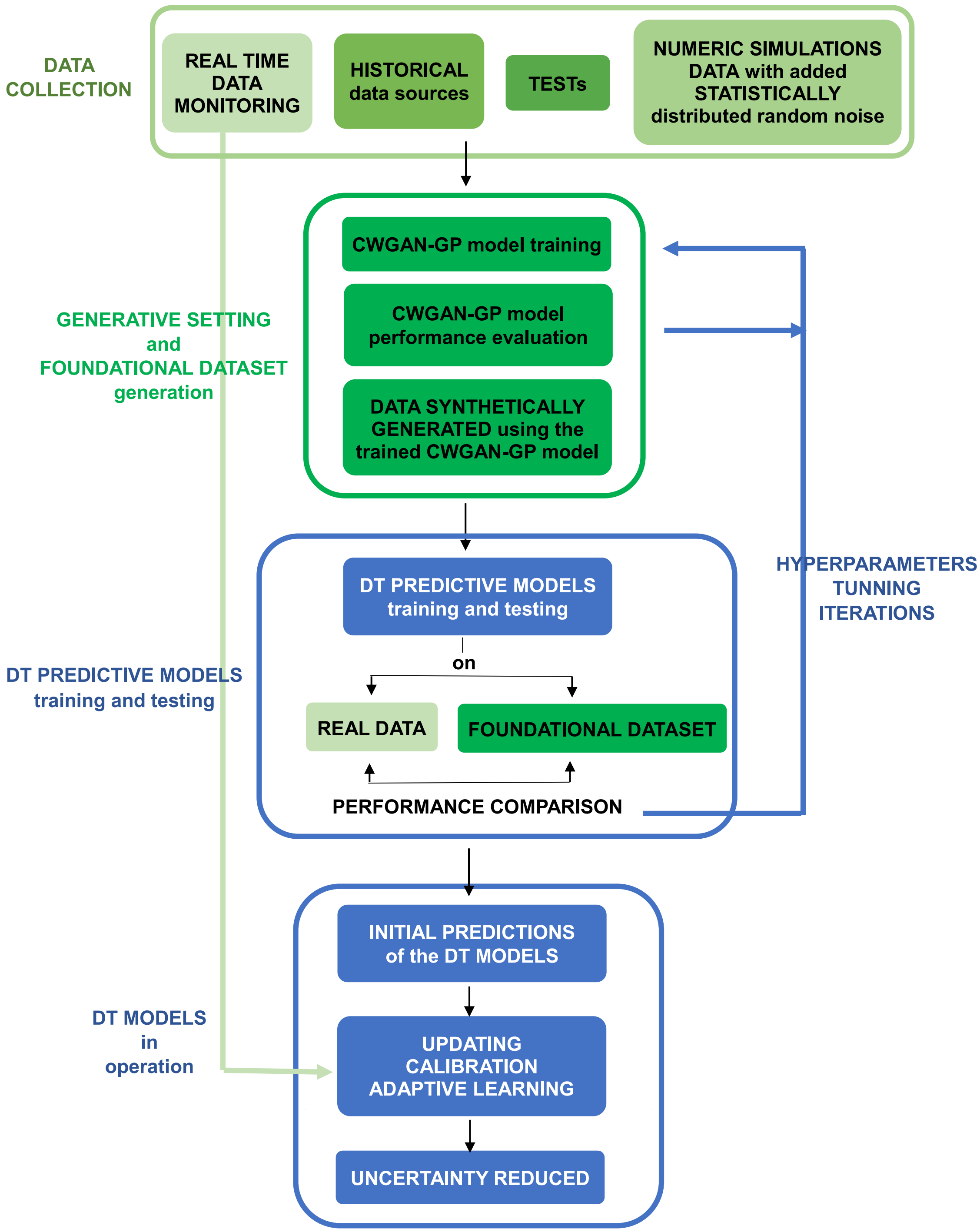


Figure 9.2: Generative approach for initial model deploying.

9.3 Generative setting deployment

The generative setting proposed in this thesis is based on the use of conditional generative adversarial models (CWGAN-GP) which, as opposed to classical simulation-based approaches, do not explicitly counterfeit the physics of the entire system, but only the responses of interest. As the data-driven model of the asset implicitly learns the system's physics from its data, there's no need for explicit prior physics knowledge. Nevertheless, this physics information is indirectly supplied through the physics-informed feature of the method, which can be included in the inputs, in the outputs, or in embedded equations conforming the loss functions of the DL method [118]. This approach results in a hybrid combination of both physics-based and data-driven elements.

In scenarios where the complexity of the system's physics makes impractical its direct inclusion into the model's loss function, the physics guidance can be achieved by embedding through the output neurons, as it is performed in the proposed methodology. The target dataset, representing the physics system's behaviour, will play a central role. During training, the generated data from the CWGAN-GP will be compared to this target dataset, serving as a reference for models evaluation and learning. The generator is guided toward producing samples that align with the physics information in the target dataset. The discriminator, on the other hand, continues to assess the realism of generated samples compared to real data. This way, the loss function is based on the difference between the target values and the data produced by the generative model and it is constructed by minimising the distance between both data distributions following the principles of training of the generative methods exposed. Consequently, the loss function incorporating the physics information through the output ($L_{PI\ CWGAN-GP}$) has the following expression:

$$L_{\text{PI CWGAN-GP}} = L_{\text{CWGAN-GP}}(D, G, \tilde{x}|c, x) + \mu \cdot L_{\text{physics}}(G, \tilde{x}|c, x_{\text{physics}}) \tag{9.1}$$

where:

$L_{\text{CWGAN-GP}}$ represents the adversarial loss,

L_{physics} denotes the physics-informed loss,

D and G are the discriminator and generator networks,

$\tilde{x}|c$ is the input data, which consists of a random vector $\tilde{x}$ conditioned to a certain information c

x is the real data,

x_{physics} is the physics data,

μ is a hyperparameter for balancing the adversarial and physics-informed components of the loss.

The generative model fine-tunes its parameters during training to minimise the differences between the data generated and the target data, accounting for the inherent noise and its variation (heteroscedasticity) present in the system. Once trained, the generative model, in a non-parametric way, releases synthetic data aligned with sensor measurements and the physics of the system. It offers a versatile tool for data generation and facilitates a nuanced understanding of the real system, not only in common scenarios but also incorporating black swan events and *what-if* situations.

Furthermore, the conditioning confers the generative model the capability of exerting precise control over generated outputs and the ability to generate structured results. It is worth noting that generative models rely on random inputs, often referred to as latent space or noise vector, to generate novel and diverse samples. By specifying certain conditions, labels, or values in the latent space, the model is guided to produce outputs that align with specific criteria, ensuring a class-balance generation or a desired outcome. This way, the generation produces tailored and fine-grained results, ensuring domain adaptation and task-specific generation. In addition, conditioning enhances training stability and convergence, contributing to the overall versatility and effectiveness of generative models.

For improved stability, the generative model includes both the Wasserstein distance and the gradient penalty as deployed in Section 5.6.4, leading to enhanced model convergence and higher-quality results. The CWGAN-GP method has been recently employed in the domain of SHM for vibration-based damage diagnostics [154]. However, to the best of the author's knowledge, its application as a generative setting for models' training within the context of the DT and discrete measurements in this specific form has not been explored previously.

A CWGAN-GP (Section 5.6.4) can be built upon various NN architectures, based on the type of data involved and the nature of the problem. FNNs are known for their simplicity in implementation and training, making them a popular choice for generating structured data like images and sequences. However, they may require voluminous architectures to capture intricate relationships within the data. In contrast, CNNs excel at capturing spatial hierarchies and patterns, making them suitable for structured data generation and manipulation and often leading to more lightweight architectures. In the present approach, CNNs are chosen as particularly well suited for handling discrete measurements acquired in the realm of SHM methodologies for the DT application [155].

With the architecture of the generative CWGAN-GP model built on CNNs, the selection of hyperparameters is an essential step in training the DT models effectively. This can be done through methods like grid search, where different combinations of hyperparameters are tested, or through an empirical trial-and-error process.

Additionally, the labelled dataset coming from available sources used for training the CWGAN-GP is crucial, and its creation depends on the specific application. Together with historical data and tests, in simple systems or components, data can be generated from state-space equations. In more complex systems, numerical methods like FE analysis are often employed to simulate the behaviour of the system and generate training data. The labels in this context represent the conditions of the generative CWGAN-GP. The labelled dataset is further elaborated by incorporating noise or other types of perturbation. This technique not only increases the amount of data but also enhances its diversity and more closely replicates the real-word measurements [106]. 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. In order to increase the realism, heteroscedasticity can be introduced, where the variance of the dependent variable increases with increasing values of the dependent variable, as commonly occurs in engineering scenarios, where noise can be input-dependent [156].

The CWGAN-GP is trained on this labeled dataset, which is sourced from available data, and afterwards, it generates the foundational data label-conditioned on which the DT models will be instructed. The algorithm corresponding to the training of the CWGAN-GP is shown as Algorithm 5.

The labels provided to both the generator and discriminator networks serve as conditional information that guides the generation process to generate samples conditioned on specific classes or categories. Within the generator, the conditional labels are concatenated with the random noise vectors as input to the generator network. This combined input (random noise vectors + conditional labels) guides the generator to produce samples that belong to or resemble the specified classes or categories represented by the conditional labels. By conditioning the generator on specific labels, the characteristics or attributes of the generated samples are controlled. The conditional labels are also provided to the discriminator network, typically concatenated with the input real and fake samples. This allows the discriminator to learn to distinguish between real and fake samples belonging to different classes or categories. Conditioning the discriminator on specific labels ensures that it can effectively discriminate between samples based on their conditional attributes.

The latent dimension represents the space in which the generator learns to map the random noise vectors to meaningful samples. A larger latent dimension can potentially allow for more complex variations in the generated samples, while a smaller latent dimension may result in simpler and more limited variations. The choice of latent dimension is a hyperparameter that can affect the quality and diversity of the generated samples.

Algorithm 5 DT Generative setting as a Conditional Wasserstein Generative Adversarial Network with Gradient Penalty (CWANG-GP)

<u>**Offline Training:**</u>

Input: m {Batch size}, n_{critic} {Number of critic iterations per generator iteration}, λ {penalty parameter}, (l_r, β_1, β_2) {Adam optimiser parameters}, num_epochs {Number of epochs}, $labels$

Output: G(), D() {generator and critic's models}

Define $latent\_dim$

for $t = 1, \ldots, num\_epochs$ **do**

 Initialise Generator and Critic's networks with random parameters θ_g, θ_c.

 while θ has not converged **do**

  for $j = 1, \ldots, n_{critic}$ **do**

   Split the dataset into batches of size m

   for $i = 1, \ldots, m$ **do**

    Sample a batch of size m of real data x_i with corresponding labels y_i, $\{(x_i, y_i)\}_{i=1}^{m}\} \sim \mathcal{D}$

    Sample noise $\{(z_i)\}_{i=1}^{m} \sim p_z(z)$

    Sample a random number $\varepsilon \sim \mathbb{N}[0, 1]$

    Generate samples $\tilde{x}_i$ corresponding to m_i labels y_i

    $\tilde{x}_i \leftarrow G(z_i|y_i, \theta_g)$

    Compute gradient penalty $G_p(\theta_c)$ and critic's loss $\mathbb{E}_{\theta_c}$, being W the Wasserstein distance

    $\hat{x}_i \leftarrow \varepsilon x_i + (1 - \varepsilon)\tilde{x}_i$

    $G_p(\theta_c) \leftarrow \frac{1}{m}\Sigma_{i=1}^{m}\left[max\left(\parallel \bigtriangledown_{\hat{x}} W(\hat{x}|y_i, \theta_c) \parallel_2 -1\right)^2\right]$

    $\mathbb{E}(\theta_c) \leftarrow \bigtriangledown_{\theta_c}\left[\frac{1}{m}\Sigma_{i=1}^{m} W(\hat{x}_i|y_i, \theta_c) - \frac{1}{m}\Sigma_{i=1}^{m} W(x_i|y_i, \theta_c)\right] + \lambda G_p(\theta_c)$

   end for

   Update the critic's parameters by ascending its gradient

   $\theta_c \leftarrow Adam(\mathbb{E}_{\theta_c}, \theta_c, l_r, \beta_1, \beta_2)$

  end for

  Execute a single generator training step

  Sample $\{(y_i)\}_{i=1}^{m} \sim \mathcal{D}$ a batch of size m of labels y_i

  Sample noise $\{(z_i)\}_{i=1}^{m} \sim p_z(z)$

  Compute gradient penalty with respect to generator's parameters considering the critic's

  $\mathbb{E}(\theta_g) \leftarrow \bigtriangledown_{\theta_g}\left[\frac{1}{m}\Sigma_{i=1}^{m} W(g(z_i|y_i, \theta_g)|y_i, \theta_g)\right]$

  Update the generator's parameters by descending its gradient

  $\theta_g \leftarrow Adam(-\mathbb{E}_{\theta_g}, \theta_c, l_r, \beta_1, \beta_2)$

 end while

end for

Save the models G() and D()

End training algorithm

<u>**Online Generation:**</u>

Input: $labels$, $num\_samples$

Output: $dataset\_generated$

Initialise the generated samples dataset

$dataset\_generated = [\ ]$

for $s = 1, \ldots, num\_samples$ **do**

 Sample noise $\{(z_s)\} \sim p_z(z)$

 Generate samples $\tilde{x}_s = G(z_s|y_s$ corresponding to the given labels y_s

 $dataset\_generated.append(\tilde{x}_s)$

end for

9.4 Metrics for the generative performance

The goodness of the generative setting and its generated foundational dataset can be approached through both qualitative and quantitative metrics.

A qualitative approach involves visually inspecting the generated samples, serving as one of the primary methods for evaluation. This involves visually examining the generated samples to determine their quality, diversity, and resemblance to the real data. It is worth mentioning that visually evaluating large-size numeric data is more challenging compared to images, whose features can be visually perceived. Thus, the quantitative approach is desirable in such cases through the use of the performance metrics, even when accounting for similarity at the same time that novelty and complexity between distributions is also complex [157].

Quantitative metrics offer objective measures of the model's performance in terms of distributional similarity and sample quality. The metrics adopted in this thesis consist of the aforementioned Wasserstein distance [129] between the generated and the real distribution, the Frechet Inception Distance (FID) [158] and the Structural Similarity Index Measure (SSIM) [157].

The Wasserstein distance, according to Equation (5.17), 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 Wasserstein-2 distance, assumes that real and generated data follow a multi-dimensional Gaussian distribution and measures the distance between these two Gaussians in the feature space by calculating their respective mean and variance, as in Equation (9.2). Again, a lower FID indicates greater similarity between the compared data.

$$FID(x, \widetilde{x}) = \|\mu_x - \mu_{\widetilde{x}}\|_2^2 + Tr(\sigma_x + \sigma_{\widetilde{x}} - 2(\sigma_x \sigma_{\widetilde{x}})^{\frac{1}{2}}) \tag{9.2}$$

where μ and σ stand respectively for the mean and 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 two aspects: creativity and diversity. The score obtained ranges from 0 to 1, with 1 indicating exact similarity and 0 representing complete dissimilarity (Equation (9.3)). In the context of GANs, the objective is not to generate outputs that are equal to the real data, so a maximum score of 0.8 is often considered.

$$SSIM(x, \widetilde{x}) = \frac{(2\mu_x \mu_{\widetilde{x}} + C_1)(2\sigma_{x\widetilde{x}} + C_2)}{(\mu_x^2 + \mu_{\widetilde{x}}^2 + C_1)(\sigma_x^2 + \sigma_{\widetilde{x}}^2 + C_2)} \tag{9.3}$$

where σ_x and $\sigma_{\widetilde{x}}$ are the variances of the compared distributions, $\sigma_{x\widetilde{x}}$ is the covariance, and C_1 and C_2 are constants to stabilize the division, typically set to 0.01 and 0.03 respectively.

Chapter 10

Damage assessment models strategy for Digital Twins

Damage assessment is one of the main capabilities of a DT in the civil engineering domain, enabling the implementation of an optimised maintenance strategy for the structural health of the systems, ultimately contributing to their efficient lifecycle management. Furthermore, the damage assessment capability empowers proactive measures through real-time informed decisions whenever it is performed online and automatically, leading to enhanced security and productivity. As detailed in Section 5.5, a robust damage evaluation in civil engineering requires the deployment of a series of models capable of addressing the four levels of the damage assessment hierarchy proposed by Rytter [103]: Level 1 (Damage detection), Level 2 (damage location), Level 3 (damage extent) and Level 4 (damage prediction).

In this thesis, it is presented a 4-level damage assessment strategy trained through a generative model virtualising the observed system. This approach enables a full evaluation of the damage, spanning from detection to prediction, including location and quantification. The strategy is implemented through a pipeline of models that advance through the damage assessment hierarchy. The architecture of these predictive models is consistent with the generative model, adapted to perform classification or regression tasks specific to each stage of the damage assessment process.

10.1 Effective model implementation for DT operability

Models, as generic representations of systems and processes, are of crucial importance in the DT technology [159] as they allow the generation of data and the simulation of the physical system or process under different conditions, anticipating unlikely scenarios and identifying potential issues that may arise. Moreover, models are leveraged to make predictions about the future based on historical and real-time data [160].

The definition of the models must be performed before the DT commences operation. In a hybrid approach, these models are tailored specifically or selected from existing analytical formulas or methods for the physics-driven component, and trained with relevant information for the data-driven constituent. Updates to the models occur either on a time-based schedule or according to specific criteria [161], which varies depending on the application. Factors such as parameter evolution, changing ambient conditions, performance metrics, or data drift may necessitate model updates to ensure they adapt to evolving patterns.

Models are trained offline but are designed to work online [162], as the DT should operate ideally in real-time [163] to maintain the synchrony between the real and the virtual twin. However, current technological capabilities often fall short of achieving real-time operations, thus 'almost real-time' serves as a more accurate descriptor. While advancements have been made in these areas, challenges such as data latency, computational complexity, and the need for high-speed data processing continue to hinder real-time implementation. Nowadays, a minimal delay but not instantaneous response remains a practical and achievable goal for many DT applications.

As mentioned, to achieve this real-time capability computational complexity needs to be reduced. For this reason, the models employed by the DT need to be as simple and computationally efficient as possible but with enough accuracy to give reliable outputs. These types of models are known as *surrogate* or *metamodels* [164], as a simplified mathematical approximation of the original ones which are complex and computationally expensive. They are adopted not only to simulate the behaviour of the real twin but also to optimise complex processes or systems, perform sensitivity analysis, and explore the design space.

Surrogate models can be developed using various techniques, ranging from statistical approximations to AI algorithms, with a particular emphasis on DL networks. These models are trained using data derived from the original model and are subsequently validated using real-world data obtained from monitoring activities, tests, or historical records. In this manner, surrogate models are considered data-driven models because they rely on empirical data rather than theoretical principles, without explicit modeling of underlying physical processes. The accuracy of the surrogate models will depend on the complexity of the original model, the method employed to create it, and particularly, the amount and quality of the data.

To effectively train these models, data with sufficient relevance, quality, and quantity is needed in advance, something that is not always feasible in the civil engineering sector where testing is expensive and time-consuming, difficult to scale, and historic data is scarce and incomplete [165], often ignoring the interoperability standards (such as BIM), ontologies (such as the IFC), and formats (as XML, CSV, etc.) [166]. Data scarcity directly impacts the accuracy and reliability of the models employed in the DT for the civil engineering domain, such as damage detection, RUL prediction and maintenance decision-making processes. Consequently, the DT is also affected as the availability of high-quality data is crucial for creating robust digital representations of physical assets or systems. Scaled laboratory tests are often used to validate and calibrate the models, however, they can only reproduce well-defined conditions being unfeasible to replicate real-world situations and capture the variability and complexity of reality.

This challenge is addressed by the development of synthetic datasets has emerged as a paradigm shift [149]. The use of synthetic data is becoming increasingly prevalent for training models [167], directly impacting the DT framework [26, 168]. The advantages of synthetic data come from the fact that can be generated to cover all the needs of the models in size, diversity, complexity and class balance, to cite any [154]. The generated data can be almost identical to the original and statistically replicate real-world information, being highly scalable and smarter than the actual one. The challenges include possible inconsistencies consequence of the attempt to replicate the complexity of the original data. Even when the primary generation of the data is done by simulation from physical models (FE, etc), complemented afterwards with randomness coming from statistical approximations or DL methods, real-world data is required to test the models once they have been trained and, again, its quantity and quantity will determine the goodness of the synthetic dataset. The alignment of both distributions (real and synthetic) has to be good enough to avoid introducing bias [169]. Besides, the objective of creating synthetic datasets is to mimic real-world data, but it should only resemble them, not be an exact duplicate. Consequently, synthetic data may not include outliers that might be present in the real world and could have some relevance.

10.2 Model functionalities for the damage assessment strategy of the DT

The four-level damage assessment strategy is constructed over a pipeline of four predictive models, with each model dedicated to a specific level. Initially, a binary classifier is employed for damage detection in the first level, followed by a multiclass classifier for damage localisation at the second level. Subsequently, regressors are developed for the third and fourth levels of assessment. These models are DL hybrid models, as surrogates trained with physics-informed foundational datasets.

The data required to train these models for the damage assessment strategy has been produced by the generative setting detailed in Chapter 9, which provided a series of foundational datasets to supply the entire pipeline. As already mentioned in Section 5.6.4, when a conditional GAN is used to generate class-balanced datasets, the generator is constrained on specific class labels to control the characteristics of the generated samples. Taking advantage of this flexibility, the condition will be set to binary to generate a class-balanced dataset for binary classification tasks such as damage detection at Level 1. In this case, there is one label with a binary value: healthy (0) or damaged (1), corresponding to the structural state. When the task is a multiclass classification as occurs at Level 2: damage location, the condition will have multiple labels, each of them corresponding to a binary value of 0 (healthy) or 1 (damaged) at each of the monitored critical points of the structure.

The models' architecture relies on CNNs, chosen for their lightweight configuration suitable for tasks involving spatial pattern recognition. This configuration ensures straightforward yet accurate operation. In Table 10.1 there is a description of the predictive models operating in the presented damage assessment strategy. The choice of the last activation function is essential as it determines the format of the model's output and influences the loss function during training.

Table 10.1: Damage assessment strategy.

	Level 1: Detection	**Level 2: Localization**	**Level 3: Quantification**	**Level 4: Prediction**
Algorithmic task:	Binary classification	Multiclass classification	Regression	Regression
Last activation function:	Sigmoid	Softmax	-	-
Inputs:	Monitored data	Monitored data	Monitored data	Monitored data
Outputs:	Binary label	Binary label per monitored point	t (to calculate SDI)	t (to calculate RUL)
Loss function:	Binary Cross Entropy	Categorical Cross Entropy	MSE	MSE

SDI: Structural Damage Index, RUL: Remaining Useful Life, MSE: Mean Squared Error

For binary classification problems such as in Level 1: damage detection, where the output should be either 0 (healthy) or 1 (damaged), the sigmoid activation function is used in the final layer. For multiclass classification problems as in Level 2: damage location, where two or more classes are required (at least equal to the number of monitored points), the softmax activation function is employed in the final layer. Softmax converts the raw scores (logits) into probabilities. Finally, for regression problems where the goal is to predict a continuous value such as in Levels 3 and 4, the final layer of the NNs has no activation function, as the last value is calculated directly.

The loss function of the models is also a distinctive feature, as it measures the error between the predicted outputs and the true targets, guiding the optimisation process during training. The binary cross-entropy loss (Equation 10.1) is a specific function tailored for binary classification problems, whereas the categorical cross-entropy (Equation 10.2) measures the dissimilarity between the predicted and the true distributions in a multiclass classification context:

$$L(\hat{y}, y) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right] \tag{10.1}$$

where $\hat{y}$ is the predicted probability, y is the ground truth label, and N is the number of samples.

$$L(\hat{y}, y) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{ij} \log(\hat{y}_{ij}) \tag{10.2}$$

where $\hat{y}$ is the predicted probability distribution over classes, y is the one-hot encoded ground truth label, N is the number of samples, and C is the number of classes.

The loss function used for regression models is the Mean Squared Error (MSE) (Equation 10.3), measuring the average squared difference between the predicted values and the actual values.

$$L(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{10.3}$$

where $\hat{y}$ are the predicted values, y are the ground truth target values, and N is the number of samples.

After being trained on foundational datasets and tested on real monitoring data, the presented pipeline of models not only confirms the effectiveness of the generative procedure over traditional statistical approaches for the training of models but also establishes the DT capability for damage assessment. The acquired abilities, including detection, localisation, quantification, and prediction of damage, empower the DT for informed decision making and the development of maintenance and operation policies designed to prevent failures and optimise asset management for maximum efficiency.

In operation, the pipeline of models is fed with inputs such as sensor-monitored values. Ideally, these inputs encompass various factors such as environmental forces (e.g., wind, waves, tides) or external loads (e.g., traffic, weights) along with structural responses or reactions (e.g., displacements, strains, stresses). The outputs from these models vary depending on the hierarchical level of the task. At Level 1, aimed at damage detection, the output is a binary label indicating the structural health status, distinguishing between a healthy state and the presence of damage. For Level 2, focusing on damage localisation, the outputs consist of multiple labels, each representing the presence or absence of damage at the individual monitored points. Levels 3 and 4 share a common output, denoted as the time of damage (t), which is directly converted into the SDI and the RUL value for the structure, respectively.

The algorithm depicting the training and deployment of the pipeline of models comprising the damage assessment capability of the DT is presented below as Algorithm 6.

Algorithm 6 Pipeline of CNNs Models for the DT Damage Assessment Strategy (Binary Classification, Multiclass Classification, and Regression)

Offline Training:
Input: Training dataset (x, y), learning rate l_r, dropout probability dp, number of epochs n_{epochs}, batch size m
Define the 1D CNN and dense layers architecture.
Initialise the 1D CNN model parameters: convolutional layer weights W_i, biases b_i, dense layer weights W_{dense}, biases b_{dense}, output layer weights W_{output}, bias b_{output}
for $t = 1, \ldots, n_{\text{epochs}}$ **do**
  Split the dataset into batches of size m
  for each batch $(x_{\text{batch}}, y_{\text{batch}})$ in training dataset **do**
    Clear gradients: $\nabla_{W_i} \leftarrow 0$, $\nabla_{b_i} \leftarrow 0$, $\nabla_{W_{\text{dense}}} \leftarrow 0$, $\nabla_{b_{\text{dense}}} \leftarrow 0$, $\nabla_{W_{\text{output}}} \leftarrow 0$, $\nabla_{b_{\text{output}}} \leftarrow 0$
    Compute forward pass output:
      $x_{\text{output}} \leftarrow x_{\text{batch}}$
    for each convolutional layer i **do**
      $x_i \leftarrow x_{\text{output}}$ ▷ Input data for layer i
      for each filter j in layer i **do**
        $conv\_output_{i,j} \leftarrow \text{convolution}(x_i, W_{i,j}) + b_{i,j}$ ▷ Convolution operation
        $conv\_output_{i,j} \leftarrow \text{ReLU}(conv\_output_{i,j})$ ▷ ReLU activation
      end for
      $x_i \leftarrow \text{concatenate}(conv\_output_{i,1}, ..., conv\_output_{i,K_i})$ ▷ Concatenate feature maps
    end for
    if pooling is applied **then**
      for each convolutional layer i **do**
        $z_i \leftarrow \text{pool}(x_i)$ ▷ Pooling operation
      end for
      $x_{\text{output}} \leftarrow z_i$ ▷ Output of the last pooling layer
    end if
    if flattening is applied **then**
      $x_{\text{output}} \leftarrow \text{flatten}(x_{\text{output}})$ ▷ Flatten feature maps
    end if
    Dense layer operation
    Compute dense layer output: $x_{\text{dense}} \leftarrow x_{\text{output}} \cdot W_{\text{dense}} + b_{\text{dense}}$
    $x_{\text{dense}} \leftarrow \text{ReLU}(x_{\text{dense}})$ ▷ ReLU activation
    $x_{\text{dense}} \leftarrow \text{dropout}(x_{\text{dense}}, dp)$ ▷ Apply dropout
    Output layer operation with activation function σ, being $\sigma = sigmoid$ in binary classification, $\sigma = softmax$ in multiclass classification, and $\sigma = 1$ in regression
    Compute last layer output: $y_{\text{output}} \leftarrow \sigma(x_{\text{dense}} \cdot W_{\text{output}} + b_{\text{output}})$ ▷ Output layer operation
    Compute loss L, being $L =$ *binary cross entropy* in binary classification, $L =$ *categorical cross entropy* if multiclass classification, and $L =$ *MSE* in regression
    Compute loss L: $L \leftarrow \text{L}(y_{\text{output}}, y_{\text{batch}})$
    Accumulate batch loss: $batch\_loss \leftarrow batch\_loss + L$
    Compute backward past of gradients of loss with respect to parameters:
    $\nabla_{W_i}, \nabla_{b_i}, \nabla_{W_{\text{dense}}}, \nabla_{b_{\text{dense}}}, \nabla_{W_{\text{output}}}, \nabla_{b_{\text{output}}}$ using backpropagation
  end for
  Compute average epoch loss: $epoch\_loss \leftarrow batch\_loss / num\_batches$
  Update model parameters using Adam optimizer with learning rate l_r: $W_i \leftarrow W_i - l_r \cdot \nabla_{W_i}$, $b_i \leftarrow b_i - l_r \cdot \nabla_{b_i}$, $W_{\text{dense}} \leftarrow W_{\text{dense}} - l_r \cdot \nabla_{W_{\text{dense}}}$, $b_{\text{dense}} \leftarrow b_{\text{dense}} - l_r \cdot \nabla_{b_{\text{dense}}}$, $W_{\text{output}} \leftarrow W_{\text{output}} - l_r \cdot \nabla_{W_{\text{output}}}$, $b_{\text{output}} \leftarrow b_{\text{output}} - l_r \cdot \nabla_{b_{\text{output}}}$
end for
return Trained pipeline of CNN models for damage assessment

Online Damage Assesment:
Input: monitored data
Output: damage status, damage location, SDI, RUL

Part III

Case studies

Chapter 11

Digital Twin of a 2D Metal tower

The first case study focuses on the development of a DT of a 2D tower structure, where a comprehensive technological deployment has been implemented to monitor forces and displacements. This deployment, set to be utilised in the current case study and subsequently in Chapter 12, includes the setup of sensors to capture real-time data, a platform for processing, visualising and storing these data in a database, and an uncertainty quantification analysis to infer the forces acting on the tower. This case study aims to demonstrate the practical implementation of DT technology in structural monitoring, highlighting its role in enhancing operational efficiency and facilitating informed decision making with uncertainty quantification.

11.1 Description

The present case study is a laboratory scale test consisting of a two-story metal structure 0.4m high, monitored using IoT-based sensors. This structure has been designed with fixed supports (representing the foundations) and rigid joints, and is subjected to an unknown horizontal point load (F3x) applied to node 3, as shown in Figure 11.1. The proposed DT of this structure is developed within a structural integrity context to provide online decision support toward excessive deformation under the action of unknown loads.

The structural integrity scenario is conceptualised around the applied force, specifically when F_{3x} exceeds a user-defined threshold force value F_{ξ}, prompting an alarm. The proposed structural DT is expected to autonomously:

(i) Detect the application of force F_{3x} to the structure;

(ii) Assess if the digital state requires updating based on a predefined displacement sensitivity;

(iii) Conduct Bayesian model updating of the virtual entity to match the state of the physical one, including the inference of the unknown force F_{3x} with quantified uncertainty, among others values;

(iv) Determine whether the integrity alarm should be activated based on the inferred values of F_{3x};

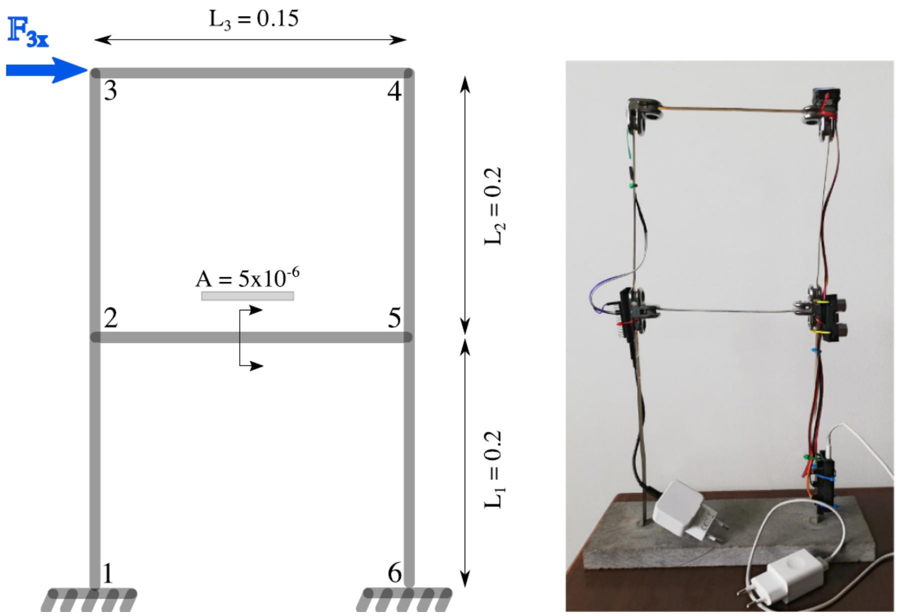


Figure 11.1: Virtual (left) and physical (right) 2D metal tower operated in the case study as a DT proof of concept.

(v) Trigger a pin LED and a digital screen connected to a relay when the alarm is activated, and reset the system after receiving notification from the actuator;

(vi) Maintain continuous communication between the virtual and physical entities, ensuring visualization of the virtual entity in accordance with the state of the physical one;

(vii) Provide instant information on any sensor measurements.

The structural state of the DT is defined by both the measured and modelled elastic displacements of the structural joints. In mathematical terms, the physical state is represented by $\mathrm{s} = \mathrm{s}(\mathrm{e}, \mathrm{w}) = [s_{ix}, s_{iy}, s_{iz}]$ whereas the digital state is described by $\hat{\mathrm{s}} = \hat{\mathrm{s}}(\theta, \mathrm{u}, \mathrm{e}) = [\hat{s}_{ix}, \hat{s}_{iy}, \hat{s}_{iz}]$, respectively, where $i = 1, \ldots, 6$, and x, y, z refers to horizontal, vertical, and rotation displacements. In this study, the measured displacements are taken for joint 5, thus s is specified here as $\mathrm{s} = [0, -, -, -, \{s_{5x}, -, -\}, 0]$. The measurement error vector w is set to $\mathrm{w} = \left[10^{-4}\right]$, expressed in meter units, and relates to the sensitivity of the sensor.

The modelled displacement at joint 5, namely $\hat{\mathrm{s}}$ in the virtual entity, is calculated using the equation of motion in its simplest form for a beam-element frame of one degree of freedom as follows:

$$\hat{\mathrm{s}}(\theta, \mathrm{u}, \mathrm{e}) = [\mathrm{K}]^{-1} \, [\mathrm{F}(\theta)]^{T} \tag{11.1}$$

where $[\mathrm{F}(\theta)] = \left[\mathrm{F}_1, 0, \{\underbrace{F_{3x}}_{\theta}, 0, 0\}, 0, 0, \mathrm{F}_6\right]$ is the vector of applied forces and moments to the joints of the frame structure. It is worth noting that F1 and F6 represent the forces and moments applied to joints 1 and 6, respectively. However, since these joints are fixed, their displacements and rotations are known and set to 0.

In Equation 11.1, the stiffness matrix $[\mathrm{K}] = [K_{ij}], i, j = 1, \ldots, 6$ represents the elastic properties of a plane framed structure with rigid joints. This stiffness matrix is of size 18×18. The values of K_{ij} depend on the geometrical and material input parameters $\mathrm{u} = (L_1, L_2, L_3, E, A, I)$, where L_1, L_2, L_3, and A are depicted in Figure 11.1. The term I represents the cross-sectional moment of inertia, given by $I = 10^{-9}$ m^4, and $E = 9 \cdot 10^8$ $\mathrm{N/m}^2$ denotes the Young's modulus of the material.

Lastly, it is important to notice that in this proof of concept, the influence of environmental variables (e.g., temperature, humidity) is not considered in the structural response. Therefore, $\mathrm{e} = \emptyset$.

11.2 Technology integration

To support the implementation of the DT in both the two case studies of this thesis, a comprehensive integration of technology has been orchestrated to enable the DT deployment. The DT is presented here as a cohesive ensemble of devices, communication tools, and software that function synergistically and especially autonomously (Figure 11.2). This autonomy entails machine-to-machine (M2M) communication, facilitated by the IoT, eliminating the need for direct human intervention. In this context, the physical world is perceived through a sensor network spatially distributed around the monitored structure. Concurrently, devices can interact with the physical world via actuators when commanded. This seamless data exchange between diverse devices and their interoperability, ensured by various protocols and standards, represents a cornerstone of IoT systems, along with the energy constraints that may exist.

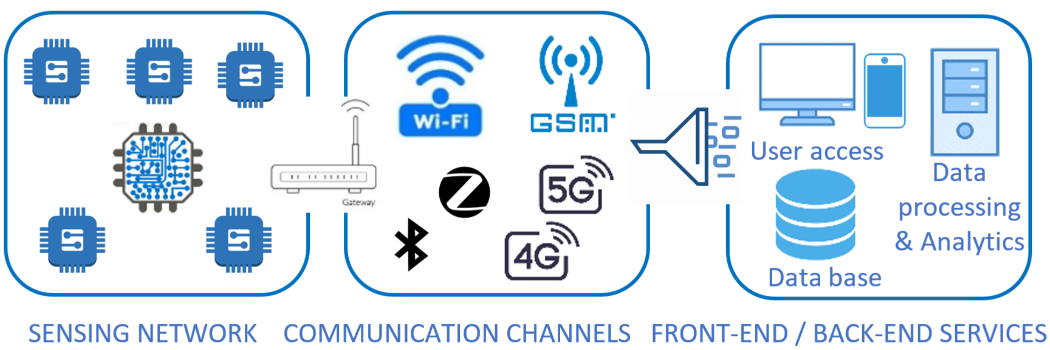


Figure 11.2: Three main blocks comprising the Technology Integration of the proposed DT.

The proposed DT architecture in this case study adheres to the aforementioned principles, comprising three primary components: *smart devices* (incorporating sensors and actuators with microcontrollers and/or microprocessors), *network connectivity* (encompassing the physical media -wired or wireless, gateways, communication channels, and transmission protocols and standards), and an *integration platform*. The integration platform consists of a frontend (facilitating user interaction via an API web) and a backend (responsible for data storage, analytics, IoT applications, and security), serving as a binder of the entire system. These components are elaborated upon in the following sections and are depicted in Figure 11.3 for visual reference.

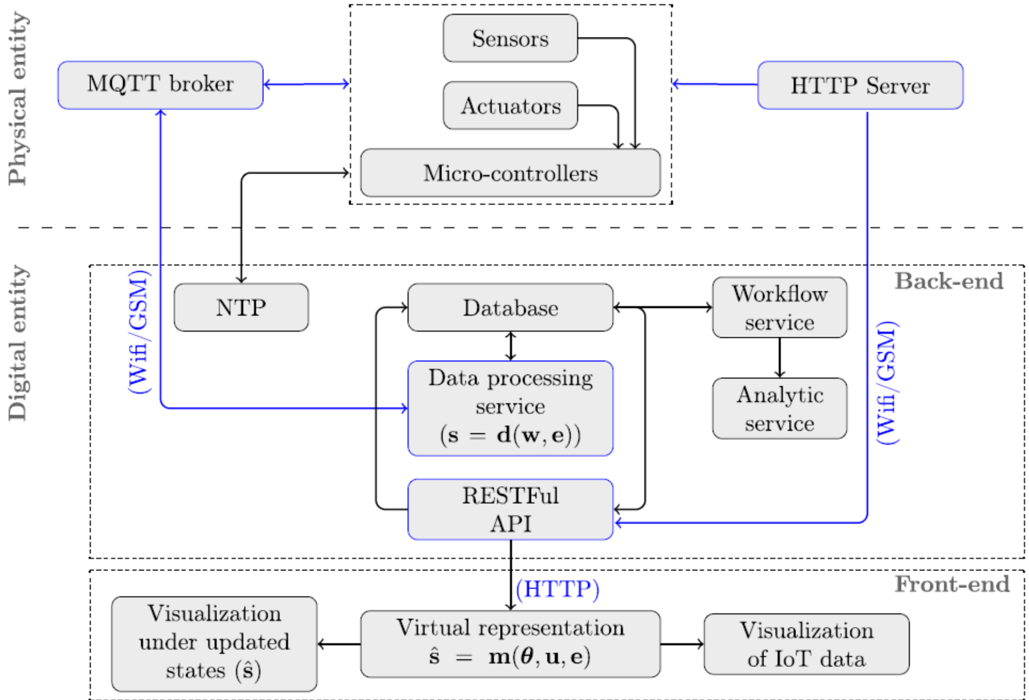


Figure 11.3: Technological integration architecture of the proposed DT.

11.2.1 Smart devices in the sensing network

Smart devices gather data from physical assets through sensors, which transform physical states into signals. These devices also include actuators, which convert signals into physical actions and execute commands when triggered. Integrated into these devices are microcontrollers and microprocessors responsible for processing digital signals from sensors and actuators, respectively, thereby converting them into usable information.

The selection of sensor types, quantities, and placements is critical for capturing the essence of the physical asset and its surroundings [170], and conform the basis of well-known SHM techniques. A wide range of sensors is available, ranging from smart materials to fibre-optic sensors, each suited to specific use cases. However, microelectro-mechanical systems (MEMS) are particularly suitable for wireless network communications within an IoT environment due to their reliability, flexibility, high digital capacity, compact size, and low power consumption [171].

The structure under investigation in this case study is continuously monitored in real-time using two smart devices: a sensor denoted as S_1 positioned at joint 5, and a relay identified as A_1. Their respective locations are illustrated in Figure 11.4. These devices are powered by electricity provided through a micro USB cable, which converts the standard 220V AC input to 5V DC output with a current rating of 500mA. Table 11.1 presents the key properties and specifications of the smart devices utilised in this particular scenario.

Table 11.1: Properties of the sensing and actuator devices in the proposed DT.

ID	Device	Use range	Output	Electrical inputs	Notes
S_1	HC-SR04-P, Ultrasonic sensor	Proximity (< 4 [m])	Distance [m]	3.3 - 5 [V] / 15 [mA]	40 [Hz] pulse-echo ultrasonic signal.
A_1	SRD-05VDC SL-C Relay	Switch (< 10 [A])	250 [V] AC/ 30 [V] DC	5 [V] / 20 [mA]	Generates electro-mechanical switch.

The signals captured by the smart devices are seamlessly integrated into the DT environment through the connection to two distinct IoT boards:

- The first IoT board utilized is the ESP8266 12-e board, which establishes communication with the integration platform via Wi-Fi.

- The second IoT board employed is the ESP32 TTGO T-CALL board, which offers cellular GSM connectivity and supports alternative cellular communication methods such as SMS messaging.

Figure 11.4 offers a visual representation outlining how the smart devices interface with the IoT boards and are physically affixed to the structure. In this particular proof of concept, Wi-Fi serves as the primary communication channel facilitated by the ESP8266 12-e board. This board features a system-on-chip (SoC) architecture, integrating a 32-bit processor that acts as a host for the API web. However, in situations where Wi-Fi connectivity is unavailable, the GSM channel provided by the ESP32 TTGO T-CALL e-board offers an alternative means to establish a connection to the API web.

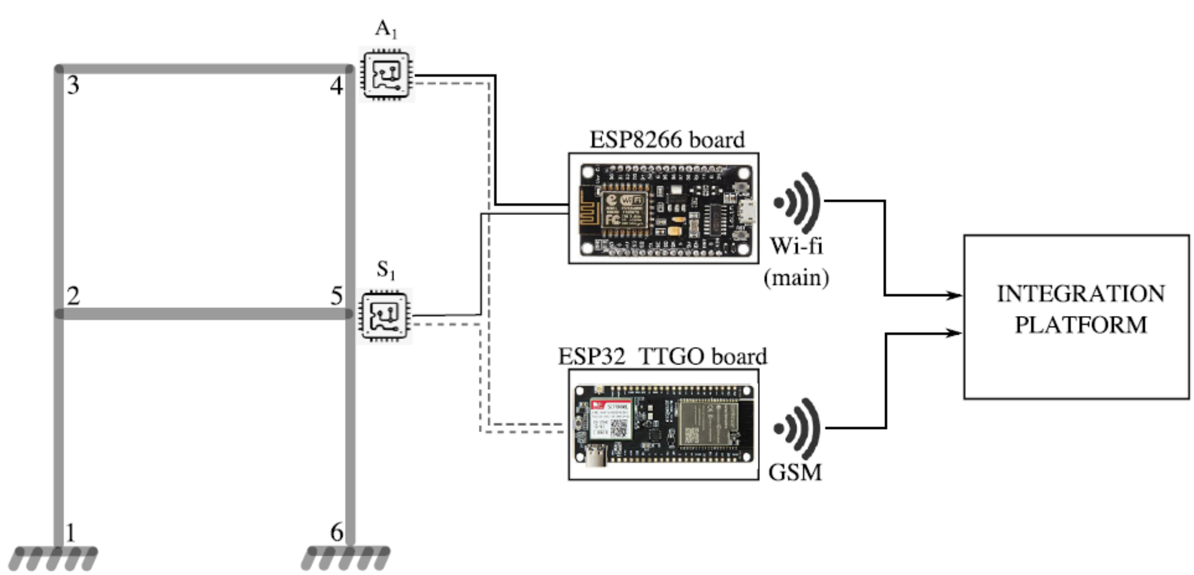


Figure 11.4: Connection scheme of smart devices used in the case study of the 2D metal tower.

11.2.2 Communication and data transmission

In this work, the exchange of data between the devices interconnected within the DT network is enabled by IoT. Various technologies are available for different aspects such as connectivity (e.g. 6LowPAN, IPV6/RPL, IPv6), transport and communication (Wi-Fi, Bluetooth, Satellite -GSM, 3G, 4G, 5G, Radio Frequency, NFC, RFID), data transmission (MQTT, CoAP, HTTP), security (DTLS, TLS, MTLS, SSL) and device management (OMA-LwM2M, OMA-DM), to name the most employed ones. Common standards, such as the OSI model ([172]) and the TCP/IP model ([173]), define the functions and processes used in these communication technologies.

Two communications channels were enabled for connectivity: Wi-Fi and cellular networks, ensuring seamless communication of encrypted information (using SSL) between the physical and the virtual entities. These channels utilise MQTT and HTTP transmission protocols, both adhering to the TCP/IP standard, offering low power consumption, embedded security, and scalability. Further details about the communication and data-transmission aspects are shown in Table 11.2.

Table 11.2: Properties of the data transmission technologies and communication channels in the proposed DT.

Message Transmitting Protocols	MQTT	HTTP
Specification	ISO/IEC 20922:2016	IEEE 802.15.4
Application	M2M & IoT devices	Wireless LAN, broadband Internet access
Standard	TCP/IP	TCP/IP
Methodology	Data-centric	Document-centric
Message size	Small	Large
Default port	1883 or 8883 (over SSL)	80 and 443 (over SSL)

Communication channel	Wi-fi	Cellular network
Specification	Based on IEEE 802.11a/b/g	Based on GSM and GPRS
Application	Wireless LAN connectivity, broadband Internet	Mobile radio connectivity, voice and data services
Band	Worldwide unlicensed 2.4 GHz	850/900/1800/ 1900 MHz (2G)
Topology	Line, ring, star, tree, or mesh	Cells
Transmission distance	approx. 100m	approx. 35km
Max. number of nodes	Unlimited	Unlimited

MQTT serves as the primary communication channel due to its lightweight and rapid message transmission capability. It enables bidirectional connections with a callback option for event reception and smart device reconfiguration, if necessary. Data collected via MQTT are made available to subscribers through an MQTT broker, such as Mosquitto, based on applied policies.

HTTP functions as the secondary channel, mainly for event reception and support of the API web for real-time monitoring status. Through HTTP, each sensor is assigned a unique IP address for identification and communication. Data are encapsulated into IPv6 packets and forwarded to an Apache server operated in Linux when a request is made. The Apache server, working as an open-source server, implements the HTTP/1.1 protocol and functions as a virtual site according to the RFC standard [174]. Although HTTP is less efficient than MQTT for IoT, it facilitates the transmission of heavier information and supports communication with the API web.

These communication channels enable data transmission from sensing nodes through the network to the DT's API web, where data are stored in a database for further processing. Lightweight data interchange formats like XML or JSON are adopted, aligning with the open data formats recommended by the Open Group IoT standard [175].

11.2.3 Web-based integration platform

An inclusive platform has been developed to support interaction among devices, networks, and software within the DT ecosystem in an efficient, reliable, and secure way. This platform is an Application Programming Interface (API), conceived as a set of standards, protocols, and tools that allow different software applications to communicate with each other to request and exchange information between software components from devices or systems.

For the construction of the API, a representational state transfer (REST) [176] software architecture, namely RESTful, is adopted. This architecture effectively manages data from IoT devices via HTTP and provides real-time data access to users [24]. By this means, the API-RESTful platform establishes a bidirectional connection among DT components using a web-based framework with decentralised control [177].

In the presented DT approach, a *Service Oriented Architecture* (SOA) is adopted and applications are conceptualised as *services* exchanged between components through their respective APIs and communication protocols. These services are accessible remotely, allowing them to be acted upon and updated independently.

The *frontend* of the platform provides a user-friendly interface, simplifying the management of available information for end users. It facilitates user interaction with the DT nodes and devices, allowing tasks such as requesting real-time measurements or reconfiguring devices. Indeed, since IoT-based monitoring implies supervising the operation of the sensors, including communications and data conveyed, if a fault is registered in any node of the physical entity, it can be remotely reconfigured or rebooted directly from the API.

Through the *backend*, synchronisation between the platform and the DT smart devices is achieved using a *network time protocol* (NTP), enabling real-time response and event synchronisation. In this study, the platform is hosted on a dedicated server, although alternative hosting on cloud infrastructure is feasible for applications requiring pervasive monitoring.

The core computing modules of the DT are hosted in the backend as services. Within the proposed DT framework, there are three key services: *dataprocessing*, *analytic*, and *workflow* service. These services are supported by a Python-based kernel, which interacts with the database, as illustrated in Figure 11.3.

The data processing service manages the DT data, ensuring that data from different smart devices are standardised into a unified format and stored in the database. Additionally, this service can communicate with connected smart devices via MQTT to send data requests and configure data settings as needed.

The analytic service is responsible for conducting structural model simulations within the virtual environment, as well as facilitating learning through probabilistic inference. This involves employing algorithms to generate simulated structural responses $\hat{s} = m(\theta, u, e)$, and conducting Bayesian inference to determine model parameters θ (see Chapter 7 for reference). The outputs of the analytic service are called upon request by the workflow service.

The workflow service, serving as the central coordinator of the DT, orchestrates various tasks autonomously through a Petri net, as elaborated in Chapter 8. It handles the sequence of DT events, such as Bayesian updating of the digital representation based on the physical entity, ensuring these actions are automated and adaptive based on the current state of the physical asset. This service also manages the relationships between other services, controlling the flow of information and data exchange between them.

Data processing service

The data processing service of the DT manages the data received from the sensors and actuators. In the pilot scenario, the distance sensor operates by taking discrete measurements at regular intervals and when a distance threshold has been surpassed. These measurements are captured by the data processing service, which acts as an intermediary between the sensor and the analytic service.

Upon receiving the discrete measurements, the data processing service processes the data, ensuring its accuracy and reliability. Subsequently, the processed data is forwarded to the analytic service for further analysis and interpretation. This analysis may involve tasks such as performing Bayesian inference, identifying patterns, detecting anomalies, or making predictions based on the received data.

Furthermore, the data processing service is responsible for storing the processed data in a database, ensuring its persistence and accessibility for future reference. The data is typically stored in a structured format, such as JSON, which allows for efficient storage and retrieval of information.

Overall, the data processing service acts as a vital component within the DT ecosystem, facilitating the flow of data from sensors to analytical modules while ensuring data integrity and reliability throughout the process.

Analytic service

The analytic service of the DT refers to a module responsible for conducting analytical tasks related to the structural behaviour and performance of the asset. This service typically includes functionalities such as structural modelling, simulations, health monitoring, Bayesian inference, predictive maintenance and decision making.

In this case study, the analytic service conducts Bayesian inference to determine the unknown force associated with the observed displacement. This involves solving an inverse problem, where each time a displacement surpasses a predefined threshold, the PDF of the force causing this displacement is estimated through Bayesian inference.

The model representation, namely $m(\theta, u, e)$, needs periodic updating based on the actual structural states s observed through the SHM system. The updating of a structural model using information gathered from sensors can be understood as an inverse problem, where there exist parameters to be inferred, as seen in Section 7.2.

In the referred Bayesian inference module, the Metropolis-Hastings algorithm is adopted as a stochastic simulation method to solve Equation 7.4, given its versatility and implementation simplicity [178]. The Metropolis-Hastings algorithm is typically used in the context of the inverse problem as the one presented in this case study. In this context, the goal is to infer the parameter of a model, currently the value of the unknown force, given observed data. The implementation took a total of $N_s = 5 \cdot 10^4$ simulations and a Gaussian PDF as proposal distribution with standard deviation given so that the resulting acceptance rate lies between the recommended interval $[0.2, 0.4]$ [179]. F_{3x} is the force applied to the structure, which is a priori unknown, thus it is represented using the model parameter $\boldsymbol{\theta}$. A uniform prior PDF was chosen to represent $\boldsymbol{\theta}$ within the interval [0.1, 5] N, denoted as $\mathcal{U}[1, 5]$ N, reflecting the initial belief about the potential values of the force.

Workflow service

The workflow service for autonomous structural-integrity decision making, as illustrated in Figure 11.5, is modelled through a high level Petri net (HLPN). This model comprises eight places ($p1$ to $p8$), seven transitions ($t1$ to $t7$), and two cold transitions (ϵ) designated for data arrival and system rearm. Each place corresponds to a discrete-event state, such as 'data arrival', 'system updated', 'waiting mismatch evaluation', etc. In Figure 11.5, coloured text labels are used to aid in the visual interpretation of the system states within the HLPN graph. The dark small rectangles indicate symbolic transitions, whereas the grey text labels provide explanatory information about key places.

Changes in the state of the DT system are initiated by a series of automated actions triggered by the firing transitions $t1$ to $t7$. Table 11.3 provides an overview and description of the actions associated with each transition. Notably, transitions $t1$, $t5$, and $t6$ are governed by transition conditions $C1$, $C5$, and $C6$, respectively. The algebraic predicates defining these conditions are presented in the third column of Table 11.3. Activation of the transitions occurs when their associated variables satisfy the respective conditions C_i, where $i = 1, 5, 6$. In the third row, the symbol τ_3 represents the time in seconds required for the DT analytic service to process the Bayesian inference. The term $\jmath$ refers to the mismatch value, and is given in Equation 11.2.

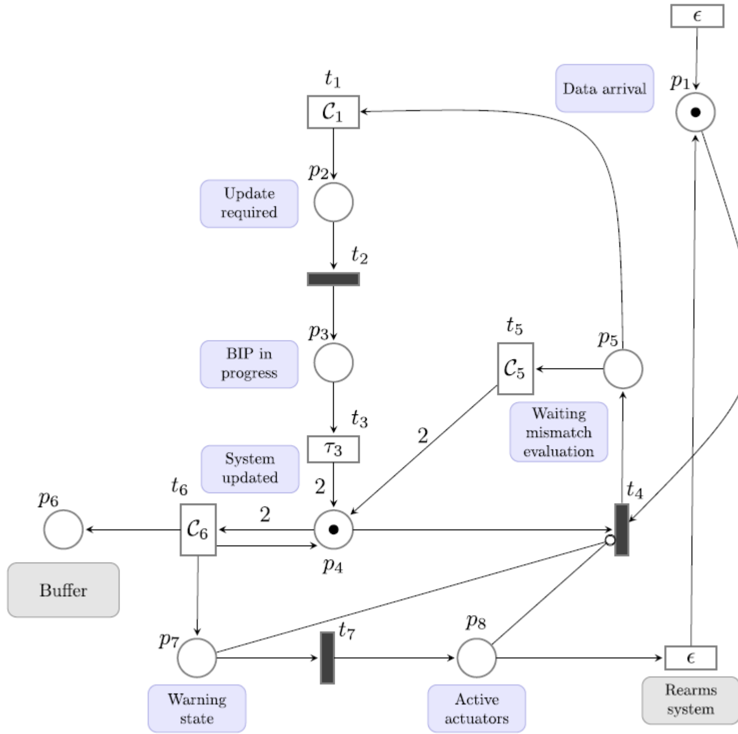


Figure 11.5: HLPN employed as workflow model of the 2D metal tower proof of concept.

Table 11.3: Transitions of the HLPN workflow model.

Transition	Type	Rule	Description
t_1	Conditional	$\mathcal{C}_1 : \{\jmath \geqslant 0.1\}$	Evaluates mismatch
t_2	Symbolic	–	Call the BIP module
t_3	Timed	Enabled after τ_3	Execute the BIP
t_4	Symbolic	–	Initiates VE–PE mismatch
t_5	Conditional	$\mathcal{C}_5 : \{\jmath < 0.1\}$	Evaluates mismatch
t_6	Conditional	$\mathcal{C}_6 : \{\text{mean}(F_{3x}) \geqslant F_\xi\}$	Checks value of inferred force
t_7	Symbolic	–	Activates actuator

The dynamics of the HLPN can be outlined as follows. The system initiates at time $k = 0$ upon the arrival of new data from sensor S_1. Initially, the Virtual Entity (VE) representing the DT is updated based on the Physical Entity (PE) constituted by the monitored metal tower. This initial state is denoted by one token each in places p_1 and p_4, represented as $\mathrm{M}_0 = (1, 0, 0, 1, 0, 0, 0, 0)^T$. Subsequently, transition $t4$ is fired, resulting in one token placed in p_5 while removing it from p_4. This signifies the DT's awareness of a new, unidentified force acting on the structure, prompting a decision regarding force identification and VE update.

This decision-making process involves transitions t_1 and t_5, governed by conditions $\mathcal{C}1$ and $\mathcal{C}5$ (refer to Table 11.3), respectively. These transitions are activated based on a mismatch evaluation, determined as follows:

$$\jmath = \frac{\|s_{5x} - \tilde{s}_{5x}\|}{\tilde{s}_{5x}} \tag{11.2}$$

Here, s_{5x} denotes the current horizontal displacement measured at node 5, while $\tilde{s}5x$ represents its previously recorded value after Bayesian inference, initialised as $\tilde{s}5x = 0$ for $k = 0$. Consequently, if the latest measured displacement deviates by more than 10% from the previously recorded value (stored in the database, as illustrated in Figure 11.5), transition t_1 is triggered, prompting the DT to execute a structural Bayesian update. This process is delineated by the workflow sequence p_2, t_2, p_3, t_3, ultimately resulting in the placement of one token in p_4, returning the system to the 'updated' state.

Alternatively, if the condition is not met, transition t_5 directly places one token in p_4, indicating that the DT does not require a Bayesian update of the VE concerning the PE. Consequently, the DT maintains its previous 'updated' state.

In the HLPN graph depicted in Figure 11.5, it is noted that while the DT remains in the 'updated state'—indicating the marking of place p_4— an assessment is conducted regarding the inferred force values to ascertain if their mean value surpasses the threshold F_ξ. Upon detection of such an occurrence, transition t_6 is activated. This transition initiates a series of warning states and actions, characterized by nodes p_7, t_7, p_8, which autonomously signal when the structure faces a force potentially compromising its integrity. Subsequently, a visual alarm is triggered by the LED and screen actuators via transition t_7, and the system shifts to the 'warning state'.

In this state, the system is reset, awaiting new data arrival, symbolized by the cold transition (ϵ), until the warning state is revisited for reevaluation. Furthermore, when the warning sequence is activated, a token is accumulated in p_6, serving as an information buffer. This token count can be leveraged for diagnostic purposes, providing insight into the frequency with which the structure has been subjected to force values surpassing the integrity threshold F_ξ.

11.3 Bayesian inference of unknown parameters

After activating the smart devices and establishing connections with their respective e-boards, a series of 15 load cases are systematically applied to node 3 of the test structure to explore various structural integrity scenarios. In each scenario, the overall DT behaviour is commanded through the HLPN outlined in Figure 11.5, whose dynamics are evaluated using the state evolution equation along with the execution rules detailed in Section 11.3. This evaluation yields the sequence of system states represented by the marking M_k, where $k > 0$. The results illustrating the behaviour of the DT, including both applied and inferred loads, are presented in Figure 11.6. The applied forces to the test structure, depicted by the grey box-dotted line, are measured externally using a thin-film force sensor and are not processed within the DT environment. These measured values serve solely for visual validation of the inferred values generated by the DT. Additionally, the figure highlights instances of significant state changes within the HLPN.

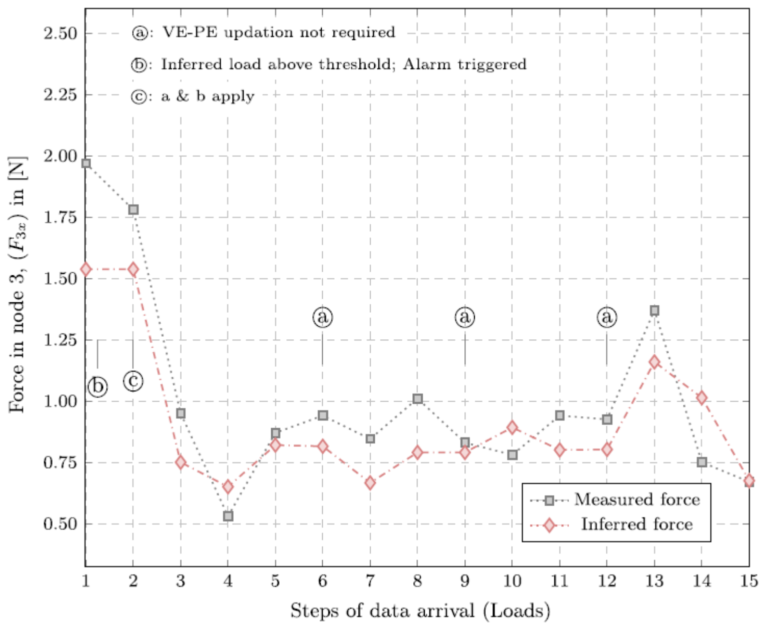


Figure 11.6: Sequence of applied (in grey) and inferred (in red) loads with indication of main DT actions in the case study of the 2D metal tower

In each load case requiring an update to the DT, the Bayesian module is invoked to generate a posterior PDF of the inferred values of the unknown force θ. This allows for the reproduction of the remaining model responses, including the displacements and rotations of nodes. As an illustration of the visual capabilities of the DT frontend, Figure 11.7 displays a plot comparing the posterior and prior PDFs of the model parameter θ for load case number 13 of Table 11.4, alongside a graphical representation of the VE. Additionally, the front end provides visual cues for certain key state variables from the PE under the specified load case.

Additional insights into the DT response to the test loads can be found in Table 11.4. Furthermore, summarised results regarding the behaviour of the HLPN model are presented for test cases 1, 2, and 10 in Tables 11.5, 11.6, and 11.7 respectively. Within these tables, the fourth column outlines the sequence of primary events, such as the activation of the warning state and/or the triggering of conditional transitions. The rightmost column offers descriptions of the overall DT behaviour in connection with the HLPN states.

In load case 1 (see Table 11.5), it is noted that transition t_1 is triggered at $k = 1$ due to its activation conditions, with a token present in its preset place p_5 and the condition $\mathcal{C}_1$ being satisfied. This action leads to the generation of a token in p_2, initiating the Bayesian updating of the DT states. Subsequently, the system returns to the updated state with two tokens in p_4. Upon updating, if the resulting inferred force exceeds the threshold force $F_\xi = 1.5$ [N], transition t_6 is activated, initiating the warning sequence p_7, t_7, p_8 along with their associated actuators. This sequence can be observed in Table 11.5 starting from time step $k = 4$.

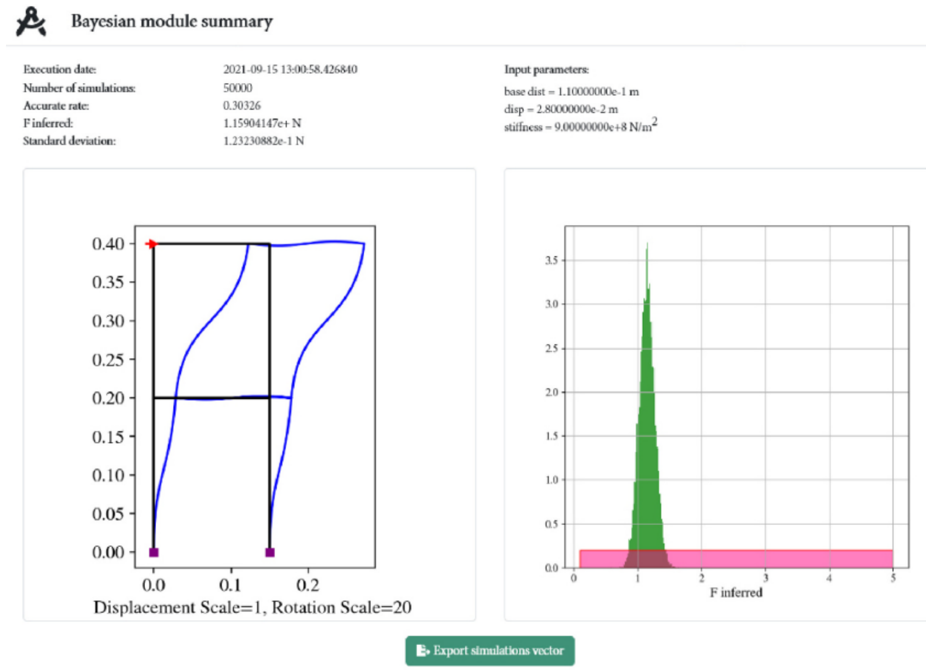


Figure 11.7: Plot of the Bayesian module for load case 13 (refer to Table 11.4) in the frontend of the platform, with indication of the posterior PDF of the inferred force (right side) along with a graphical representation of the updated VE (left side). In the right panel, the pink rectangle represents the uniform prior PDF of the applied force, whereas the posterior PDF is given in green colour.

Table 11.4: Measured and inferred DT variables for the sequence of 15 load cases applied to the test structure. In the third and fourth rows, the symbol θ represents the unknown parameter, which coincides with the force F_{3x}

Load case	Measured force [N]	θ_{mean} [N]	θ_{std} [N]	s_{5x} [mm]	$\jmath \geqslant 10\%$	Activates warning
1	1.92	1.60	0.13	39	Yes	Yes
2	1.77	1.60	0.13	36	No	Yes
3	0.95	0.75	0.12	20	Yes	No
4	0.53	0.65	0.11	15	Yes	No
5	0.87	0.82	0.12	21	Yes	No
6	0.94	0.82	0.12	19	No	No
7	0.85	0.67	0.12	16	Yes	No
8	1.01	0.77	0.12	18	Yes	No
9	0.83	0.77	0.12	19	No	No
10	0.78	0.88	0.12	23	Yes	No
11	0.94	0.80	0.12	20	Yes	No
12	0.92	0.80	0.12	22	No	No
13	1.37	1.16	0.12	28	Yes	No
14	0.75	1.01	0.12	24	Yes	No
15	0.67	0.66	0.13	16	Yes	No

Alternatively, in load case 2, the ultrasound sensor registers a displacement value of 36 mm, which deviates by less than 10% from the previous measurement (i.e., 39 mm). Consequently, the HLPN determines that updating the VE relative to the PE is unnecessary, maintaining the system in the updated state by initiating t_5 to return two tokens to p_4. However, despite this, the HLPN identifies that the force applied to the structure still exceeds the threshold force $F_\xi = 1.5$ N. Therefore, transition t_6 is triggered once more, leading to the activation of warning events and the system transitioning from normal operation to a warning state. In this mode, the actuator A_1 initiates visual alarms, as described earlier. These actions are documented in Table 11.6 starting from time step $k = 2$.

Finally, load case 10 signifies the scenario where the DT recognizes the necessity to update the VE concerning the PE. After the update, the inferred force does not surpass the threshold value, prompting the workflow model to halt and await new data.

Table 11.5: Summary of events and actions carried out by the PN workflow model under load case 1

PN state	Marking ($\mathbf{M}_k$)	Firing vector ($\mathbf{u}_k$)	Main events	Description
$k=0$	$(1\,0\,0\,1\,0\,0\,0\,0)^T$	$(0\,0\,0\,1\,0\,0\,0)$	New data arrived	PN starts
$k=1$	$(0\,0\,0\,0\,1\,0\,0\,0)^T$	$(1\,0\,0\,0\,0\,0\,0)$	$\mathcal{C}_1$ {→True}	Checks to update VE & PE {→True}
$k=2$	$(0\,1\,0\,0\,0\,0\,0\,0)^T$	$(0\,1\,0\,0\,0\,0\,0)$	Update required	Call the BIP module
$k=3$	$(0\,0\,1\,0\,0\,0\,0\,0)^T$	$(0\,0\,1\,0\,0\,0\,0)$	BIP under execution	VE updating according to PE
$k=4$	$(0\,0\,0\,2\,0\,0\,0\,0)^T$	$(0\,0\,0\,0\,0\,1\,0)$	$\mathcal{C}_6$ {→True}	System updated; $F_{3x} \geqslant F_\xi \rightarrow$ {True}
$k=5$	$(0\,0\,0\,1\,0\,1\,1\,0)^T$	$(0\,0\,0\,0\,0\,0\,1)$	Action required	Activates actuator A_1
$k=6$	$(0\,0\,0\,1\,0\,1\,0\,1)^T$	$(0\,0\,0\,0\,0\,0\,0)$	Warning state	System rearm; awaiting new data

Table 11.6: Summary of events and actions carried out by the PN workflow model under load case 2

PN state	Marking ($\mathbf{M}_k$)	Firing vector ($\mathbf{u}_k$)	Main events	Description
$k=0$	$(1\,0\,0\,1\,0\,1\,0\,0)^T$	$(0\,0\,0\,1\,0\,0\,0)$	New data arrived	PN starts
$k=1$	$(0\,0\,0\,0\,1\,1\,0\,0)^T$	$(0\,0\,0\,0\,1\,0\,0)$	$\mathcal{C}_5$ {→True}	Checks to update VE & PE {→False}
$k=2$	$(0\,0\,0\,2\,0\,1\,0\,0)^T$	$(0\,0\,0\,0\,0\,1\,0)$	$\mathcal{C}_6$ {→True}	System previously updated; $F_{3x} \geqslant F_\xi \rightarrow$ {True}
$k=3$	$(0\,0\,0\,1\,0\,2\,1\,0)^T$	$(0\,0\,0\,0\,0\,0\,1)$	Action required	Activates actuator A_1
$k=4$	$(0\,0\,0\,1\,0\,2\,0\,1)^T$	$(0\,0\,0\,0\,0\,0\,0)$	Warning state	System rearm; awaiting new data

Table 11.7: Summary of events and actions carried out by the PN workflow model under load case 10

PN state	Marking ($\mathbf{M}_k$)	Firing vector ($\mathbf{u}_k$)	Main PN events	Description
$k=0$	$(1\,0\,0\,1\,0\,2\,0\,0)^T$	$(0\,0\,0\,1\,0\,0\,0)$	Data arrival	PN starts
$k=1$	$(0\,0\,0\,0\,1\,2\,0\,0)^T$	$(1\,0\,0\,0\,0\,0\,0)$	$\mathcal{C}_1$ {→True}	Checks to update VE & PE {→True}
$k=2$	$(0\,1\,0\,0\,0\,2\,0\,0)^T$	$(0\,1\,0\,0\,0\,0\,0)$	Update required	Call the BIP module
$k=3$	$(0\,0\,1\,0\,0\,2\,0\,0)^T$	$(0\,0\,1\,0\,0\,0\,0)$	BIP under execution	VE updating according to PE
$k=4$	$(0\,0\,0\,2\,0\,2\,0\,0)^T$	$(0\,0\,0\,0\,0\,0\,0)$	–	System updated; waiting new data

Overall, these findings confirm the efficacy of the proposed DT framework in autonomously adjusting to incoming data, facilitating the updating of the VE in response to the PE's performance as measured by the sensors. Additionally, the results indicate that the PE can be influenced by feedback from the VE via actuators. Furthermore, the demonstrated proof of concept illustrates that these interactions can be efficiently managed through an HLPN serving as the workflow model of the DT, offering an event-driven approach to synchronise the VE and PE at the system level.

11.4 Conclusion

In conclusion, this proof of concept has been conducted on a laboratory 2D test structure to both conceptualise the DT framework and address some of the challenges encountered in real-world applications. The following conclusions can be drawn:

- The proposed DT framework has demonstrated the feasibility of bidirectional interactions between the physical and virtual realms within the context of structural integrity.
- A probabilistic Bayesian approach has been suggested for model updating due to its maturity in structural health monitoring, structural control, and structural integrity. However, alternative learning methods such as ML techniques could be integrated either as supplements or replacements to the Bayesian approach without sacrificing generality.
- The framework was validated through a proof of concept using a laboratory test structure to simulate integrity scenarios (i.e., node displacements) without the need for excessively high forces, which might have required a cumbersome experimental setup. However, it is worth noting that employing a small-scale test may overlook certain structural aspects only observable in larger-scale models, thus posing a limitation to this study. Nonetheless, the proof of concept enabled the conceptualisation, formulation, and technological integration of the proposed DT, serving as a preliminary step toward application in larger, potentially real-world structures.
- The results indicate that the proposed DT can aid in decision making for failure prevention, as the virtual representation can be utilised for reliability and risk assessment during damage, and automated alarms can be triggered in case of failure scenarios.
- Further research is warranted to explore methods for optimal computational allocation, such as edge or fog computing, within the software/hardware integration of the DT to enable efficient application in cases necessitating complex structural models, pervasive monitoring, and citizen-centred sensors. Additionally, demonstrating the proposed DT in a full-scale structural application would be desirable.

Chapter 12

Digital Twin of a 3D Metal tower

This second case study explores the implementation of a DT for a 3D tower structure of larger dimensions. Here, the focus shifts towards the application of a generative setting and a pipeline of models for the damage assessment capability within the DT framework.

By leveraging generative models and sophisticated algorithms, it is showcased how the DT workflow can be effectively utilised to simulate, analyse, and predict structural behaviour, performing a 4-level damage assessment strategy that allows for rapid damage mitigation and proactive maintenance. The generative setting mirrors the physics system and generates a high-quality dataset that allows the training of the surrogate predictive models to instantly and accurately identify the damage situation, along with the damage location, quantification and RUL prediction, even in rare events or black-swan scenarios.

12.1 Description

This case study implementation consists of a six-storey 1.5m-high laboratory scale steel frame structure and its corresponding DT as shown in Figure 12.1, receiving discrete displacement and force measurements through sensors. The structure is exposed to variable lateral forces and experiences damage as a result of the gradual loosening of its bolts. Wireless IoT sensors employing ultrasonic methods were utilised to record the displacements of individual storeys, while an IoT digital transducer was employed to measure the force. The virtual twin is computationally simulated using a FE model developed in OpenSeesPy [180], a Python interface of the open-source software framework for analysis of structures.

Aligned to the flexible and open-source principles of this research, data was gathered using simple and affordable sensors. These sensors exhibit limited sensitivity and moderate precision, transmitting the data via the IoT. Unlike professional wired data acquisition systems, this approach, while economical, introduces additional noise in the measurements. Consequently, the DT must tackle this additional noise, devising strategies to effectively reduce its impact.

The study focuses on the static behaviour of the structure, with a primary interest in understanding its equilibrium state and displacements when subjected to loads. Given the structure's significant stiffness—stemming from the geometry and the material's density, damping and inertia effects are disregarded. In this context, the system is constrained to move within a single plane, characterised by one degree of freedom along the y-axis, as movement along the x-axis is deemed negligible and the z-axis is effectively restricted due to the presence of a rigid foundation. Therefore, this study employs the linear form of the motion equation to represent the static equilibrium condition under the influence of applied loads, with the system's response calculated as follows:

$$F = K(t) \cdot d \tag{12.1}$$

where F is the infringed force against the tower, $K(t)$ is the stiffness evolving over time due to the progression of the damage, 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.

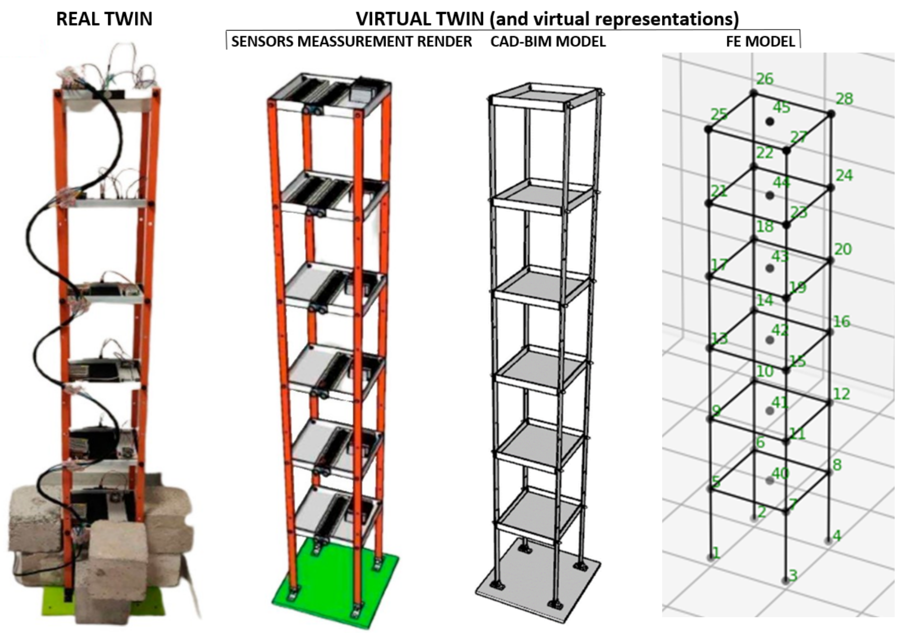


Figure 12.1: Case study of a DT deployment for a 3D metal tower.

According to the geometry and material properties of the tower, the values of the main structural parameters considered are shown in Table 12.1:

Table 12.1: Parameters of the case study structure.

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	$\sigma_i = 0.0758 \cdot t_i + 0.0448$	N

Damage-induced modifications in a structure such as stiffness lead to alterations in its displacement response, which can be effectively assessed through SHM. This type of damage is assumed to arise from the gradual loosening of the joints after forces are applied over time. Once this process starts, the damage is expected to progress for a period of $T = 100$ years, representing the estimated useful lifespan of a civil structure. The linear estimation of bolt loosening suggests a 60% reduction in the structure's stiffness after 100 years of intermittent forces, marking the conclusion of its operational life.

In this model, the stiffness coefficient K becomes dependent on time after the onset of damage, diminishing according to the formula $K(t) = \alpha \cdot t$, where:

$$\alpha = 1 - 0,006 \cdot t \tag{12.2}$$

Here, α represents a stiffness reduction coefficient associated with bolt loosening, which varies from 0 to 1. K denotes the initial stiffness, and t is the time elapsed since the damage appeared, measured in years.

12.2 Physics-model calibration

With the purpose of calibrating the FE model, various damage conditions were artificially introduced to the structure. Due to limitations in the experimental setup, it was possible to create only three specific scenarios, as illustrated in Figure 12.2. These included: an undamaged (pristine) structure, a partially damaged structure due to moderate bolt loosening, and a completely damaged structure.

In contrast, the FE model was designed to emulate a broader range of potential damage scenarios. This was achieved by methodically reducing the stiffness of connections at different levels, focusing on one story at a time. These simulations accounted for a spectrum of random forces, with magnitudes up to 450N, which is considered a critical threshold for causing structural breakdown. For each simulated scenario, the resulting displacements at various levels of the structure were carefully recorded.

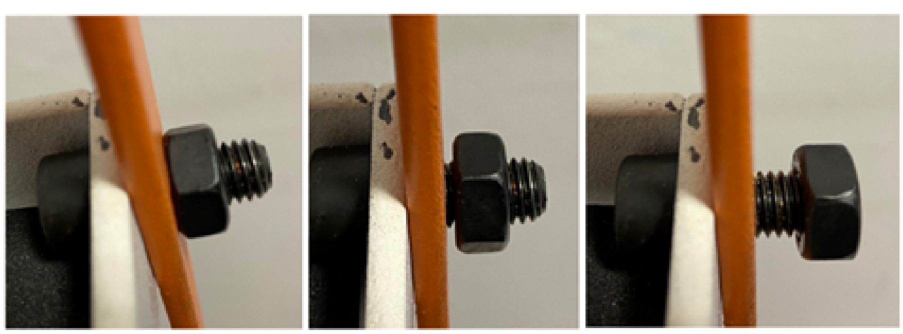


Figure 12.2: Damage caused by bolts' loosening: healthy state (left), medium damage (centre) and fully damaged (right).

Measurements collected from sensors were used to compile an initial dataset, denoted as *data_sens* (Figure 12.3), which includes data relevant to three previously described scenarios: (1) a healthy tower, (2) a partially damaged tower with medium loosening of the bolts, and (3) a fully damaged tower from complete bolt loosening. This dataset was utilised to calibrate the FE model, achieving a high accuracy against these experimental observations.

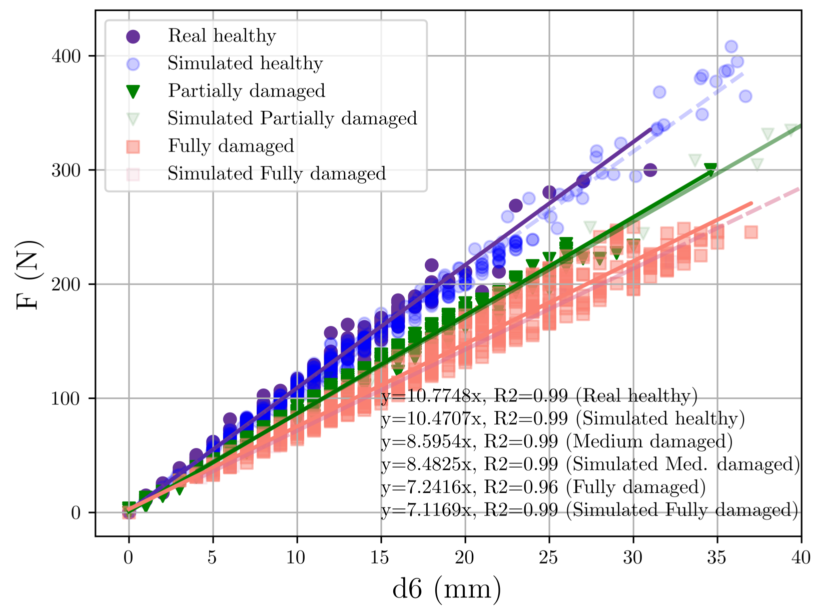


Figure 12.3: Calibration of the FE model in 3 different scenarios (healthy, medium damage, and severe damage) with data coming from the sensors (*data_sens*).

Through the application of statistical approximation methods, an augmented dataset, *data_stat*, was generated based on the outcomes of FE simulation analyses, incorporating calibrated noise. This noise adheres to a Gaussian statistical distribution with embedded heteroscedasticity (Figure 12.4), which is a common phenomenon in engineering [156] where the noise is input-dependent.

Considering equation (12.1) as a case of linear regression and with $i = 1, \dots N$ representing the number of measurements, the displacements of the structure are determined as follows:

$$d_i \;=\; K_i^{-1}(t_i) \cdot F_i + \varepsilon_{i(t_i)} \tag{12.3}$$

where the dependent variable d_i equals the independent random variable F_i times a coefficient, plus a random disturbance term $\varepsilon_{i(t_i)}$. This term presents zero mean and variance depending on both, the value of F_i 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.

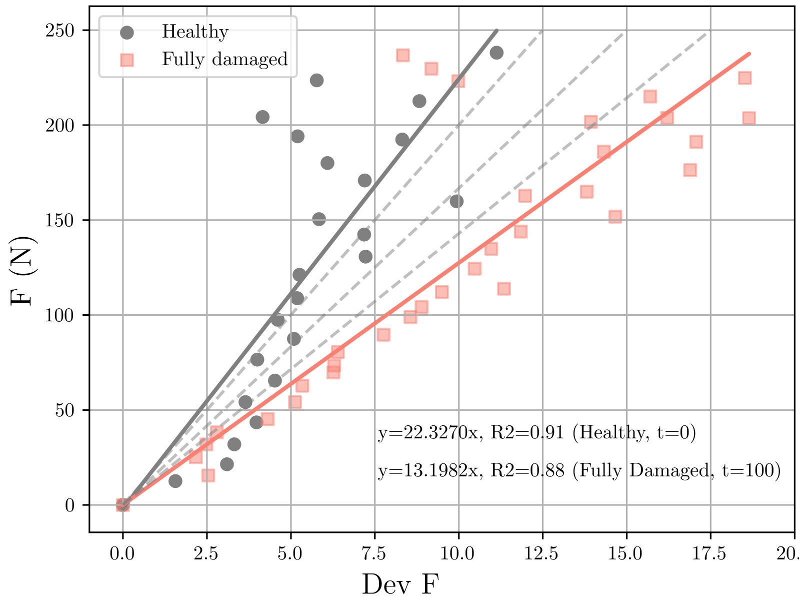


Figure 12.4: Heteroscedasticity of the variance depending on the force F_i and the time t_i from the healthy state (t_i=0) to the fully damaged (t_i=100).

According to the procedure described, a dataset with a tensor shape (the third dimension being equal to the time of damage) is 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 12.5.

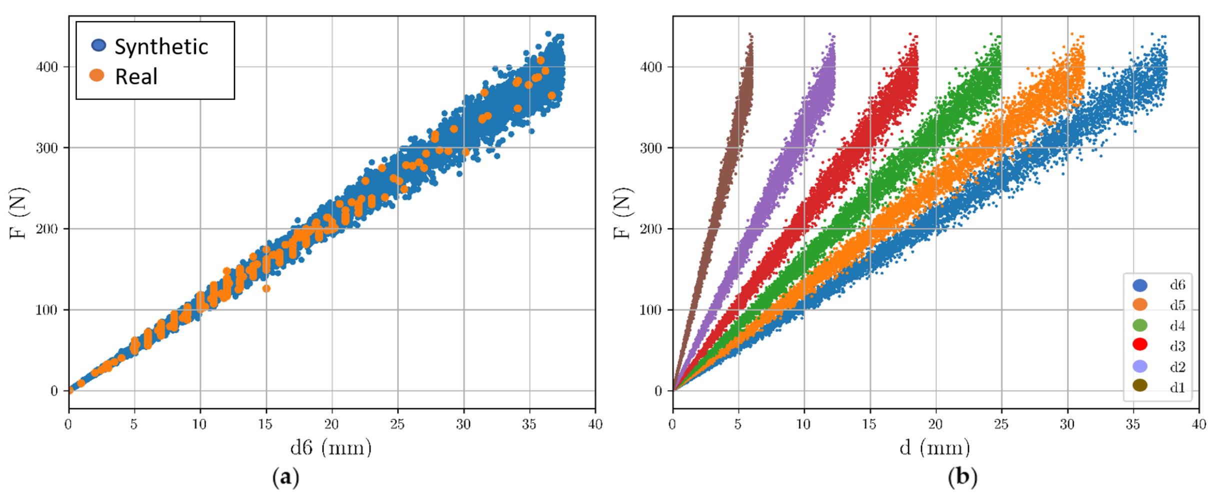


Figure 12.5: Data produced by statistical approximation (*data_stat*) for the healthy state (time t_i=0). (a) Displacements on the sixth floor, synthetic and real. (b) Displacements on each floor, first to sixth.

Emphasising the critical significance of creating a dataset that accurately mirrors real-world conditions is essential, capturing not just the errors but also their variability (heteroscedasticity). This focus on the quality of the dataset stems from the ambitious goal of training data-driven models for real-time diagnostics capable of determining the state of health or damage from sensor data directly, bypassing the preprocessing step, as required for edge computing. This approach necessitates the model's ability to assimilate and interpret error, achieving a balance between bias and variance in its predictions.

12.3 Generative setting

Using classical statistical techniques to simulate a system by numerical methods, which includes incorporating noise based on predefined statistical distributions, generates datasets bound to these set distributions. Models trained on such data are at risk of overfitting and often struggle to accurately represent uncommon or rare scenarios.

Conversely, the present approach is based on the use of conditional deep generative models (in short, conditional generative models built on top of multiple layers of interconnected artificial neurons) which, as opposed to classical simulation-based approaches, do not explicitly counterfeit the physics of the entire system, but only the responses of interest. In this approach there is no need for explicit prior physics knowledge, as the physics information is indirectly supplied through the training data derived from the statistically augmented dataset, resulting in a hybrid approach that combines both physics-based and data-driven elements. This method has the capability to produce data that closely mimics the real-world one, without being constrained by any specific predefined statistical distribution, in a non-parametric manner. Its flexibility allows the generation a wide variety of data samples that enhance the robustness and effectiveness of the models trained on this enriched dataset. Furthermore, the conditioning confers the generative model the capability of exerting precise control over generated outputs and the ability to generate structured results. It is worth noting that generative models rely on random inputs, often referred to as latent space or noise vector, to generate novel and diverse samples. By specifying certain conditions, labels, or values, the model is guided to produce outputs that align with specific criteria, ensuring a class-balance generation or a desired outcome, tailored to the specific problem being addressed. This way the generation produces fine-grained results, ensuring domain adaptation and task-specific generation. In addition, conditioning enhances training stability and convergence, contributing to the overall versatility and effectiveness of generative models.

The proposed generative framework is particularly well suited for handling discrete measurements acquired in the realm of SHM methodologies. Within the SHM framework, measurements are strategically obtained at certain locations of the structure where the sensors are specifically placed, such as high-stress/strain areas, as well as those subject to heightened environmental influences. In these locations, the structural behaviour exhibits a high level of sensitivity towards damage or changes, representing the critical and most vulnerable points of the structure. The strategic focus on these specific points enables the acquisition of comprehensive information while optimising resources.

The generative model fine-tunes its parameters during training to minimise the differences between the data generated and the target data, accounting for the inherent noise present in the system. Once trained, the generative model releases synthetic data aligned with sensor measurements and the physics of the system. It offers a versatile tool for data generation and facilitates a precise understanding of the real system, not only in common scenarios but also incorporating black swan events and 'what if' situations.

Once trained, the output of the generative model is a structured data set (*data_GAN*) composed of the force, damage time, and displacements of the six floors on the horizontal axis, all accurately labelled with healthy (0) or damaged (1) tags. In the presented case study, there are six combinations of damage (one per floor) caused by bolt loosening, with a time span ranging from 0 to 100 years since the damage began.

12.3.1 Architecture

The architecture of the generative setting consists of a conditional Wasserstein generative adversarial network with gradient penalty (CWGAN-GP) based on convolutional neural networks (CNNs). While the first ensures the creation of an original, diverse, and class-balanced dataset that closely resembles the real data distribution [181], the second seamlessly detects and reproduces the spatial and temporal patterns in the data with great accuracy [182]. This model is able to reproduce the dynamics and system's intricacies after being trained on the augmented dataset (*data_stat*) comprising both real sensor measurements and the FE simulations with heteroscedastic noise incorporated.

The order of the CNN is set according to data dimensionality. In the present case, as data primarily exhibit variations along a single dimension such as a single spatial axis, the order is fixed to a 1D CNN problem.

The CNNs have a recognised ability to process data with a grid-like shape, such as the displacement vector over the six stories of the tower in the case study. The patterns shown by this signal (Figure 12.6) will allow the models to learn that when the tower is healthy, the signal presents a common linear tilt through the 6 stories. Conversely, 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 manner, the resultant CWGAN-GP possess the capability to generate realistic data in all the case scenarios for the healthy state and the damaged state, considering the full range of forces that the tower can resist without compromising its integrity (maximum allowed displacement for a force of 450N). The damage was labelled with a '0' if healthy and a '1' if damaged, and this label information will condition the generation of new data.

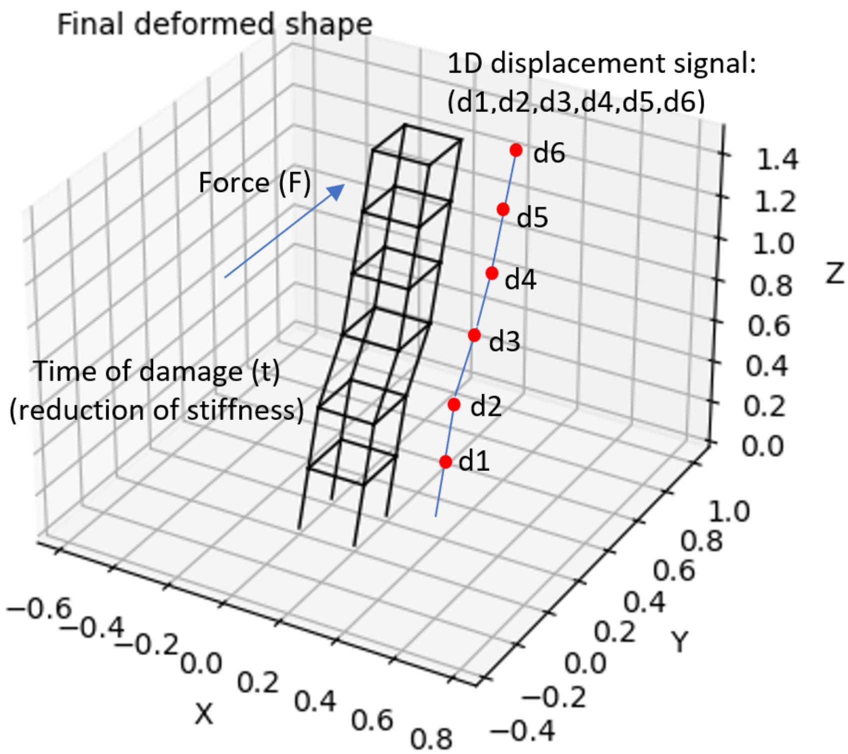


Figure 12.6: Plot of a 1-D displacement signal corresponding to a force F and a time t of damage affecting one of the tower's floors.

Within the CWGAN-GP, the configuration of both generator and discriminator or critic is detailed in Table 12.2 and Figure 12.7. The generator employs CNNs, batch normalisation, and ReLU activation functions followed by fully connected dense layers; before the output layer, a non-linear hyperbolic tangent activation function was introduced. The critic also accounts for CNNs, layer normalisation, and LeakyReLU, which better performs with gradient penalty. Besides, dropout is also applied to avoid overfitting.

The primary hyperparameters are set according to the recommended values for using the WGAN-GP [126] and are included in Table 12.3. 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.

When a conditional WGAN-GP is used to generate class-balanced datasets, the generator can be conditioned on specific class labels to control the characteristics of the generated samples. Taking advantage of this flexibility, the condition will be set to binary to generate a class-balanced dataset for binary classification tasks such as damage detection at Level 1. In this case, there is one *label*: healthy '0' or damaged '1', corresponding to the global state of the tower, and the $condition = label$. When the task is a multiclass classification as occurs at Level 2: damage location, the condition will have multiple labels, up to 6 (*label1*, *label2* ...*label6*), each of them corresponding to '0' if healthy or '1' if damaged, on each of

Table 12.2: Architecture of the generative CWGAN-GP model.

Generator	**Discriminator (Critic)**
Input (latent space, condition)	Input (d1,d2,d3,d4,d5,d6,F,t,condition)
CNN 1D (ReLU, 32 , 3)	CNN 1D (LeakyReLU, 32 , 3)
BatchNormalization / Maxpooling	BatchNormalization / Dropout
CNN 1D (ReLU, 32 , 3)	CNN 1D (LeakyReLU, 32 , 3)
BatchNormalization / Maxpooling	CNN 1D (LeakyReLU, 32 , 2)
CNN 1D (ReLU, 32 , 2)	
BatchNormalization / Maxpooling	
Flatten()	Flatten()
Dense (ReLU, 64)	Dense (1, 1)
Dense (ReLU, 32)	
Dense (Tanh, 1)	
Output (d1,d2,d3,d4,d5,d6,F,t, condition)	Output (Critic's value)

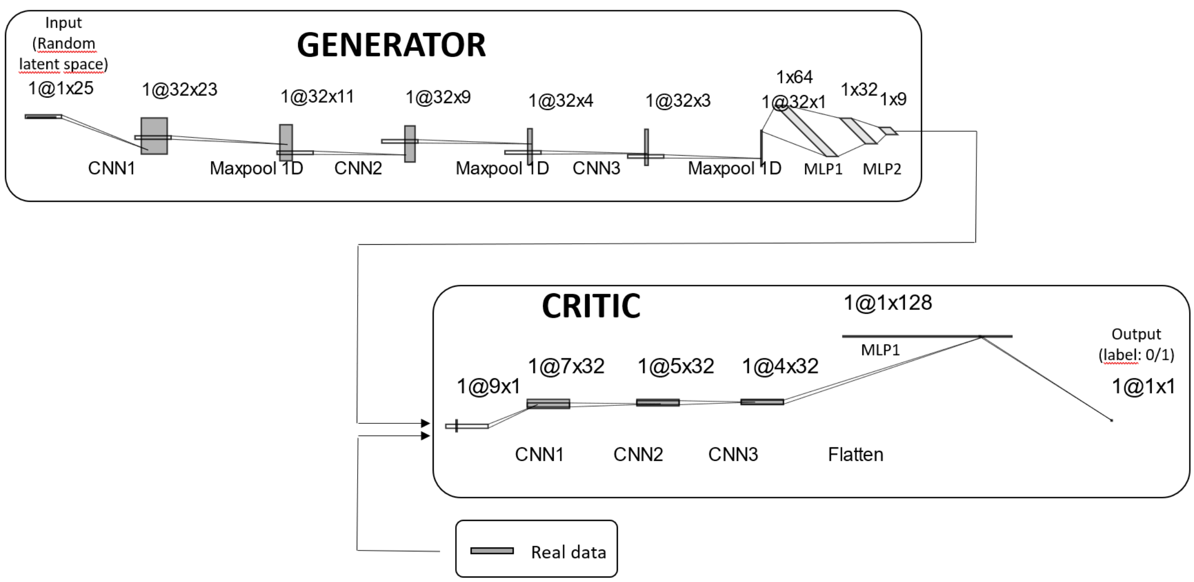


Figure 12.7: Configuration of the CNN-based CWGAN-GP architecture.

Table 12.3: Hyperparameters of the CWGAN-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

the six floors of the tower. In this case, $condition = (label1, label2, ..., label6)$. By controlling the balance of classes during data generation, the resulting datasets are ensured to be suitable for training classifiers, whether binary or multiclass.

12.3.2 Generation of the foundational dataset

The outcome of the data generation utilising the proposed CNN-based CWGAN-GP approach is illustrated in Figure 12.8, showcasing the real and the generated values corresponding to the force (F), the displacement on the 6th floor (d6) and the time of damage (t). The damage in this study is quantified in terms of years since the damage began in the structure, following a temporal stiffness reduction law due to bold loosening increasing with time, as described by Equation 12.2.

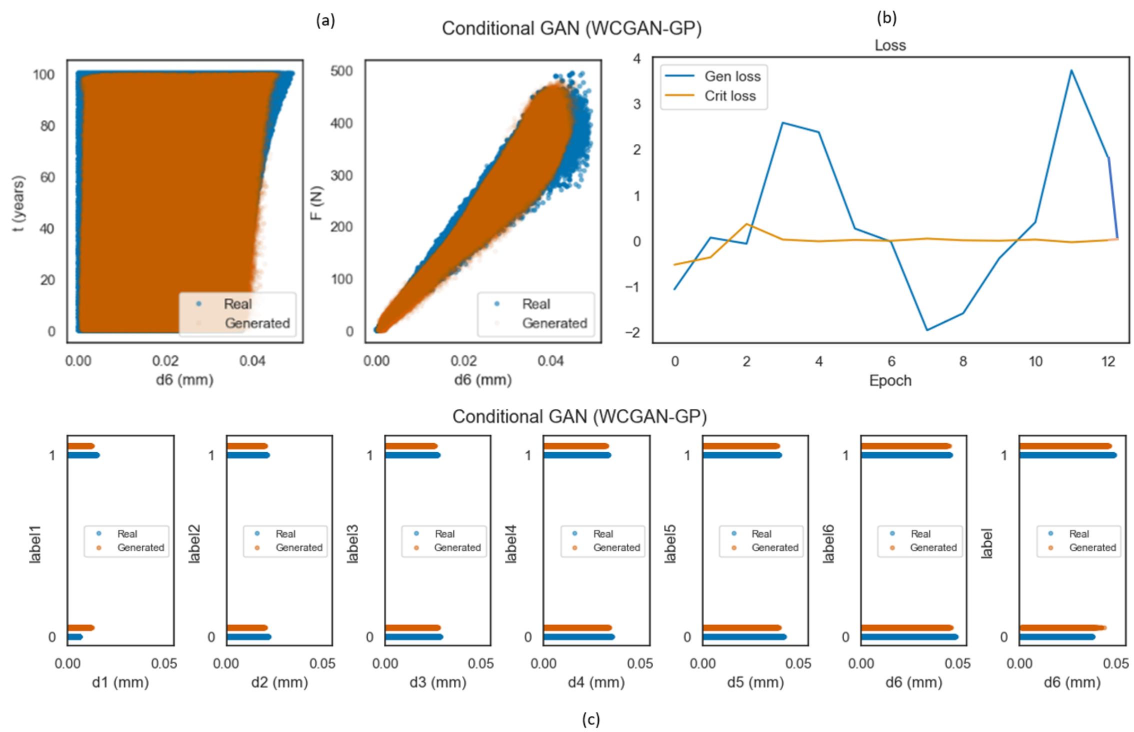


Figure 12.8: (a) Main results from the generative setting (F,t,d6). (b) Loss values in generator and critic versus the number of epochs. (c) Main variables (F,t,d6) and conditions (labels), with a comparison of real versus model-generated values.

This resulting dataset, named *data_GAN* is considered as a *foundational dataset* to highlight its role in providing a solid, reliable basis for models' training across different applications. Training on this dataset ensures that the resulting models are accurate, able to generalise, scalable and efficient, ensuring a balance between fairness and bias mitigation. As a result, these models become empowered to deliver meaningful insights across a range of domains.

Figure 12.8 not only showcases the generated values but also includes a learning curve chart that illustrates the change in loss across the number of epochs for both the generator and the critic. Furthermore, the figure contains six charts that display the generated displacements for each specific label, indicating whether they belong to the healthy '0' or damaged '1' class: label 1 corresponds to displacement on the first floor (d1), label 2 to displacement on the second floor (d2), and so on. Additionally, there is a chart dedicated to highlighting the biggest displacement, particularly on the 6th floor (d6), in relation to the global label that reflects the tower's overall condition of health or damage. The representation includes a comparison between the real values and the values produced by the generative model.

The visual outcome reveals that the results produced by the generative model closely mirror actual data, demonstrating a notable level of authenticity. Moreover, the generated dataset achieves a balanced distribution, applicable not just in the binary scenario (label) but also throughout the multiclass labels (label1, label2, ..., label6). Maintaining such balance across categories is vital for the effective training of the predictive models, ensuring they learn accurately from the data.

The stability of the training process offers valuable insights into both the training dynamics and the quality of the generated samples. The loss curves for both the generator and the critic demonstrate rapid convergence, highlighting the efficiency of the CNNs in learning the data distribution and spatial patterns relevant to the problem with just a few epochs. This optimal number of epochs was determined through a methodical process of experimentation. As depicted in Figure 12.8, the critic initially progresses more quickly, benefiting from direct access to the training dataset and thereby having a greater initial understanding than the generator. Nonetheless, after several epochs, the generator starts to more effectively grasp the gradients and produce data that closely matches the real samples, evidenced by a decrease in its loss. Notably, the vertical scale shows minimal loss oscillation (within ± 4 units at most), suggesting a stable training process. Ultimately, both the critic and the generator reach a point of convergence, indicating the successful training of the CWGAN-GP model.

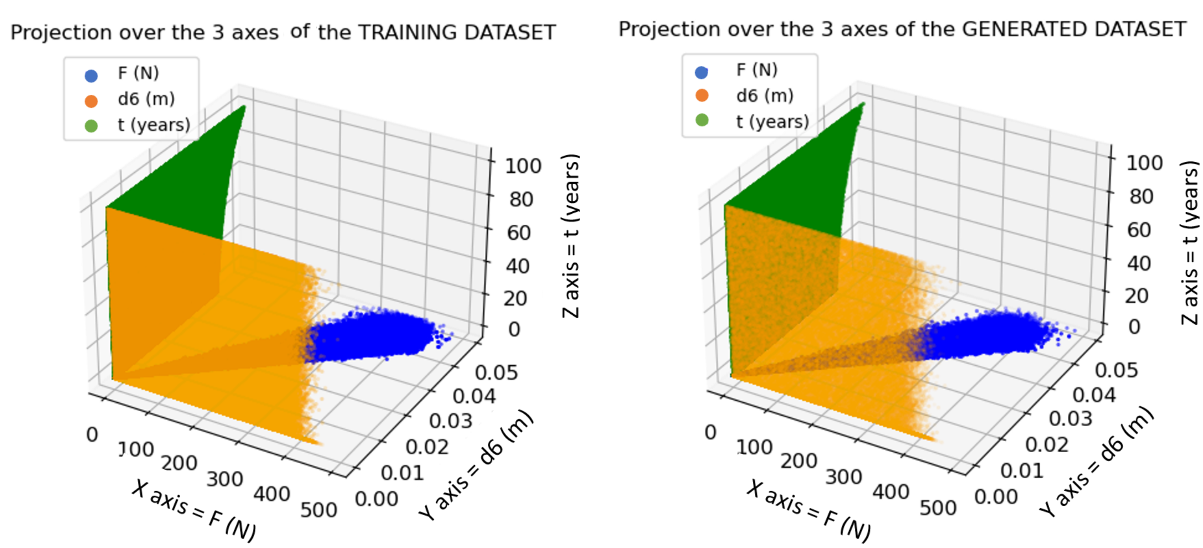
It has been verified that 10 epochs suffice for the training process to reach stability, with effective convergence occurring at this point. This implies that extending the training beyond does not yield significant improvements, eliminating the need for additional epochs. The extensive size of the *data_stat* dataset utilised for training the generative framework, coupled with the appropriateness of the hyperparameters employed, diminishes the necessity for a large number of iterations.

The criteria for stopping the CWGAN-GP's training has been twofold: not only the stability reached by the loss functions but also the metrics adopted, consisting of a qualitative visual inspection of the results together with several quantitative figures: the aforementioned Wasserstein distance [129] between the generated and the real distribution, the Frechet Inception Distance (FID) [158] and the Structural Similarity Index Measure (SSIM) [157].

12.3.3 Quality evaluation of the generated foundational dataset

Evaluating a dataset generated by GANs involves a blend of qualitative and quantitative approaches to assess their quality, realism, and diversity. Qualitative evaluations typically involve visual inspections, where generated samples are directly compared to real ones to judge their visual fidelity. Quantitatively, metrics like the Frechet Inception Distance (FID) measure the diversity of data, and statistical distances like the Wasserstein distance assess how closely the generated data distribution mirrors the actual data. Additional metrics, such as the Structural Similarity Index Measure (SSIM) evaluate data quality and similarity. Together, these methods offer a comprehensive framework for assessing the performance and applicability of GAN-generated datasets for their intended use.

For the qualitative assessment of the generated data through visual comparison, a 3D figure is utilised. Figure 12.9 displays the results of the data generated using the proposed CNN-based CWGAN-GP approach, showing the (force, displacement, damage) triads in a three-dimensional graphic. Visually, the results obtained through the generative method bear a resemblance to the original dataset but they are not identical. The main goal of the generative model was to capture and learn the underlying distribution and patterns present in the data, enabling the generation of novel samples that exhibit shared features with the original data. This process introduced a level of variability, ensuring that the generated data is not a perfect replication of any specific instance in the training set, striking for a balance between similarity to the original data and the introduction of variations, allowing for the production of new and authentic instances.



(a) (b)

Figure 12.9: 3-D visual comparison of the outcomes from the proposed generative setting. 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). (a): training dataset (*data_stat*). Panel (b): dataset generated by the proposed generative setting (*data_GAN*).

Figure 12.10 illustrates another 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 state, medium damage state, and fully damaged state, depicted as panels (a) to (c), respectively. The results demonstrate high accuracy 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.

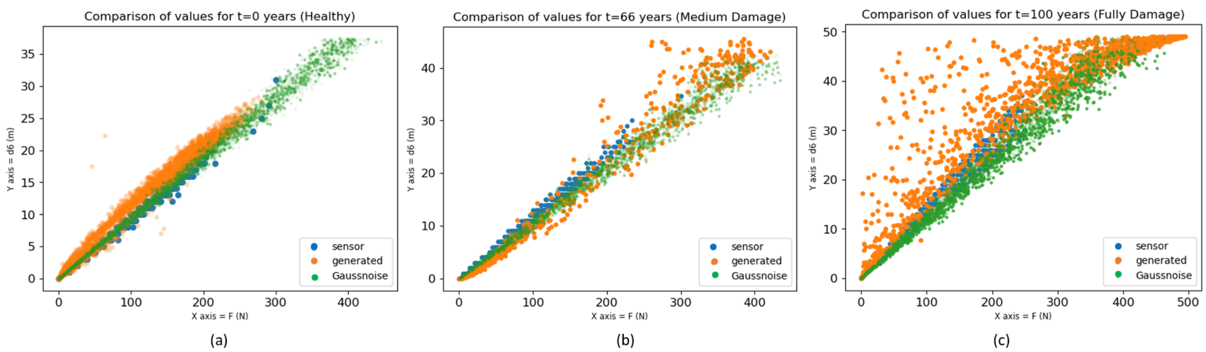


Figure 12.10: Visual comparison of three datasets: *data_sens*, *data_stat*, and *data_GAN*. Panel (a): Dataset corresponding to t = 0 (healthy state); Panel (b): Dataset corresponding to t = 66 years (medium damage state), and panel (c): Dataset corresponding to t = 100 years (fully damaged state).

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 [157].

In this work, three metrics are used, namely the Wasserstein distance [129], the Fréchet Inception Distance (FID) [158], and the Structural Similarity Index Measure (SSIM) [157]. The Wasserstein distance, according to Equation (5.17), 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 (12.4). Again, a lower FID indicates greater similarity between the compared data.

$$FID(x, \widetilde{x}) = \|\mu_x - \mu_{\widetilde{x}}\|_2^2 + Tr(\sigma_x + \sigma_{\widetilde{x}} - 2(\sigma_x \sigma_{\widetilde{x}})^{\frac{1}{2}}) \tag{12.4}$$

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 (12.5) 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.

$$SSIM(x, \widetilde{x}) = \frac{(2\mu_x \mu_{\widetilde{x}} + C_1)(2\Sigma_{x\widetilde{x}} + C_2)}{(\mu_x^2 + \mu_{\widetilde{x}}^2 + C_1)(\Sigma_x^2 + \Sigma_{\widetilde{x}}^2 + C_2)} \tag{12.5}$$

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

Table 12.4 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.

Table 12.4: Results in terms of performance metrics of the proposed CWGAN-GP 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

12.4 Damage assessment pipeline of models

12.4.1 Training methodology

When high-quality class-balanced labelled data from both damaged and undamaged states of the structure are available, supervised DL algorithms can effectively perform all the damage assessment levels, from Level 1 (damage detection) to Level 4 (damage prediction). The extensive quantity of data required for the training can come from sensing devices, physical-based models (like FE), and created by generative methods, among other sources.

The 4-Level damage assessment strategy presented in this work is based on both, unsupervised (for the generative setting) and supervised (for the predictive pipeline) DL approaches, and the algorithmics related are the following:

- Generation: to mimic the system's behaviour and provide the models with a substantial volume of high-quality labelled data class balanced, reducing the risks of overfitting and bias, and achieving improved model performance and scalability.

- Classification: to detect damage in the structure employing discrete class labels (damaged/undamaged) for a simple binary classification, and locate the damage at a specific point of the structure (multiclass classification with damaged/undamaged labels in every predefined location).

- Regression: to predict the extent of the damage based on the reduction of the material's parameter values and the RUL of the structure.

The proposed damage assessment strategy utilises a consistent model architecture, anchored in CNNs, across its entire framework. This strategy is structured as a sequence of four predictive models, with each model tasked with addressing a specific level of damage assessment, comprising two classifiers and two regressors.

The foundational dataset leveraged for training was developed employing a CWGAN-GP within the generative setting detailed in Section 12.3, resulting in a novel class-balance dataset mirroring the probability distribution of the real data. The effectiveness of this generative setting will be substantiated through a comparative analysis of the performance metrics. This comparison will contrast predictive models trained on the dataset generated by the GAN with those trained on a traditionally produced dataset, which includes data derived from FE analysis augmented with heteroscedastic variance-induced Gaussian noise. The findings from this comparison will highlight the significance of the generative setting within this strategy. They will reveal that predictive models trained on data generated in this innovative manner outperform those trained using conventional methods, demonstrating superior generalisation capabilities and reduced bias.

The damage assessment strategy is based on the fact that 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, non-parametric machine learning supervised techniques are employed to train predictive 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. Their performance validation 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 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 (*data_test*). This novel real data was obtained from the sensors and has not been previously encountered by the models

(Figure 12.11). This new data for testing, which comprises more than 1000 samples, was collected on the aforementioned three damage pilot 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 the samples were labelled '0' for the healthy state and '1' for the damaged scenarios.

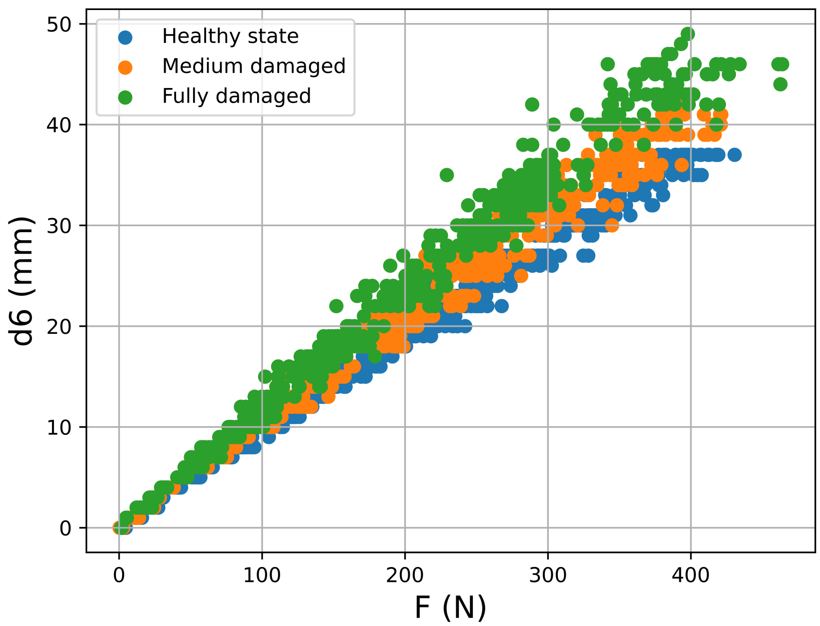


Figure 12.11: Real data never seen before by the models, obtained in a test and included in *data_test*.

The performance metrics employed in the classification tasks (Levels 1 and 2) were: accuracy (correctly predicted labels on the total), recall (ratio of true positives out of all correctly predicted values), and precision (fraction of true positives out of real and false positives). For regression tasks (Levels 3 and 4), the performance of the prognostic models has been evaluated using metrics like the Mean Squared Error (MSE) to measure the average squared difference between the estimated values and the true values, and its variant using absolute values: the Mean Absolute Error (MAE). In addition to this, the R^2 score or *coefficient of determination* will be also used. This coefficient evaluates the proportion of the variance in the dependent variable that is explained by the independent variables within the regression model, scoring from 0 to 1 (the greater, the better).

The collection of datasets employed in the development of this case study is outlined in Table 12.5, with their primary statistical descriptors included in Table 12.6. 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 hours. 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 (600k); nevertheless, it was noted that smaller sizes yield comparable results in training. Consequently, a second dataset (*data_GAN*) was generated with a reduced size (115k), yet it showed comparable performance.

Table 12.5: 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)	3mm 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)	3mm resolution	IoT sensors
data_stat	data produced with a FE model with added noise	600000	Numeric, Categorical	CSV	(t,F,d1,d2,d3, d4,d5,d6,label)	1mm	Statistical generation
data_GAN	synthetic data generated from the datasets *data_sens* and *data_stat*	115000	Numeric, Categorical	CSV	(t,F,d1,d2,d3, d4,d5,d6,label)	1mm	GAN generation

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

Table 12.6: 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

StDev: Standard Deviation, *CI*: Confidence Interval, IQR: Inter Quartile Range, d6: 6th floor-displacement (mm), F: force (N), t: time of damage (years)

12.4.2 Models training and deployment

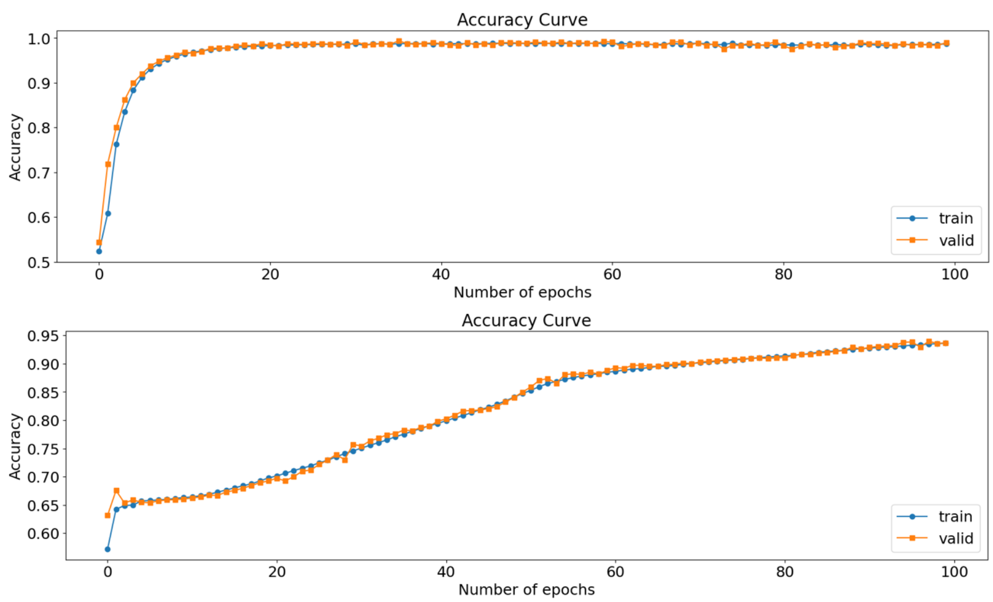
As previously mentioned, the damage assessment strategy consists of a pipeline of four predictive models: a binary classifier for damage detection, followed by a multiclass classifier for damage localisation and subsequently, regressors stand for the third and fourth levels of assessment. These models are DL hybrid models, as surrogates trained on a physics-informed foundational dataset. The process of training and deploying these models is outlined below.

Level 1 model for Damage Detection

The damage detection model consists of a binary classifier that discerns between a healthy or damaged state. This model was trained using first the *data_stat* obtained through the classic method and then on the *data_GAN* created via the generative approach.

The input layer of the damage detection model includes the available data, namely the displacements at the six floors of the structure and the force: ($d1$,$d2$,$d3$,$d4$,$d5$,$d6$,F), and the output has only one node for the binary classification *label*. In this model, the *Sigmoid* activation function outcomes a value between 0 and 1, representing the probability that the input belongs to the positive class (*label* $= 1$, meaning damage). The decision threshold was set to 0.5 to classify the predicted probabilities into 'healthy' or 'damaged' and the loss function for binary classification was the *binary cross-entropy*, to measure the difference between the predicted and the actual labels.

During the training, the differential learning rates observed in the model trained on the datasets *data_stat* and *data_GAN* can be attributed to their respective characteristics, as seen in Figure 12.12. The model trained on *data_stat* exhibits rapid learning due to the dataset being produced through statistical methods (Figure 12.12 (a)), facilitating the easy recognition of patterns. On the other hand, the model trained on *data_GAN* (Figure 12.12 (b)) displays a more gradual learning curve, given the generated dataset's increased variability and unstructured nature.



(a)

(b)

Figure 12.12: Learning curves of the level 1: damage detection binary classifier model trained on (a) *data_stat* and (b) *data_GAN*.

Performance metrics, including accuracy (correctly predicted labels over the total), recall (ratio of true positives out of all correctly predicted values), and precision (fraction of true positives out of the real and false positives), are employed to evaluate the model's performance trained on both datasets: *data_stat* and *data_GAN*, and the results are presented in Table 12.7 shown below.

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

Table 12.7: Comparative metrics for the level 1: damage detection model trained on different datasets (*data_stat* and *data_GAN*).

	Metrics								
	Train			Validation			Test		
Dataset	**ACC**	**REC**	**PREC**	**ACC**	**REC**	**PREC**	**ACC**	**REC**	**PREC**
data_stat	0.987	0.975	0.998	0.981	0.973	0.996	0.912	0.961	0.814
data_GAN	0.938	0.956	0.921	0.933	0.908	0.916	0.920	1.000	0.818

ACC: Accuracy, REC: Recall, PREC: Precision

model is trained on *data_GAN*. Attending to the test, results demonstrate that the models trained with *data_GAN* outperform the models trained without it, achieving better performance. This indicates that the GAN-generated dataset enhances the model generalisation capability, effectively capturing the noise present in the real data and enabling accurate diagnosis and prognostic predictions.

Furthermore, the results are analysed using a normalised confusion matrix [183], which categorises the model predictions into True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) based on the comparison between the predicted labels and the true labels. Figure 12.13 shows the normalised confusion matrixes for the test phase with the model trained on *data_stat* (Figure 12.13 (a)) and trained on *data_GAN* (Figure 12.13 (b)). It can be seen that the model trained on *data_GAN* demonstrated perfect accuracy (1.0) for detecting the healthy state in comparison with the 0.96 of accuracy achieved by the model when trained on *data_stat*.

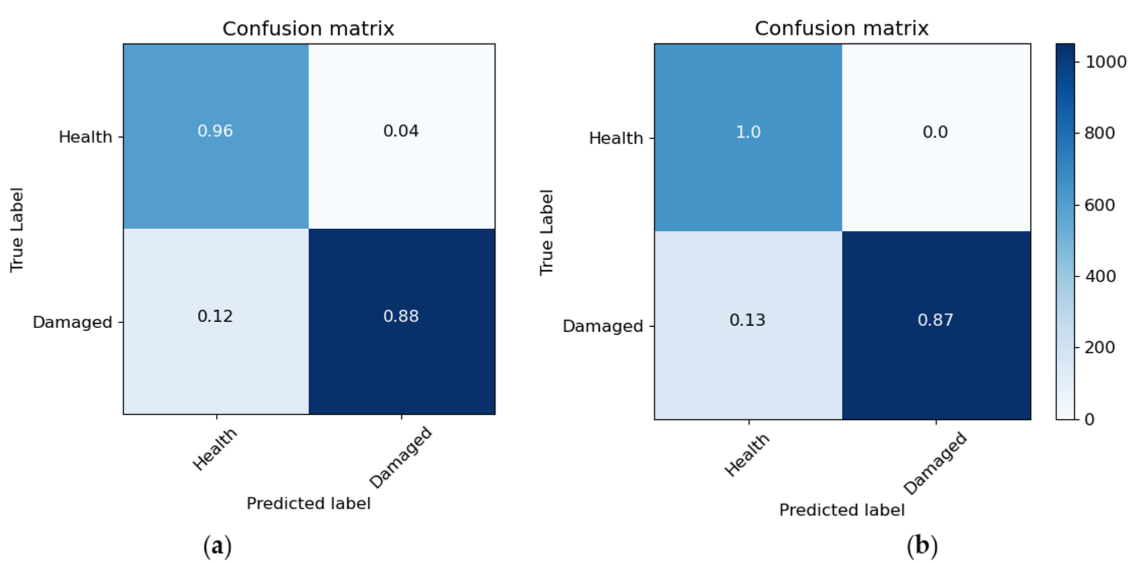


Figure 12.13: Normalised confusion matrices during testing for the level 1: damage detection binary classifier model trained on (a) *data_stat* and (b) *data_GAN*, performing on the new dataset *data_test*.

In addition, both cases showed similar figures for detecting the damaged state (0.88 and 0.87 respectively). It has been noticed that for earlier stages of damage that are often negligible, there is a tendency for false negatives in both cases. 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 (false negative), which can be risky. This tendency can be addressed by employing techniques such as oversampling early damage cases.

To graphically deploy the accuracy of a binary classifier, the Area Under (AUC) the ROC Curve [184] (Receiver Operating Characteristic) are used. A higher AUC and a ROC curve closer to the top left corner indicate better performance [185], as happens with the model trained using *data_GAN* (Figure 12.14 (b)).

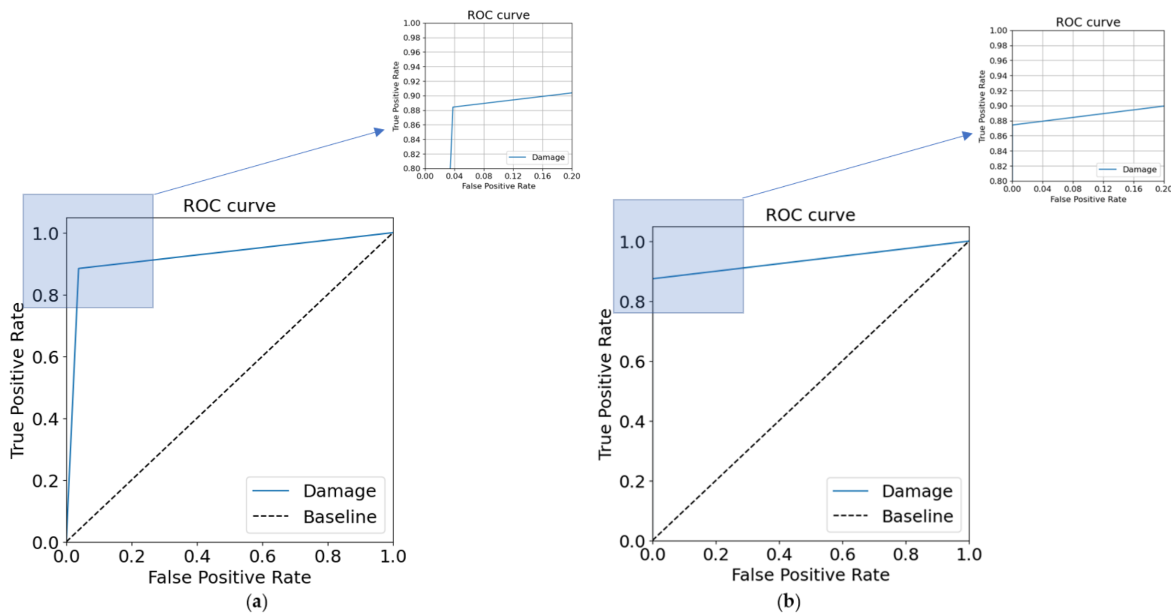


Figure 12.14: ROC curves for the level 1: damage detection model trained on (a) *data_stat* and (b) *data_GAN*, performing on the new dataset *data_test*.

Level 2 model for Damage Location

The damage location model is a 'single-label' multiclass classification model, meaning that the classes are mutually exclusive and each instance can belong to one single class. In the present case study, the damage can be identified only at one floor at a time.

To localise the damage, six labels are assigned, namely ($label1, label2, label3, label4, label5, label6$), each corresponding to damage on the designated floor. These labels constitute the classes in which the input data are categorised. As part of the supervised learning process, the classifier is trained using a labelled dataset. Each data point in this dataset is paired with an associated label. The model then learns to map input features to the correct class by leveraging this labelled training data. The activation function is the *softmax*, which converts the raw output scores into probabilities. Hence, the output of the classifier involves assigning a probability distribution across all classes and the class with the highest probability is predicted as the final class. The input in this problem is the same as in Level 1: Damage Detection;

a vector with the forces and displacements of the six floors of the structure $(F, t, d1, d2, d3, d4, d5, d6)$. In contrast, the output this time has 6 nodes, one for each of the classes of the multiclass classification $(label1, label2, label3, label4, label5, label6)$, representing the presence of damage at each floor. The loss function was the *categorical cross entropy*.

It is important to highlight that in order to achieve accurate results, the training datasets for the multiclass classification model, comprising both *data_stat* and *data_GAN*, are balanced across the six labels $(label1, label2, label3, label4, label5, label6)$.

During the training of the model on both datasets *data_stat* and *data_GAN*, a comparison of the learning curves reveals that the former (Figure 12.15 (a)) exhibits signs of overfitting. Notably, it demonstrates superior performance on the validation set compared to the training set, and the learning curve converges rapidly, reaching an accuracy rate of 1. Furthermore, it required only 100 epochs to reach convergence, a significant difference from the model trained on *data_GAN*, which necessitated up to 1400 epochs. The presence of intricate patterns, complexity, and variability in *data_GAN* necessitated a more extended learning process for the model to effectively capture and generalise the underlying relationships present in this data, reaching lower accuracy figures (close to 0.9).

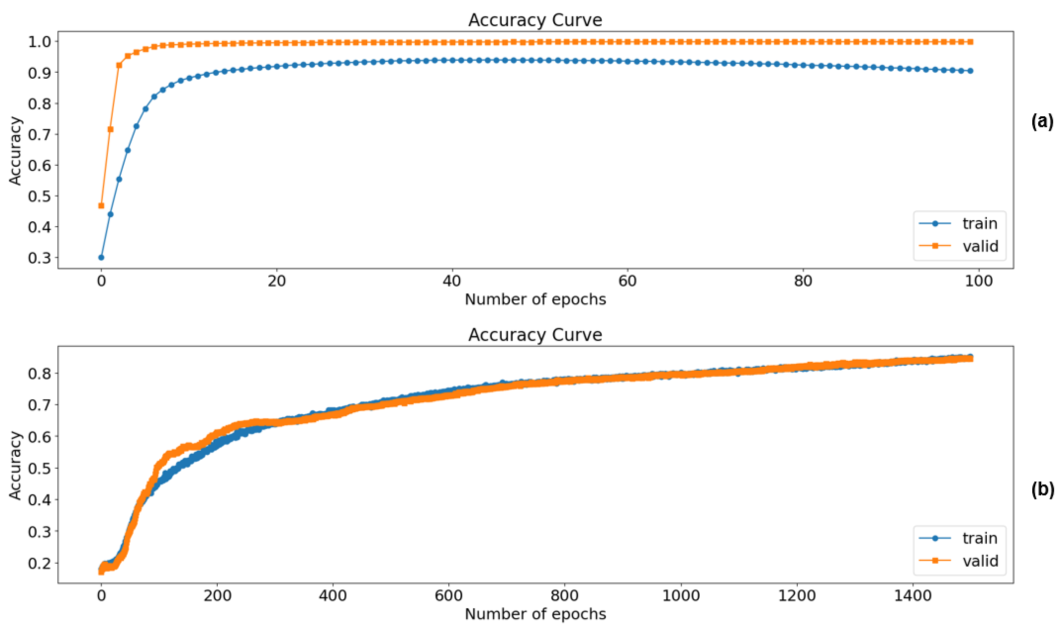


Figure 12.15: Learning curves of the level 2: damage location multiclass classifier model trained on (a) *data_stat* and (b) *data_GAN*.

The performance metrics for training, validation and testing are included in Table 12.8. From this table, it can be noted that the model trained on *data_stat* effectively exhibits overfitting. Its performance metrics on the test set were significantly worse than those on the training and validation sets due to the underperformance of the model in generalising effectively to new, unseen data.

Table 12.8: Comparative metrics for the level 2: damage location model trained on different datasets (*data_stat* and *data_GAN*).

		Metrics			
		Train		Test	
Label	Dataset	ACC	REC	ACC	REC
(d1,d2,d3,d4,d5,d6)	*data_stat*	1.000	(0.93,0.89,0.89,0.97,0.88,0.86)	0.846	(0.75,0.88,0.92,0.94,0.83,0.78)
(d1,d2,d3,d4,d5,d6)	*data_GAN*	0.898	(0.94,0.91,0.94,0.97,0.89,0.92)	0.874	(0.78,0.90,0.92,0.93,0.86,0.82)

ACC: Accuracy (total), REC: Recall (per label)

The normalised confusion matrixes of the model (Figure 12.16) reveal superior performance when trained on *data_GAN* than when trained on *data_stat*. Notably, lower figures are observed on floor 1 (d1), suggesting challenges in accurately predicting displacements at this floor due to their inherently small magnitudes.

Conversely, the model exhibits enhanced accuracy in predicting damage at floor 4 (d4). This could be attributed to a subtle balance between the magnitude of displacements (where larger displacements are easier to predict) and the influence of the number of floors beneath, making the predictions more difficult at higher floors.

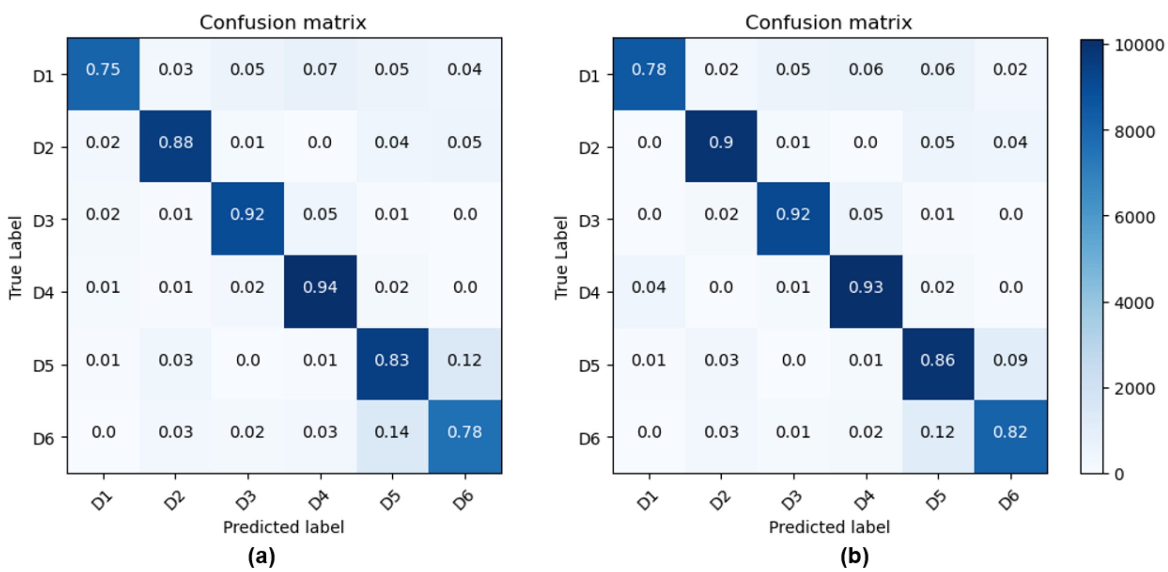


Figure 12.16: Normalised confusion matrices for the level 2: damage location multiclass classifier model trained on (a) *data_stat* and (b) *data_GAN*.

The AUC and ROC curves provide valuable insights into the model's ability to discriminate between classes in a multiclass setting (Figure 12.17). Noticeably, the model trained on *data_stat* (Figure 12.17 (a)) presents ROC curves closer to the diagonal line, indicating worse discriminatory capacity, along with a reduced ability to detect positive cases (damage) compared to the model trained on *data_GAN* (Figure 12.17 (b)).

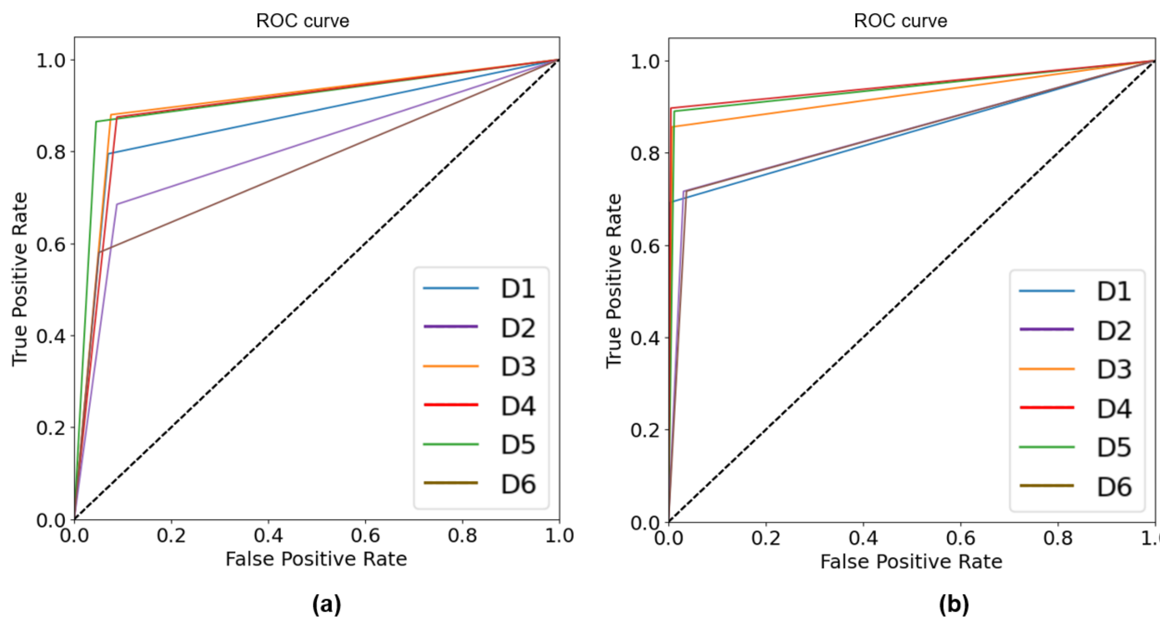


Figure 12.17: ROC curves for level 2: damage location model trained on (a) *data_stat* and (b) *data_GAN*, performing on the new dataset *data_test*.

Level 3 model for Damage Extent

To determine the extent of the damage, the regression model takes the force *(F)* and the displacements of each floor *(d1, d2,d3, d4, d5, d6)* as input. The output comprises the predicted *time of damage (t)*, which is used to calculate the *amount of damage* by adjusting Equation (5.10) to conform to Equation (12.6):

$$SDI = t/UL \tag{12.6}$$

where *SDI* is the severity damage index, *UL* is the useful life of the structure and t is the time of damage, directly correlated with the loss of stiffness in the structure, progressing linearly according to Equation (12.2).

The learning curves of the model, depicted in Figure 12.18, reveal distinctive patterns that align with the overarching trends observed throughout this study. Particularly, when employing dataset *data_GAN* for training (Figure 12.18(b)), the model shows a need for an increased number of training epochs, influenced by the high diversity of the dataset, requiring approximately three times the number of epochs compared to the model trained on dataset *data_stat* (Figure 12.18(a)). Furthermore, the curve exhibits a more gradual progression.

At the same time, the learning process during training and validation differs between the two datasets, as illustrated in Figure 12.19. Learning with *data_stat* (Figure 12.19(a)) demonstrates a more linear approach, while *data_GAN* (Figure 12.19(b)) explores a wider range of potential solutions.

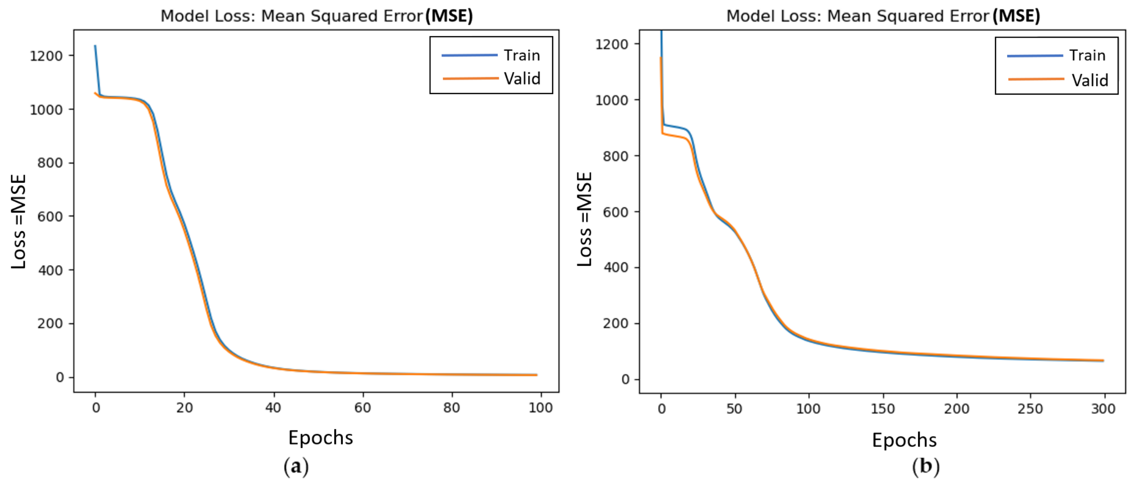


Figure 12.18: Learning curves of the regression model trained on (a) *data_stat* and (b) *data_GAN*.

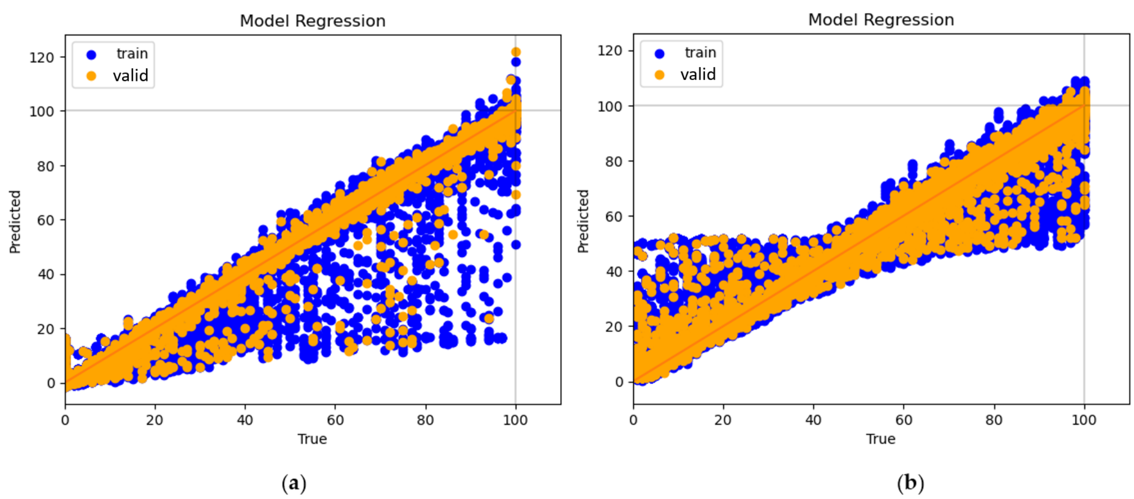


Figure 12.19: *Predicted values* versus *True values* during training with (a) *data_stat* and (b) *data_GAN*.

The performance of the 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 Mean Absolute Error (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, ranging between 0 and 1 with higher values being the better.

Consistently observed throughout this study, the metrics in both training and validation phases favour the model trained on *data_stat* over the model trained on *data_GAN*. However, in all test phases, the model trained on *data_GAN* outperforms its counterpart, demonstrating superior generalisation skills and an ability to handle unforeseen scenarios adeptly. Once again, the metrics pertaining to the test phase demonstrate the superior performance of the model trained on *data_GAN*.

Table 12.9: Comparative metrics for the level 3: damage extent predictive model trained on different synthetic datasets (*data_stat* and *data_GAN*.

	Metrics								
	Train			Validation			Test		
Dataset	**MSE**	**MAE**	R^2	**MSE**	**MAE**	R^2	**MSE**	**MAE**	R^2
data_stat	5.080	0.850	0.995	5.371	0.851	0.994	273.548	12.046	0.843
data_GAN	91.898	5.106	0.909	93.876	5.109	0.911	264.601	11.746	0.871

MSE: Mean Squared Error, MAE: Mean Absolute Error, R^2: Coef. of Determination.

Level 4 model for Damage Prediction

As already anticipated, the model employed for damage extent prognostic enables the estimation of the structural RUL by predicting the progression of the damage once detected. The maximum available structural useful life (UL) is estimated for the structure to be equal to one hundred (100) years. Hence, the RUL is calculated using the following Equation (12.7), where t_i is the predicted time of damage at a discrete time i, expressed in years:

$$RUL(years) = 100 - t_i \tag{12.7}$$

The regressor specifically forecasts the time during which the structure has been suffering damage since it began, taking into account the applied force and the displacements measured at each of the six floors of the structure. The smaller the RUL is, the sooner the end of the useful life of the structure is.

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 Table 12.10 and Figure 12.20.

Table 12.10: Comparison of predicted values distribution obtained from the models trained on different synthetic datasets (*data_stat* and *data_GAN*) and tested on the real data (*data_test*)

	Properties of the predicted values distributions							
Dataset for the training:	**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

Q1: Percentile 25%, Q3: Percentile 25%, IQR: interquartile range, MIN: Minimum value, MAX: Maximum value, OUT: Outliers

As it can be seen in the Figure 12.20 , the interquartile range exhibited a wider span. However, the predicted median showed greater proximity to the actual value and, importantly, no outliers were observed.

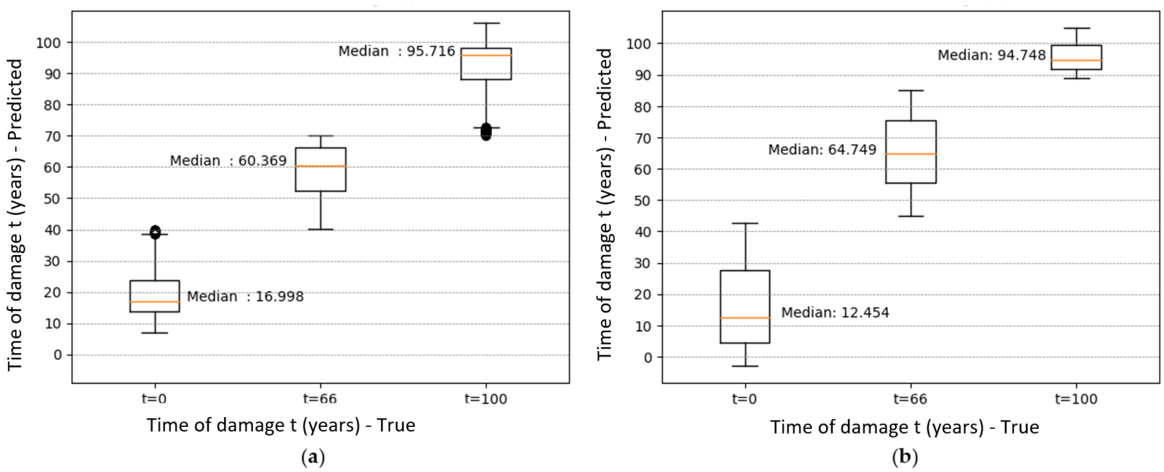


Figure 12.20: Time of damage boxplots predicted by the prognostic models; (a) model trained on $data\_stat$ and (b) model trained on $data\_GAN$.

12.4.3 Results and discussion

In this case study, a framework that integrates a generative model mimicking the data from the real system alongside the inclusion of a pipeline of predictive models trained on the generated data for online damage assessment has been put in practice. Both components have worked synergistically to address key challenges in the DT by providing valuable capabilities for mirroring and optimising decision making in real-world systems.

Firstly, the generation of quality data for model training is essential to overcome limitations associated with potentially scarce or incomplete data. DT strive to accurately simulate and predict the behaviour of complex systems, and having a method for generating additional synthetic data ensures a more comprehensive and diverse training set. This, in turn, enhances the accuracy and generalisation capabilities of the DT models.

Simultaneously, the inclusion of models for online damage assessment is pivotal for real-time diagnostics and prognosis, together with a proactive intervention. Systems in the real world are dynamic and subject to constant changes. Models for online damage assessment enable the DT to continuously evaluate the current state of the physical system, detect anomalies or potential issues in real time, and provide timely insights for decision making, which can be autonomous or assisted through a human-in-the-loop.

The framework employed in this case study encompasses the data generation, predictive 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 the predictive surrogate models. These models possess a deep understanding of a wide range of possible scenarios, as they were trained not only on monitored data but also on synthetic datasets generated by the proposed procedure.

To validate the efficacy of the exposed framework, a set of models for a 4-level damage assessment strategy was trained and evaluated in a laboratory-scale structure as a case study. Both generative and predictive models were built on state-of-the-art AI methods such as CNN and CWGAN-GP, sharing a common architecture adapted to their respective tasks. The results obtained from the GAN-generated dataset were compared to a dataset created using traditional statistical techniques (e.g. FE results with added heteroscedastic Gaussian noise), demonstrating superior performance in the former case. Subsequently, the trained models were integrated into the DT, thereby enabling its decision-making capabilities in real-time scenarios.

12.5 Inclusion into the DT worflow

This section shows how the CNN-based predictive models trained with synthetically CWGAN-GP - 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 a change in stiffness, 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-based framework, which represents the logic of discrete events in a dynamically distributed system and effectively handles the workflow of the DT [46]. PNs can represent the structural components, interactions, and transitions in a way that captures the system's temporal and spatial aspects. Furthermore, when there is a preference to restrict the degree of automation within the workflow, a human-in-the-loop can be incorporated. In this way, interaction is enabled and human operators can influence the structural behaviour or decision-making processes within the DT.

In Figure 12.21, a Petri net designed for the presented case study is illustrated, consisting of ten places (p_1 to p_{10}) representing discrete system states, nine transitions (t_1 to t_9) denoting symbolic or conditional actions (including post-firing actions), and two cold transitions ϵ for data arrival and system rearm. Changes in the states of the DT are the outcome of automated actions triggered by firing transitions t_1 to t_9.

It is assumed that the system is initiated at time $T = 0$ when new data are received from the force and displacement sensors. At this point, the virtual twin is updated in relation to the physical twin, as one token arrives at p_1. Subsequently, firing transition t_1 occurs, placing one token in p_2. Such 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_9, each based on their respective transition conditions (Table 12.11).

When t_2 is activated, indicating a healthy state, a token is produced for p_7, meaning that the DT

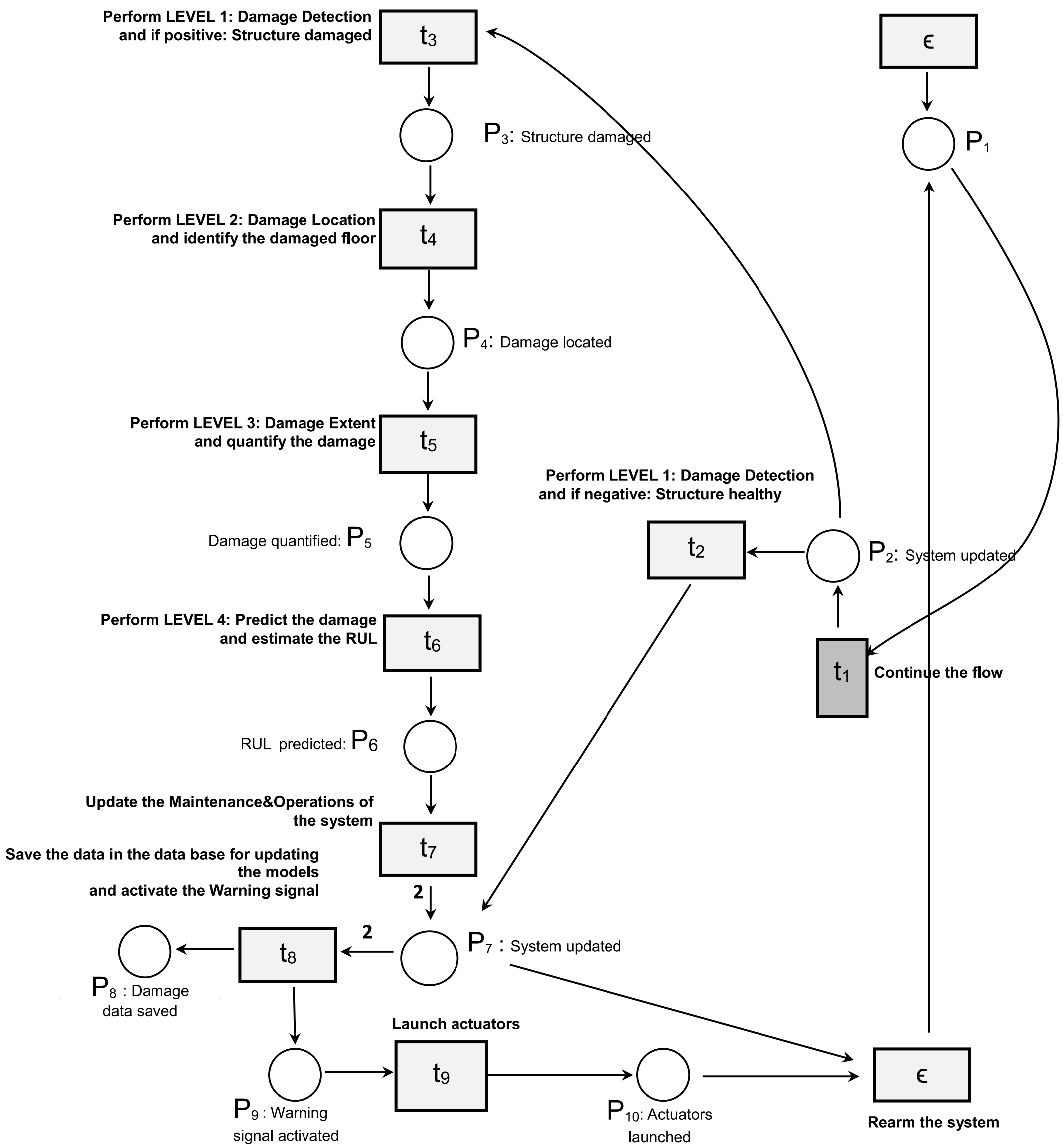


Figure 12.21: PN workflow of a DT assisted by the predictive models for the 4-level damage assessment
.

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 in t_3, the workflow sequence $p_3, t_4, p_4, t_5, p_5, t_6, p_6, t_7, p_7$ will occur, placing one token in p_7 and updating the system to the current state.

Table 12.11: Transitions in the PN-assisted workflow of the DT.

Transition	Description
t_1	If new data arrives, continue the flow
t_2	Perform level 1: damage detection and if negative, the structure is healthy
t_3	Perform level 1: damage detection and if positive, the structure is damaged
t_4	Perform level 2: damage location and determine on which floor is the damage located
t_5	Perform level 3: damage extent and quantify the damage through the SDI
t_6	Perform level 4: damage prediction and estimate the RUL
t_7	Update the Maintenance & Operation policy of the system
t_8	Save the data in the data base and activate the warning signal
t_9	Launch actuators

At this point, the data corresponding to the four levels of the damage assessment strategy are saved in the database at t_8, which also triggers a sequence of warning states and actions represented by nodes p_9, t_9, p_{10}, that autonomously indicate that the structure is exposed to a force that may compromise its integrity. In such cases, actuators trigger an alarm (through the firing of t_{10}), and the system enters a 'warning state' that demands predictive maintenance based on the damaged extent (SDI) and the estimated Remaining Useful Life (RUL), or alternatively, corrective actions if the threshold has been surpassed. 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.

This workflow has the capability to operate in real-time thanks to the surrogate predictive models of the 4-level damage assessment. These models were effectively trained with CWGAN-GP generated synthetic datasets, which conferred valuable properties of interpreting raw data directly from sensor measurements and demonstrated generalisation abilities to make accurate predictions.

12.6 Conclusion

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. The data created by the generative setting conferred the predictive model an enhanced robustness to variability, enabling an accurate damage assessment through the 4 levels: detection, location, extent and prediction.

The present research has the potential for broader applicability beyond the civil engineering field, offering valuable insight into sectors in which main features can be monitored and modelled to be seamlessly integrated into a DT implementation. 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.

Finally, the impact of this research extends beyond the experimental setting. The presented DT framework driven by AI-generated data for model training, addresses several critical challenges in the civil engineering sector. It tackles the complexities associated with generating realistic and diverse training data, enabling the effective learning of the AI models for an accurate damage assessment. The framework extends its scope to encompass the intricate tasks of structural diagnosis and prognosis, leveraging the capabilities of AI to enhance predictive analytics. It further addresses challenges related to the integration of AI seamlessly into DT, handling unforeseen scenarios, capturing dynamic structural behaviour, and accommodating human-in-the-loop interactions for balanced decision making. Additionally, the framework strives to enhance model generalisation across different domains, ensuring cybersecurity and sustainability. Through this comprehensive approach, the presented framework aims to contribute to the advancing of a robust and efficient DT application in civil engineering and the AECO sector.

Future research endeavours should focus on expanding the capabilities of the framework to other types of structures and signals, with an increased level of complexity. The inclusion of the IFC standards and the BIM taxonomy together with the uncertainty quantification for risk-based decisions are also crucial steps for advancing the operations and maintenance strategy of the DT.

Part IV

Conclusions and future works

Chapter 13

Conclusiones y trabajo futuro

La ingeniería civil es uno de los sectores que más contamina y menos digitalizado está en pleno siglo XXI. Se trata de un sector que mueve grandes cifras económicas, es muy conservador, y tiene una gran inercia para adaptarse a los cambios, al reunir grupos implicados de carácter muy diferente: productores de materias primas y energía, empresas constructoras, gobiernos, y la sociedad civil como usuaria.

A esta situación se le une el hecho de que la mayoría de las grandes obras de ingeniería civil en nuestro país y el resto del mundo (puentes, presas, grandes edificios. . .), están llegando al final de su vida útil de diseño, calculada en 100 años según las normativas, ya que fueron ejecutadas a principios del siglo XIX. El gasto en mantenimiento es muy elevado, sin embargo no es eficiente y en ocasiones llega demasiado tarde, cuando la infraestructura falla y puede ocasionar daños humanos, además de materiales.

El cambio climático y los diferentes usos y acciones a los que se ven sometidas las infraestructuras también suponen una amenaza para las mismas, ya que por ejemplo, las velocidades y el tráfico de vehículos no son los mismos que hace 50 años, ni los caudales de diseño o los registros sísmicos.

La Unión Europea así como Naciones Unidas presentan dentro de la agenda 2030 varios objetivos (SDG.9, Industry, Innovation, and Infrastructure y SDG11. Sustainable Cities and Communities) dedicados a la sostenibilidad a través de la innovación en la industria y la construcción, que directamente se alinea con el propósito de la presente investigación: la aplicación del gemelo digital en la ingeniería.

Con la introducción del gemelo digital se pretende lograr que las tecnologías que actualmente funcionan en silos, se coordinen de manera sinérgica y eficiente para representar en tiempo real el estado de una infraestructura y conocer su salud estructural, con el fin de tomar las decisiones óptimas para su explotación y mantenimiento de acuerdo a sus nuevas condiciones. De esta manera se amplia la vida útil de las infraestructuras existentes haciéndolas más sostenibles, se reduce el gasto y se aumenta la seguridad.

Con el fin de servir de acelerarador para la plena implantación del gemelo digital en la ingeniería civil se ha desarrollado la presente tesis, cuyas conclusiones se establecen a continuación con respecto a cada hipótesis y objetivo planteado.

1. ***Hipótesis 1:*** *El paradigma del gemelo digital requiere de una definición clara y un soporte a través de aplicaciones prácticas ilustrativas para ganar impulso y lograr una adopción generalizada dentro del campo de la ingeniería civil. Actualmente, la tecnología se emplea en silos, de manera aislada, y existe una necesidad apremiante de digitalización para mejorar la eficiencia y la sostenibilidad de cara a la Agenda 2030 de la Comisión Europea y Naciones Unidas.*

   Para responder a esta necesidad y explorar esta hipótesis, se ha desarrollado una pormenorizada conceptualización del gemelo digital en la ingeniería civil en el Capítulo 6 partiendo de la revisión bibliográfia recogida en el Capítulo 2, incluyendo aspectos tan relevantes como el caracter interdisciplinario del gemelo digital (Sección 6.1), su principal propósito y objetivos (Sección 6.3) y las fuentes de datos de las que se alimenta (Sección 6.5). Se ha desarrollado el concepto desde el punto de vista matemático y computacional (Sección 6.2) y se ha descrito cada componente al detalle (Sección 6.4), acompañando todo ello de una serie de imágenes ilustrativas.

   Además, esta tesis incluye la implementación de dos casos de estudio que sirven como aplicaciones prácticas del gemelo digital en ingeniería civil. El primer caso (Capítulo 11) desarrolla el gemelo digital de una torre metálica en 2D y está especialmente enfocado a la integración de la tecnología que respalda al gemelo digital y a la utilización del marco Bayesiano para resolver el problema inverso. El segundo caso de estudio (Capítulo 12) implementa el gemelo digital de una torre en 3D a mayor escala y se concentra en el entorno generativo y la implementación de modelos de evaluación del daño dentro del gemelo digital. En ambos casos, se ha prestado atención especial a la implementación de la gestión del flujo de trabajo del gemelo digital.

   Como conclusión, cabe comentar que la exploración pormenorizada del concepto de gemelo digital en ingeniería civil, incluyendo una definición completa, descripciones detalladas de sus componentes y varios estudios de casos prácticos, subraya su potencial como disruptor y cohesionador de tecnologías. La aplicación exitosa del gemelo digital muestra su naturaleza interdisciplinaria, requiriendo la colaboración de la ingeniería, la ciencia, y la informática, entre otras. La integración de tecnologías avanzadas como la inteligencia artificial resalta la importancia de la innovación tecnológica para realizar todo el potencial del gemelo digital. Además, la introducción del marco Bayesiano para la cuantificación de la incertidumbre y la atención a la gestión del flujo de trabajo, asegura un funcionamiento efectivo con gestión del riego en escenarios del mundo real.

   En el futuro, la investigación debe centrarse en validar modelos para su uso dentro del gemelo digital, mejorar la adaptación dinámica a condiciones cambiantes, y escalar los gemelos digitales para bienes y procesos de mayor tamaño y complejidad. La mejora en la integración con dispositivos IoT y las contribuciones a los esfuerzos de estandarización avanzarán aún más en el campo y facilitarán una adopción más amplia del gemelo digital en la ingeniería civil.

2. ***Hipótesis 2:*** *El enfoque Bayesiano puede ser incorporado para cuantificar la incertidumbre tanto en los modelos como en los datos, con el fin de realizar una evaluación de riesgos más fiable y un proceso de toma de decisiones informado dentro del contexto del gemelo digital en la ingeniería civil.*

   Con el fin de integrar una herramienta adecuada para la cuantificación de la incertidumbre dentro del gemelo digital se ha adoptado el marco Bayesiano (Capítulo 7), el cual se ha aplicado en dos áreas clave: primero, para abordar el problema directo que implica la propagación de la incertidumbre, como se explica en la Sección 7.1. En segundo lugar, se ha utilizado en el problema inverso en la actualización de los parámetros del modelo y la inferencia de magnitudes desconocidas, como se detalla en la Sección 7.2.

   Con visión a futuro, se necesita más investigación para la integración de los métodos Bayesianos con otras tecnologías, con el fin de reducir el tiempo de ejecución de procesos de sampleo e iteración, y automatizar al máximo el ajuste de hiperparámetros.

3. ***Hipótesis 3:*** *La implementación de una herramienta dedicada es crucial para orquestar el flujo de trabajo del gemelo digital en ingeniería civil, garantizando la colaboración eficiente entre los diversos elementos, la gestión simplificada y la adaptabilidad a condiciones cambiantes. Las redes de Petri son un método adecuado para gestionar el flujo de trabajo del gemelo digital en aplicaciones de ingeniería civil.*

   En esta tesis, la gestión del flujo de trabajo del gemelo digital se ha realizado mediante el uso de redes de Petri, método elegido por su idoneidad para manejar entornos operativos complejos y dinámicos (Capítulo 8). La Sección 8.1 detalla la descripción del flujo de trabajo del gemelo digital con sus distintas etapas y procesos, mientras que la Sección 8.2 proporciona una explicación exhaustiva de cómo se gestiona este flujo, incluyendo un ejemplo ilustrativo para mejorar la comprensión. Tras este desarrollo se observa que a través de una red de Petri se puede garantizar la gestión efectiva del flujo de trabajo del gemelo digital, permitiendo una navegación eficiente a través de procesos síncronos, concurrentes y paralelos, teniendo en cuenta la logística y la cronología en cada proceso.

   Como trabajo futuro se pueden explorar enfoques para la mejora continua de la gestión del flujo de trabajo del gemelo digital, como la implementación de sistemas de retroalimentación y aprendizaje automático por refuerzo (reinforced learning) para identificar y corregir posibles deficiencias o áreas de mejora, así como la posibilidad de realizar aprendizajes por tranferencia (transfer learning) para escalar modelos reducidos a otros similares de mayor tamaño.

4. ***Hipótesis 4:*** *El desarrollo de una metodología basada en inteligencia artificial para generar datos en calidad y cantidad destinados al entrenamiento efectivo de los modelos de DT, puede superar los desafíos relativos a la escasez de datos. Además, también se contribuye a combatir los problemas relacionados con la privacidad y seguridad de los datos, la falta de interoperabilidad y la sobrecarga de tráfico, al hacer posible que los modelos entrenados permitan la computación "in the edge".*

   El desafío planteado por la escasez de datos de calidad para el entrenamiento de los modelos del gemelo digital, especialmente en las etapas iniciales cuando los datos de monitorización son escasos, puede llevar a modelos poco precisos, dificultades para capturar la variabilidad y la representación limitada de las características. Con el tiempo, a medida que el gemelo digital acumula datos de los sensores, los modelos pueden ser continuamente refinados con el fin de mejorar su precisión y efectividad. Sin embargo, el desafío asociado con los datos persiste, ya que existen altos costos asociados con los sistemas de adquisición de datos y las sobrecargas en el tráfico de grandes volúmenes de datos, junto con el cumplimiento de las políticas de privacidad y seguridad relacionadas con su manejo. Además hay que añadir los problemas de interoperabilidad al integrar diversas fuentes de datos y tecnologías, exacerbados por la falta de estandarización en el sector.

   Con el fin de dar respuesta a este desafío y probar la hipótesis correspondiente, en la presente tesis se ha desarrollado un entorno generativo basado en inteligencia artificial (Capítulo 9) para suministrar datos en calidad y cantidad suficiente para el entrenamiento efectivo de los modelos del gemelo digital, reduciendo los costes y el tiempo asociado, mejorando la eficiencia computacional, cumpliendo con los estándares de formato de datos y garantizando la ciberseguridad. Este entorno favorece estrategias como la generación de datos sintéticos, la ampliación de bases de datos existentes y la inclusión del conocimiento experto y la información a través de los modelos físicos, junto con la computación de los modelos "in the edge". La computación "in the edge" mejora las capacidades del DT al proporcionar una operación fluida cerca de la fuente de datos, lo que aumenta la ciberseguridad, reduce la latencia de la transmisión de datos y disminuye el tiempo de respuesta para la toma de decisiones en tiempo real.

   Con enfoque a futuro, se pueden investigar distintos planteamientos para integrar de manera más efectiva el conocimiento experto y la información de modelos físicos en el proceso de generación de datos, lo que podría mejorar la representación de información relevante en los modelos del gemelo digital. Igualmente el desarrollo de métricas más representativas de acuerdo con el tipo de datos generado haría posible una evaluación de los datos generados más rápida y efectiva, lo que redundaría en la calidad de los modelos entrenados sobre ellos.

5. ***Hipótesis 5:*** *El desarrollo de modelos surrogados, que son representaciones simplificadas de modelos o sistemas más complejos, facilita una aproximación precisa de manera computacionalmente eficiente. Diseñados para capturar características y relaciones esenciales dentro de los datos, estos modelos surrogados mantienen la eficiencia al mismo tiempo que habilitan el rendimiento en tiempo real del gemelo digital.*

   Otro desafío surge de la complejidad de los modelos a utilizar por el gemelo digital. A pesar de proporcionar una representación detallada de la realidad, estos modelos a menudo tienen altas demandas computacionales, requieren datos extensos y de alta calidad, enfrentan problemas de escalabilidad y carecen de interpretabilidad. Además, presentan desafíos en la calibración y validación, son susceptibles al sobreajuste, tienen dificultades con la adaptación dinámica, conllevan altos costos de mantenimiento, y demandan recursos y habilidades sustanciales para su mantenimiento y actualización. Todo ello plantea la necesidad de crear modelos surrogados (también llamados sustitutos o metamodelos) como representaciones simplificadas de modelos o sistemas de mayor complejidad, que aproximen su comportamiento de manera adecuada pero computacionalmente eficiente. Estos modelos surrogados deben estar diseñados para capturar las características esenciales y las relaciones dentro de los datos, al tiempo que mantienen su eficiencia para permitir el buen rendimiento en tiempo real del gemelo digital. En el ámbito de la ingeniería civil, estos modelos se centran en la implementación de una estrategia de evaluación de daños para predecir los fallos antes de que ocurran y poder ejecutar las acciones de mantenimiento y/o prevención que resulten convenientes con el fin de ahorrar costes y aumentar la seguridad.

   En la presente tesis se considera que una de las principales competencias que permite al gemelo digital facilitar la toma de decisiones en la ingeniería civil es la implementación de la capacidad de evaluación de daños. Esta capacidad se despliega en el Capítulo 10 de esta tesis a través de un conjunto de modelos subrogados diseñados para realizar una evaluación de daños de cuatro niveles (detección, localización, cuantificación y predicción del fin de la vida útil) en tiempo real. La metodología para implementar efectivamente estos modelos se detalla en la Sección 10.1, mientras que las funcionalidades de los modelos para cada nivel se detallan en la Sección 10.2.

   Sería deseable en próximos trabajos mejorar tanto la explicabilidad como la precisión de los modelos subrrogados, que en ocasiones debido a la simplificación de los algoritmos y el empleo de técnicas como la reducción de dimensiones, se convierten en cajas negras difíciles de interpretar.

Chapter 14

Conclusions and future works

Civil engineering, despite its significance, faces several challenges in the XXI century. Firstly, it remains one of the most polluting industries due to its reliance on traditional construction methods and materials. Additionally, the sector has been slow to adopt digital technologies compared to other industries, leading to inefficiencies and missed opportunities for innovation.

One major issue confronting civil engineering is the ageing infrastructure. Many of the monumental structures built in the early XIX century are nearing the end of their design lifespan. This poses a dual challenge: the high cost of maintaining ageing infrastructure and the potential risks associated with infrastructure failures, which can have catastrophic consequences.

Furthermore, infrastructure must adapt to changing environmental conditions and different uses and actions. Climate change is altering weather patterns and increasing the frequency and severity of extreme events such as floods and earthquakes. Meanwhile, urbanisation and population growth are placing greater demands on infrastructure, requiring it to accommodate higher volumes of traffic and support denser populations.

To address these challenges, there is a growing recognition of the need for sustainable and innovative approaches to civil engineering. The European Union and the United Nations have set ambitious sustainability goals, including SDG 9 (Industry, Innovation, and Infrastructure) and SDG 11 (Sustainable Cities and Communities), which emphasise the importance of innovation in construction and maintenance.

A promising solution is the application of DT technology in civil engineering. DTs are virtual representations of physical assets or systems that can simulate their behaviour in real time. By deploying DTs, engineers can monitor the condition of infrastructures in real time, predict maintenance needs, and optimise their performance. This can extend the lifespan of existing infrastructure, reduce maintenance costs, and increase safety.

This thesis aims to explore the potential of DT technology in civil engineering and provide insights into its implementation. By developing a comprehensive understanding of the challenges and opportunities associated with DTs, this thesis seeks to accelerate their adoption and contribute to the advancement of sustainable and resilient infrastructure.

1. ***Hypothesis 1:*** *The DT paradigm needs clear definition and practical examples to gain traction and widespread adoption in civil engineering, which currently lacks digitalisation and faces significant sustainability challenges.*

   To address this need and explore this hypothesis, a detailed conceptualisation of DT in civil engineering has been developed in Chapter 6, starting from a literature review included in Chapter 2. Relevant aspects of the DT are also considered, such as its interdisciplinary nature (Section 6.1), data sources (Section 6.5), and its main purpose and objectives (Section 6.3). The DT concept has been described from both mathematical and computational perspectives (Section 6.2), including an extensive description of each component in Section 6.4, accompanied by illustrative images.

   Furthermore, this thesis includes the deployment of two case studies serving as practical applications of DT in civil engineering. The first case study (Chapter 11) focuses on the development of a DT of a 2D metal tower, focused on the integration of the technology supporting the DT and the inclusion of the Bayesian framework to undertake the inverse problem for inferring an unknown magnitude of interest. The second case study (Chapter 12) implements a DT of a 3D larger-scale tower and concentrates on the generative environment and the implementation of damage assessment models within the DT. In both cases, particular attention has been given to implementing the DT workflow management.

   In conclusion, the elaborated exploration of the DT concept in civil engineering presented in this thesis, including its comprehensive definition, detailed descriptions of its components, and various practical case studies, underscores its potential as a disruptive and integrative approach for different technological domains. The successful application of the DT demonstrates its interdisciplinary nature, requiring collaboration from engineering, science, and computing, among others. The integration of state-of-the-art technologies such as AI highlights the importance of technological innovation to realise the full potential of the DT. Furthermore, the introduction of the Bayesian framework for uncertainty quantification and the attention to workflow management ensures effective operation with risk management in real-world scenarios.

   In the future, research should focus on validating models for use within the DT, improving dynamic adaptation to changing conditions, and scaling for larger and more complex assets and processes. Improvements in integration with IoT devices and contributions to standardisation will further advance the field and facilitate broader adoption of DT in civil engineering.

2. ***Hypothesis 2:*** *The Bayesian approach can be incorporated into the DT framework to quantify uncertainty in both models and data for a more reliable risk assessment and informed decision-making process within the context of civil engineering, despite model simplifications and noisy sensor data.*

   In order to integrate a systematic approach to incorporate uncertainty quantification within the DT, a Bayesian framework has been adopted in Chapter 7 and applied in two key areas.

   Firstly, it addresses the direct problem of uncertainty propagation in Section 7.1, where uncertainties in input parameters propagate through the model to generate uncertainty in the output predictions. By employing Bayesian methods, it becomes possible to quantify and propagate uncertainties effectively, providing a more comprehensive understanding of the uncertainty inherent in the DT predictions.

   Secondly, the Bayesian framework is utilised in the inverse problem in Section 7.2, which involves updating model parameters and inferring unknown quantities based on observed data as in Section 11.3. This is particularly valuable in scenarios where certain parameters or variables cannot be directly measured but need to be estimated based on available data.

   Looking ahead, further research is needed for the integration of Bayesian methods with other technologies, aiming to reduce the runtime of sampling processes and iterations and to fully automate hyperparameter tuning to streamline the process. By continuing to develop and refine Bayesian approaches within the DT context, the capability to provide accurate and reliable predictions for decision making in complex engineering systems will be enhanced, providing a more comprehensive understanding of the uncertainty inherent in the DT predictions.

3. ***Hypothesis 3:*** *A dedicated tool is essential for orchestrating the DT workflow in civil engineering, coordinating real-time monitoring, data sources, analytics, and management, and adapting promptly to changes. Petri nets are well suited for representing and managing the dynamic, event-driven behaviour of such systems.*

   In this thesis, the management of the DT workflow has been addressed using a Petri net, a modelling tool selected for its effectiveness in handling the complexities and dynamic nature of the civil engineering operational environments (Chapter 8). Section 8.1 delves into the intricacies of the DT workflow, outlining its different stages and processes. This section provides a comprehensive understanding of how the DT operates within its environment. Additionally, Section 8.2 goes into depth on the management of the DT workflow. It offers a comprehensive explanation of how this workflow is structured and orchestrated, providing insights into the mechanisms ensuring a smooth operation. Additionally, an illustrative example is provided to enhance understanding. Through this development, it is observed that the use of a Petri net ensures effective management of the DT workflow, enabling efficient navigation through synchronous, concurrent, and parallel processes, taking into account the logistics and chronology at every step.

In future work, approaches for the continuous improvement of DT workflow management can be explored, such as the implementation of feedback systems and reinforcement learning to identify and correct potential deficiencies or areas of improvement. Additionally, the possibility of transfer learning can be considered to scale reduced models to similar larger-sized ones.

4. ***Hypothesis 4:*** *An AI-based approach to generate sufficient quality and quantity of data can address challenges such as data scarcity, privacy, interoperability and traffic overload, improving the training of the DT models and enabling them to perform at the edge.*

   To investigate this hypothesis, this thesis has developed a generative environment based on AI (Chapter 9) to supply data in sufficient quality and quantity for effective training of DT models, reducing associated costs and time, improving computational efficiency, complying with data format standards, and ensuring cybersecurity. This setting supports strategies such as synthetic data generation, augmentation of existing databases, and the incorporation of expert knowledge, domain expertise and physics information guidance, along with the computation of models "in the edge". 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 diminishes the response time for real-time decision making.

   In the future, various approaches can be explored for better integrating expert knowledge and information from physical models into the data generation process, potentially enhancing the representation of relevant information in the DT models. Similarly, enhancing the development of more relevant metrics tailored to the type of generated data could facilitate faster and more effective evaluation of the datasets, ultimately resulting in improved model quality after training.

5. ***Hypothesis 5:*** *Surrogate models simplify complex systems, enabling accurate approximation in a computationally efficient manner, ensuring real-time performance for the DT despite model complexity, computational demands, data requirements, and scalability challenges.*

   In this thesis, it is argued that one of the key competencies empowering the DT to streamline decision making in civil engineering lies in its ability to conduct damage assessment. This capability is explored in Chapter 10, where a pipeline of surrogate models is employed to execute a real-time, four-level assessment process encompassing detection, localisation, quantification, and RUL prediction. The methodology for effectively deploying these models is described in Section 10.1, while the specific functionalities of each model level are elucidated in Section 10.2.

   Moving forward, there is a clear imperative to enhance both the transparency and accuracy of these surrogate models. At present, their interpretability is hindered by the simplification of algorithms and the utilisation of techniques like dimensionality reduction, rendering them opaque 'black boxes' in terms of understanding their inner workings.

Thus, future research endeavours should prioritise strategies aimed at improving both the precision and clarity of these surrogate models to ensure their utility and trustworthiness in DT decision-making processes within civil engineering contexts.

Part V

Appendixes

Appendix A

Research records

A.1 Journal articles

The approaches and outcomes outlined in this thesis, alongside further contributions from the author, have been partially disseminated through the following publication:

- Manuel Chiachío, María Megía, Juan Chiachío, Juan Fernandez, María L. Jalón, Structural digital twin framework: Formulation and technology integration, Automation in Construction, Volume 140, 2022, 104333, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2022.104333. Keywords: Digital twin; Petri nets; Bayesian learning; Internet of things; Structural health monitoring

This article is pending of approval:

- María Megía, Francisco Javier Melero, Manuel Chiachío, Juan Chiachío, Generative Adversarial Networks for Improved Model Training in the Context of the Digital Twin, Structural Control and Health Monitoring. Keywords: Digital Twin; Generative Adversarial Networks; Convolutional Neural Networks; Synthetic Data

And this research is being prepared for submission to a journal:

- María Megía, Francisco Javier Melero, Manuel Chiachío, Juan Chiachío, Deep Generative Models for Damage Assessment in Digital Twins, Keywords: Digital Twin, Structural Health Monitoring, Damage Assessment, Conditional Wasserstein Generative Adversarial Networks with Gradient Penalty, Convolutional Neural Networks, Petri Nets

A.2 Open access code

A fundamental Python implementation of the CWGAN-GP generative model and the four predictive models for the damage assessment strategy is available on GitHub. It can be accessed via the following link: `https://github.com/mmmmaria/Digital-twin.gitk`.

A.3 Other contributions

Technical seminars as invited speaker

- María Megía. Research Communication in the ENHAnCE H2020 MSCA-ITN project. Universidad de Granada, Spain (UGR). Training Week 1, ENHAnCE. October, 2020.
- María Megía. The digital twin explained. Noche Europea de los Investigadores. Universidad de Granada, Spain (UGR). September, 2022.
- María Megía. Digital twin technology in structural engineering. Universidad de Granada, Spain (UGR). Training Week 9, ENHAnCE. May, 2023.

Appendix B

Technological implementation details

This Appendix presents the technological framework used to monitor the case studies in this thesis. Developed by the iPMLab[1] at the University of Granada, this framework was created during the preparation of the thesis. The author contributed to the development of the platform and its analytical modules, including programming, content creation, and testing.

B.1 Description

To support the implementation of the DT in the case studies, a technological integration has been developed to enable machine-to-machine (M2M) communication, from the **smart devices** (incorporating sensors and actuators with microcontrollers and/or microprocessors) to the **integration platform**, through the **network connectivity** (encompassing the physical media -wired or wireless, gateways, communication channels, and transmission protocols and standards), as depicted in Figure B.1. All the computational developments, software and frameworks employed in this thesis are open source and compatible with Python.

This technological deployment, referred as *the system*, works as a proper structural monitoring system, being able to collect in real time, through various secure protocols (subscription, publication, and request), a vast array of measurement parameters from a wide range of sensors, regardless of their type. These sensors are linked to devices which process this information and store it relationally, integrating it into a data model that supports subsequent predictive analyses.

The system is equipped with the ability to autonomously recognise and connect devices and sensors that engage in real-time communication. It features a secure web management interface for the remote reconfiguration of devices through secure encrypted messages. Additionally, it supports straightforward device relocation and replacement and allows users to add annotations. This system also offers a user-friendly, easily accessible web interface that monitors devices and sensors linked to various assets in real

[1] https://ipmlab.ugr.es/

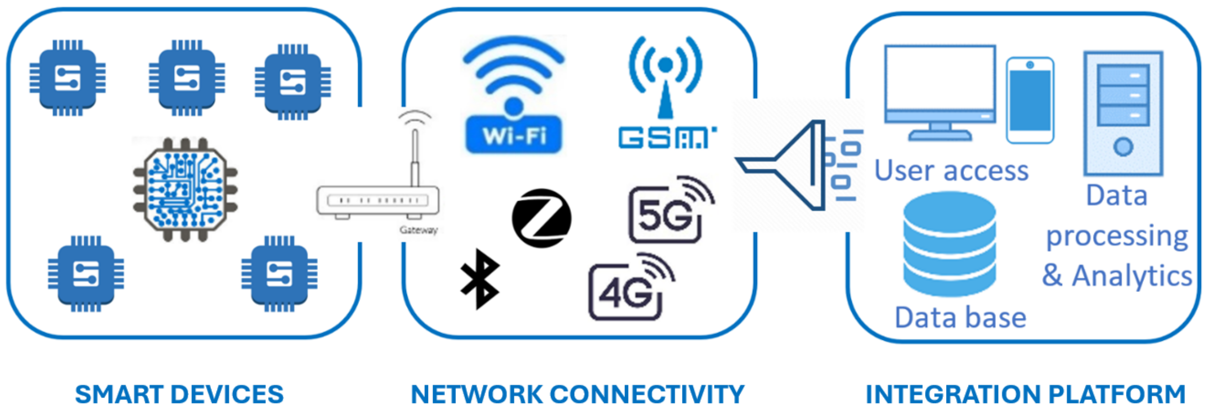


Figure B.1: Components of the DT technological implementation

time, enabling users to review and export related historical data and detect anomalies over time.

The key features of the system are detailed below:

- Universal Sensor Detection System: Leveraging serialised information transmission, the system can identify a large variety of sensors such as ultrasound, temperature, humidity, distance, etc. This feature enables the integration of external devices that adhere to common communication standards.
- Automated Device and Sensor Identification: The central node, using serialised data from devices, can securely identify newly added devices to the communication channel without requiring end-user interaction.
- Web-Based Device, Sensor, and Physical Asset Management Interface: This interface comes with a comprehensive CRUD (Create, Read, Update, Delete) system, enhanced with search capabilities and filters for efficient management. It also includes an authentication system and a firewall that can be controlled by the system's administrator.
- Real-Time Sensor Monitoring Graphical Interface: This feature allows for the real-time monitoring of various physical assets, displaying the locations of devices and sensors on a map. It offers managers and administrators the ability to handle different layouts and reposition devices through drag-and-drop functionality.
- Individual Device Monitoring System: This system allows for individual access to devices to review and export sensor history and graphically view recent measurements.
- Taxonomy-Based Annotation System: Utilizing a taxonomy-based structure, users can easily add private notes on assets, devices, and sensors. System administrators can also make global annotations visible to all users.
- Data Backup System: Designed to prevent information loss caused by improper application use by managers, this system ensures the recovery of data related to removed devices or sensors.

B.1.1 Smart devices

The IoT smart devices employed in this research consist of various components varying depending on their functionality:

- SoC (System on a Chip): The SoC of the present deployment consists of microcontrollers, which are integrated components capable of executing simple programmed commands, such as taking sensor measurements or executing communication actions with the platform. Programming these microcontrollers involved languages like MicroPython, resulting in firmware that must be flashed onto each device. The type of microcontrollers used were ESP8266 and ESP32 due to their flexible connectivity options.
- Sensors: Allow the detection of actions or events. They require one or several analogue or digital connections, or in more complex cases, the use of a digital communication BUS (Binary Unit System) like I2C (Inter−Integrated Circuit). For an analogue sensor, it will be necessary to map the voltage (V) from the reading of the internal Analogue-to-Digital Converter (ADC) of the microcontroller, with: $V = (ADC \cdot V_{Max})/ADC_{Max}$. An ADC works by sampling an analogue signal and converting it into a digital number based on the signal's voltage level relative to its maximum measurable voltage (V_{Max}). There are many types of sensors, including temperature, force resistive, accelerometers, distance meters, load cell transmitters, etc.
- Actuators: Enable the execution of actions or events. Like sensors, they may require several analog or digital connections, or in complex cases, the use of a digital communications BUS. Some of the most commonly utilised actuators are relays, displays, servo motors, variators, and more.
- Additional Modules: refer to those electronic structures that add functionality to an electronic system, which are not classified as sensors or actuators. These components communicate similarly to sensors and actuators. Among the most frequently used modules are GSM connectivity modules, GPS modules, multiplexers, and others.
- Other components or devices: To categorise the remaining electronic components, this classification encompasses the devices or electronic parts essential for a system. Included in this category are resistors, diodes, capacitors, connectors, batteries, transistors, cables, and also external devices requiring activation, such as sirens or electric motors.

The present IoT device development was based on Mosquitto MQTT subscription standards for monitoring and remote action on physical assets. This firmware is elaborated using the Arduino IDE in C++ language and allows the interconnection of the smart devices for sending sensor information or remote action in response to scheduled events. The IoT services consist of SoC sensors and actuators with a connectivity usually employing Wi-Fi connection, but also capable of using mobile technology. The microcontrollers employed in this development were ESP8266 and ESP32, as shown in Figure B.2.

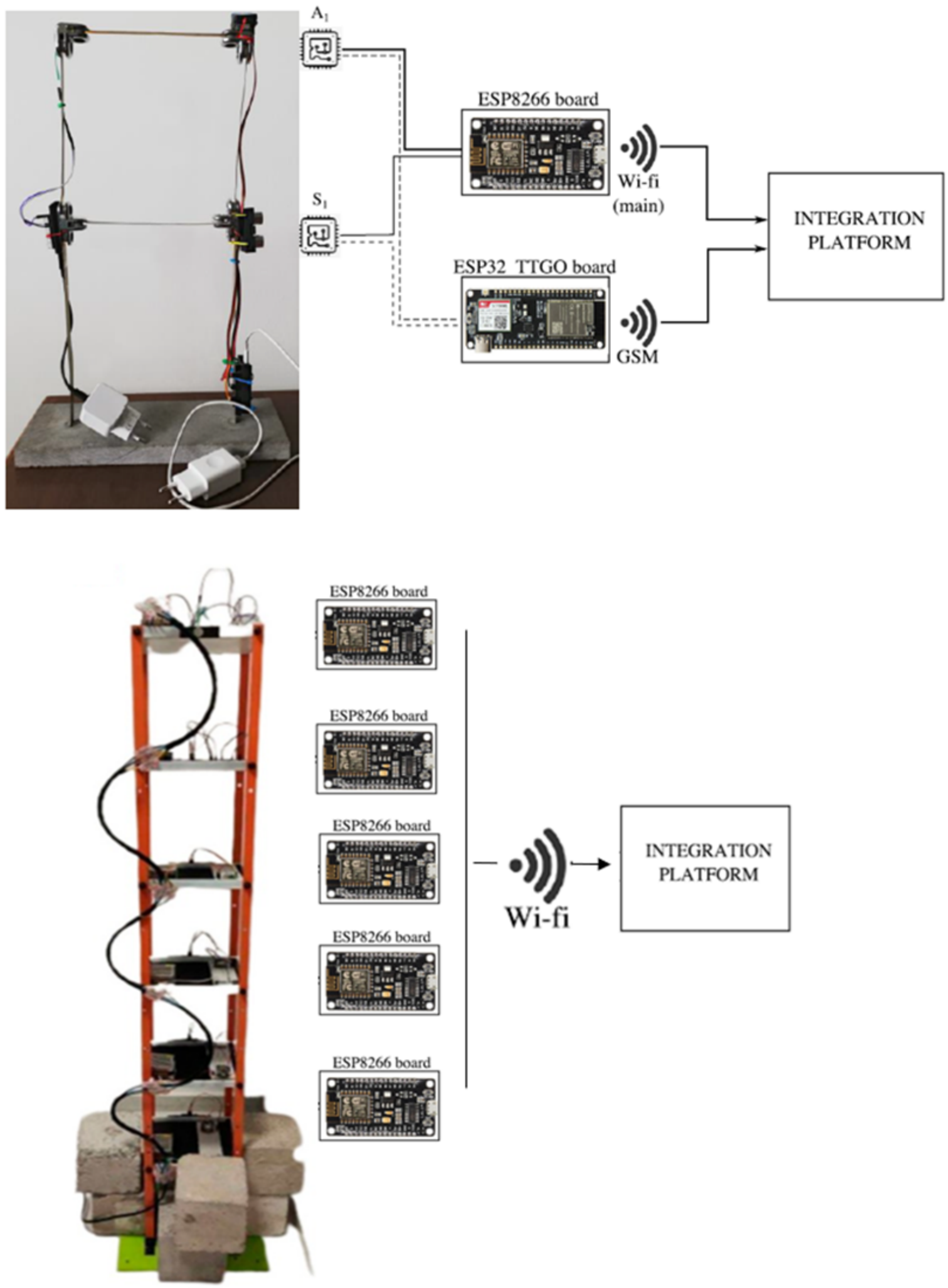


Figure B.2: Smart devices connected to the integration platform in the case studies: Case study 1 (2D tower) above and Case study 2 (3D tower) bellow.

B.1.2 Network connectivity

The node for connectivity was developed in Python using the MQTT and HTTP protocols (Figure B.3), which allow the exchange of information through serialised data strings. This node is responsible for intelligently and automatically collecting and storing data obtained from various monitoring activities, as well as triggering events defined by users. It features various monitoring modes, the capability for remote reconfiguration, and the option to use alternative communication channels, such as HTTP requests or SMS messaging, with 2.4GHz Wi-Fi and GSM 3G/4G wireless technologies.

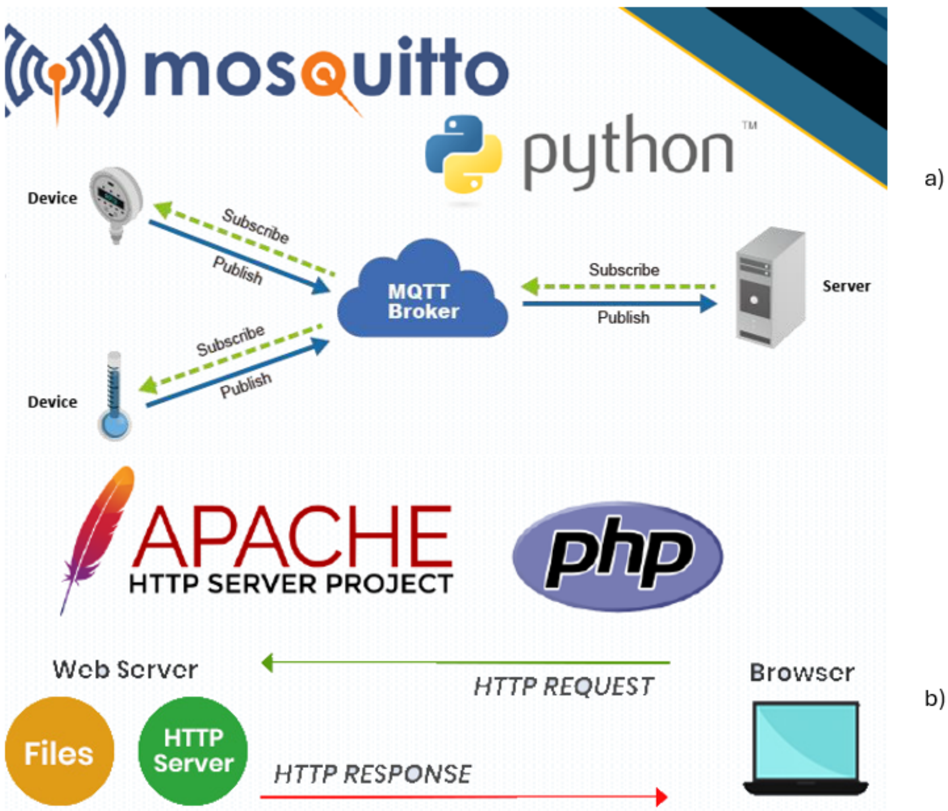


Figure B.3: Communication channels: a) MQTT and b) HTTP

In the presented development, communications use serialised JSON strings that include identifiers for devices, sensors and actuators, along with measurement data to be linked to the platform. In this way, each device or the platform itself can determine what action is requested in each communication. Each action facilitated through MQTT or HTTPS serves a specific purpose in managing the IoT ecosystem, from real-time monitoring and control of devices to configuration management and data analysis.

The MQTT communication protocol works similarly to a conversation, with the service's broker facilitating the communication channel utilised by both the devices and the platform. In this application, it establishes a two-way communication link between the devices and the platform. The actions related to MQTT are the following:

- Sensor Data Collection (iotnexus/sensor topic): Devices send sensor data such as temperature, humidity, motion detection, etc., to the platform at defined intervals or when changes exceed a predefined threshold.
- Actuator Commands (iotnexus/actuator topic): The platform sends commands to devices to perform specific actions, like turning on/off a light, adjusting a thermostat, or activating a motor.
- Measurement Requests (iotnexus/request topic): The platform requests the current reading from a specific sensor on a device, like requesting the current temperature from a temperature sensor.
- Configuration Updates (iotnexus/config topic): The platform sends configuration updates to devices, such as changing the frequency of sensor data reporting or updating firmware.

HTTP employs requests to manage different responses from clients (users or devices). In the present application, it is a unidirectional channel in terms of communication with devices. Actions for HTTP are included below:

- Device Registration/Management: Devices or users can register new devices to the platform, update device information, or deregister devices.
- Event Registration: Devices or backend services can post events or alerts to the platform, like notifying of a device malfunction or security breach.
- Data Retrieval: Authorised users or services can make requests to retrieve stored sensor data, device statuses, or historical event logs for analysis or reporting.
- User Authentication and Authorisation: Handling login requests, validating user credentials, and ensuring users have the necessary permissions to perform requested actions.
- External API Integration: Making requests to third-party services for additional data processing, storage, or triggering external workflows based on sensor data or device events.

B.1.3 Platform

As an interface between the virtual and the physical world, a web-based platform was deployed for managing devices and presenting the monitored data. This is a low-requirement web interface developed in PHP scripting language that gives users the ability to view and export data obtained from monitoring physical assets, as well as remotely reconfigure sensors and define events.

This platform was built from the ground up over the open-source operating system Ubuntu and a variety of essential open-source services and languages for enabling the communication channels:

- Apache HTTP Server: A software service/server aimed at facilitating external access to handle requests (HTTP/HTTPS) and to serve web documents.
- NTP Server: Provides time synchronisation services for connected devices.
- MariaDB: A Database Management System (DBMS) essential for data management and storage.
- MQTT Broker: Offers a two-way communication channel leveraging subscription and publishing technologies within threads.

- PHP: Facilitates the processing of backend code to generate dynamic web content.
- Python: An easily readable, high-level, interpreted, multi-platform programming language.

The platform was hosted on a physical cloud server from the University of Granada and supported by a virtual environment Proxmox VE, an open-source virtualisation management framework. Proxmox provides support for virtualisation technologies like KVM (Kernel-based Virtual Machine) for full virtualisation, making it a flexible solution for running multiple operating systems and applications on a single physical server. Proxmox virtual machine is also responsible for providing essential communication services through ports: 51183 (MQTT) and 51443 (HTTPS), the latter being redirected via ProxyPass through port 443 under the URL https://ipmlab.ugr.es/iot/.

B.2 Design

The design of the system incorporates both Object-Oriented Programming (OOP) and procedural programming paradigms (PPP). While OOP is utilised across all development aspects, devices and modules operate under a procedural programming paradigm. Additionally, the software architecture follows the Model-View-Controller (MVC) pattern (represented in Figure B.4), segregating the application's code into distinct layers based on their functions and responsibilities. Although the web architecture aligns with this pattern, Python services deviate from strict adherence due to the absence of direct client interaction. Instead, events are triggered by the MQTT service or the Petri net, leading to the exclusion of a view component for services lacking client engagement.

The design of the database (Figure B.5) was intended to be as versatile as possible while keeping as a relational database. A taxonomy system based on ISA subentities was introduced in order to link the annotation system with the device, the sensor, and the physical asset tables. This approach allowed descendant tables of taxonomy to inherit the primary key from their ancestor and link to the annotation system using a single method. As commented below, a device can have none, one, or many sensors or actuators. Furthermore, these tables are connected to their respective log tables ('SensorLog' and 'ActuatorLog'), designed to capture the most relevant data from measurements/actions along with device-specific information that requires tracking, such as its location.

For the graphical user interface, platform management, and mapping devices spatially, the tables 'Authorised', 'API', 'PhysicalAssetFile', and 'PhysicalAsset' are necessary. The 'PhysicalAsset table', in particular, is responsible for storing all information about the physical asset and linking it to the DT's historical data through the 'BayesianModule' table, built specially for Case Study 1 (Chapter 11).

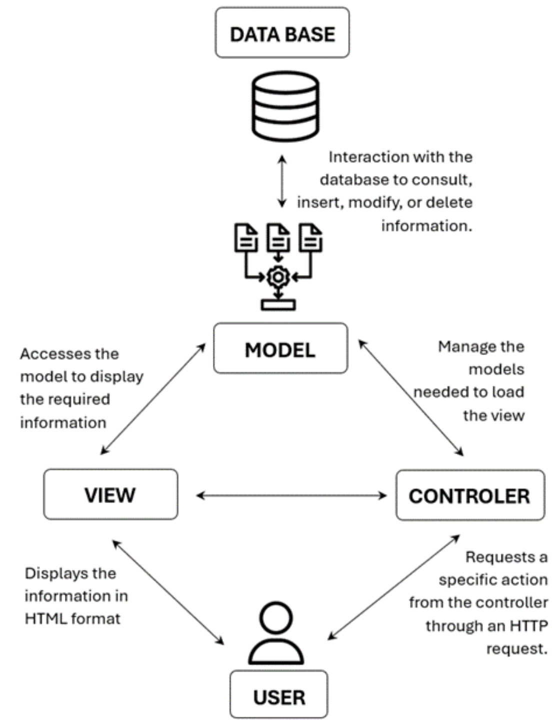


Figure B.4: Model-View-Controller (MVC) scheme

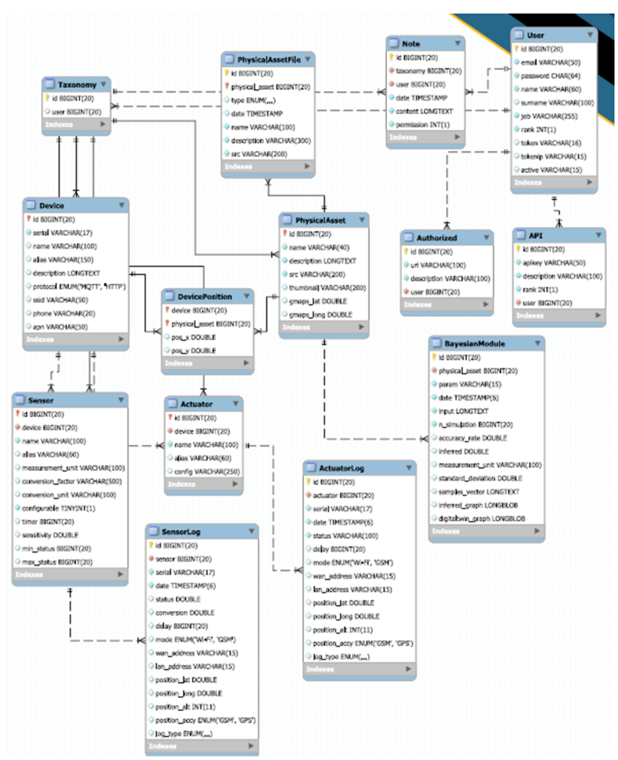


Figure B.5: Database map overview

B.3 Operation

The system operates by integrating various components and managing events across different channels. The 'Monitoring Kernel' component facilitates the management of events received via the MQTT channel, while the 'Workflow Model' component, although still in early development, enables the parallel execution of different Petri nets.

These Petri nets can interact with data models from the monitoring system, handle device requests, or automate the execution of necessary modules. Additionally, the RESTful API and the web platform allow for event and request management through HTTP communication, catering to user viewing, inquiries, or device integration needs. Finally, the database serves as the backbone for storing and retrieving information, allowing models to query, insert, modify, or delete information through the integration of the language connector with the database management system (DBMS).

The system operates with two types of events: web events and service events. Web events are events triggered through an HTTP request and subsequently, they can be classified into:

- User events: Refer to those events triggered from the web platform, either for consultation or management.
- API events: Refer to events requested directly from the controller. In this application, these events receive a response in serialised JSON format. This type of event would cover both consultation and integration events via the HTTP channel.

Service events are managed automatically, whether triggered by automated behaviour or a decision system. They can also be classified as:

- Python monitoring events: An event managed by an infinite loop that is responsible for managing all the sensor/actuator information received through MQTT.
- Reconfiguration events: If the configuration of the devices does not match what is considered on the platform, their reconfiguration will be requested via MQTT.
- Decision events (Petri net): Interaction through auxiliary procedures (helpers) that manage a Petri net automatically. These events need to be defined explicitly in the code.

B.4 Access

The platform, by default, is only accessible from the University's local network or by using its VPN through the following link: https://ipmlab.ugr.es/iot/iot/ . The platform requires users to be identified at all times, for security reasons. Once logged into the platform, by default, the user is redirected to the controller view: https://ipmlab.ugr.es/iot/monitoring/monitoring/ , where a list of all previously registered physical assets is displayed along with their thumbnail view (Figure B.6).

Figure B.6: Assets registered in the platform: 2D Tower (Case study 1 in Chapter 11) and 3D Tower (Case study 2 in chapter 12)

Clicking on any of them will provide a detailed individual view of each asset. Each detailed view is automatically constructed based on the information collected by the platform. It includes three main tabs:

- 'Monitoring' view: This view displays the status of devices, sensors, and actuators in real-time, with an update frequency of 2 seconds due to client limitations (Figures B.7 for the 2D tower and Figure B.8 for the 3D tower). Clicking on any of the devices will take to its detailed view. It also includes a feature to export historical data in a graph, organising the logs according to the type of event (either automatic or manually initiated). It is important to note that conversions can be applied to the values to transform the outcomes. This approach is especially convenient for ADC-based measurements that necessitate precise calibration. Consequently, the historical data will have an additional column that displays the conversion carried out by the platform as well as the true reading value from the ADC.
- 'Digital twin' view: Shows the state of the 'Bayesian inference' module execution built for Case Study 1 (Figure B.9), with a query frequency of 2 seconds, providing a controlled history of the last 10 states, in addition to offering a system for downloading the simulation vector used. This tab is optional and will only be available if there is relevant information in this regard. The execution of the 'Bayesian module' must be done automatically, based on Python programming, either through a scheduled task or execution commanded by Petri nets.

- ‘Resources and reports’ view: allows for accessing and downloading files related to maintenance and research, which have been uploaded by users based on their access rights. This feature is optional and becomes available only when pertinent information exists in this context.

In addition to the tabs mentioned, the user will also be provided with the ability to register and consult individual and global annotations, as well as an interactive map to learn more about the location of the physical asset.

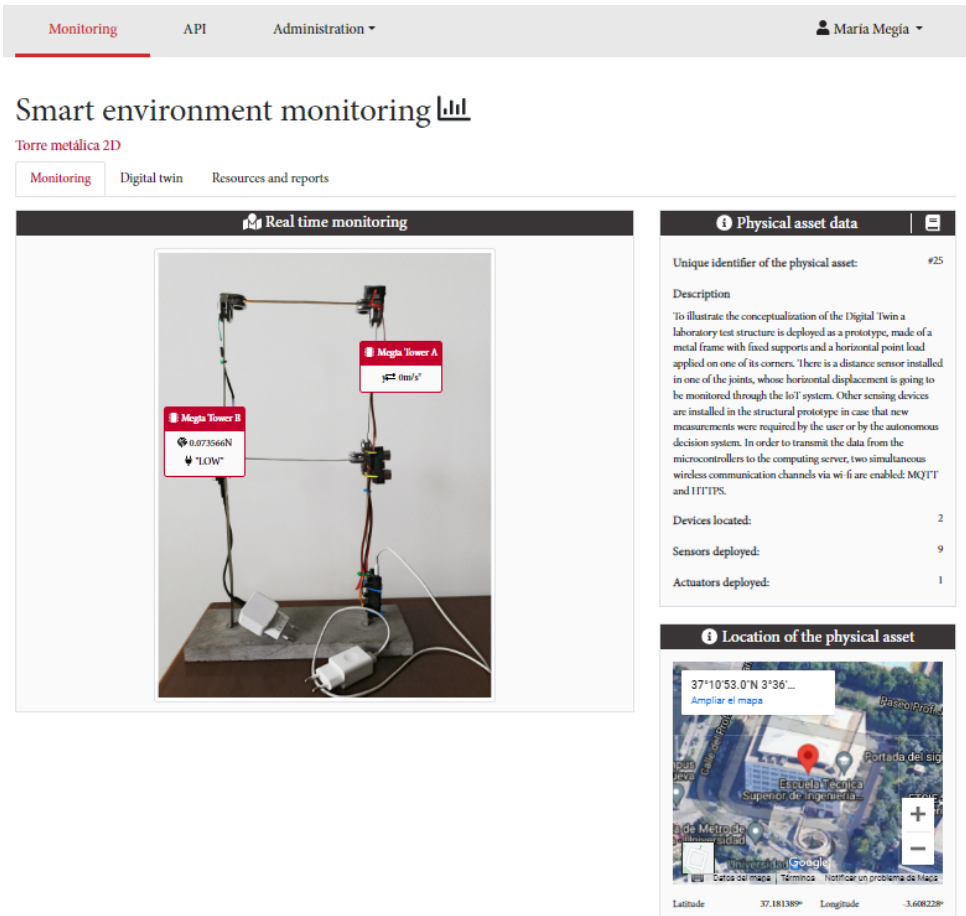


Figure B.7: View of ‘Monitoring’: devices, sensors, and actuators connected in the 2D Tower

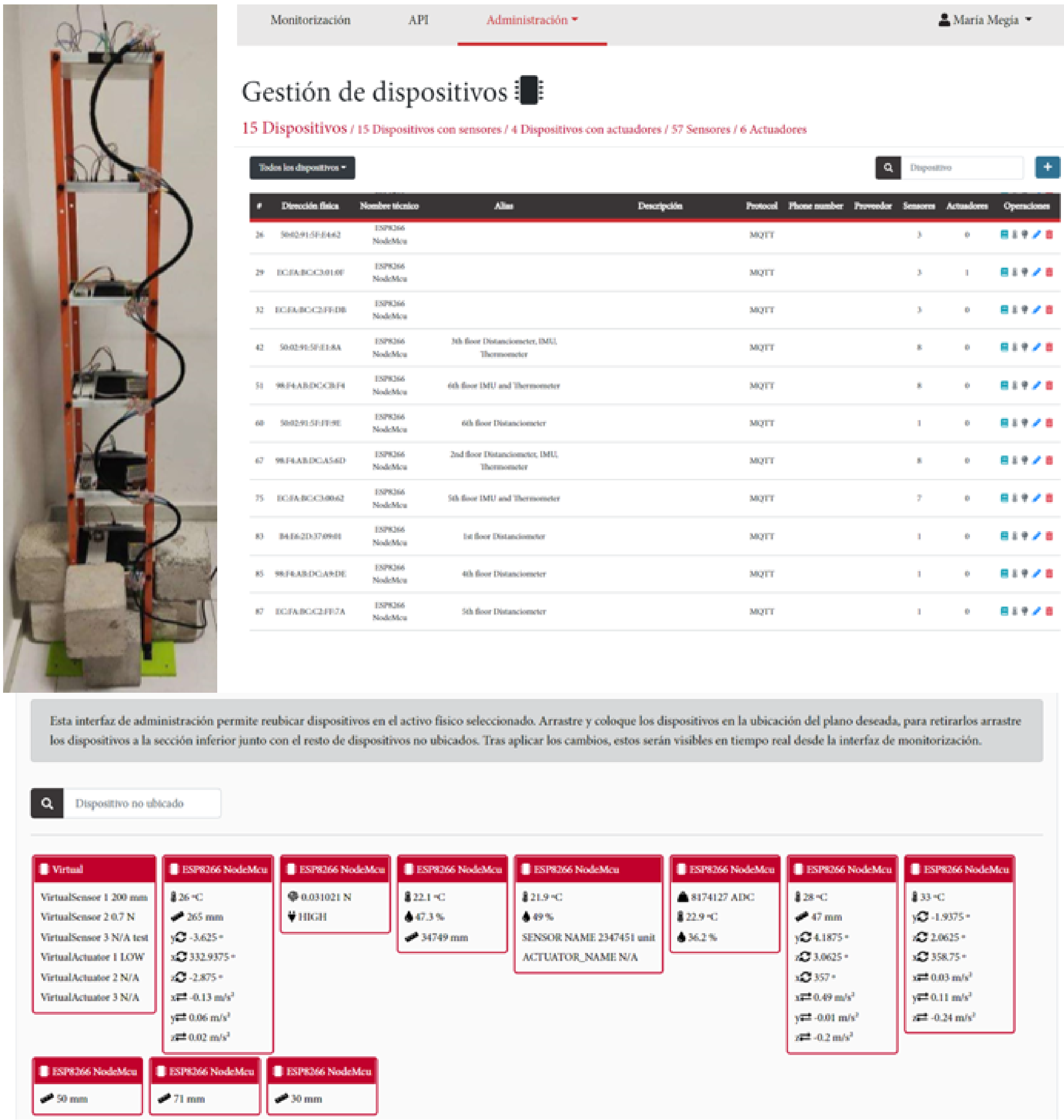


Figure B.8: View of 'Monitoring': devices, sensors, and actuators connected in the 3D Tower

Figure B.9: View of ‘Digital Twin’ with the ‘Bayesian module’ in the 2D Tower

B.5 Administration

A relevant operation that can be performed in the platform is the management of the physical assets and their devices. Within the views associated with this menu, it is possible to register new assets, manage them, locate devices based on the map, or store related files such as research or maintenance reports. Access to the CRUD (Create, Read, Update, and Delete user interface) view is available through a menu: Administration / Physical Asset Management. In this view, there is a list of assets on which various actions can be performed.

It is possible to 'add', 'edit', or 'delete' assets, storing information regarding the name, description, and image to act as a map (where devices will be placed for monitoring), thumbnail image (to be displayed in the index), latitude, and longitude (which can be obtained from Google Maps by right-clicking and copying the coordinates). Each device uses its MAC address (a unique physical network card address for each device) as an identifier.

Clicking on the 'file management' button opens a new CRUD view, where files can be added, modified, or deleted through a pop-up window. File names are generated automatically to prevent conflicts, though their title and description can be customised to enhance accessibility and usability for the user.

Clicking on the 'device location' button opens a drag-and-drop interface view, where devices can be moved to the desired location on the map. Changes saved here will update in real time for all users in the monitoring view.

In addition, clicking on 'annotation management' opens the annotation pop-up window, which can also be accessed from the asset monitoring query view.

The attributes that can be managed in a sensor are the following:

- Measurement unit: Defines the base unit of measurement in which the sensor sends data.
- Transmission interval: Sets an interval in milliseconds after which measurements must be taken, allowing to ensure that measurements are taken at regular intervals.
- Sensor sensitivity: Refers to the sensitivity threshold of the sensor, both above and below, which will serve as a reference for future measurements. If this is not exceeded, no additional measurements will be taken.
- Minimum/Maximum anomalous state: The minimum/maximum value that defines an alert in the monitoring process. If exceeded, the record of this state will be marked in the log and administrators will be notified.
- Conversion factor: Refers to the mathematical formula from which a measurement value can be recalculated, for example, changing from an ADC reading to a strain measurement.
- Conversion unit: Refers to the measurement unit in which the sensor's state will be represented after the conversion process.

For actuators, there is only one attribute:

- Advanced actuator configuration: Refers to a JSON serialised string that can be utilised for multiple purposes regarding its remote reconfiguration. Considering the diversity of actuator types, employing serialised object notation is anticipated to offer greater flexibility, even though this feature is presently inactive.

B.6 External communications via API

To enable communication between this platform and any external application, whether for inserting/integrating data or querying data managed by the platform, the API can be used. For insights into its functionality and guidelines on interaction, information is available in the API section of the platform via the main menu link.

For the administration of tokens, which enable different levels of interaction, it is necessary to navigate to the appropriate CRUD section through the menu under: Administration / API Management. Here, an initial overview presents a list of assets, alongside which various actions can be performed through related buttons:

- Buttons for adding, editing, or deleting API tokens: These are designed for creating or adjusting API access permissions. Upon clicking any of these buttons, a popup form appears, detailing the following: API Key (self-generated token), its description, and the access level of the API. Communication with the application through this API Key requires the HTTPS protocol, and access to specific features depends on the token type set by the administrator.
- Integration API: This feature enables the insertion of data into the platform, allowing sensors or scripts to contribute valuable information to the platform's database, adhering to the outlined instructions.
- Query API: This function allows for the extraction of data from the platform, thereby facilitating its use as an information repository for external websites or applications.

  For security measures, this API functions via the POST method, mandatorily requiring an *api_key* field along with a data string for operation, regardless of the mode in use.

B.7 Case Study 1: 2D Tower

B.7.1 Methodology

The test conducted for the first case study (Figure B.10) consisted of multiple sessions in which forces were applied to the tower at point 3, while measuring the displacement at point 5. The system triggers the 'Bayesian module' according to the policy configured in the Petri net, inferring the force which provoked the displacement. The data generated was stored in the database at the same time that actions were performed by the actuator according to the Petri net policy.

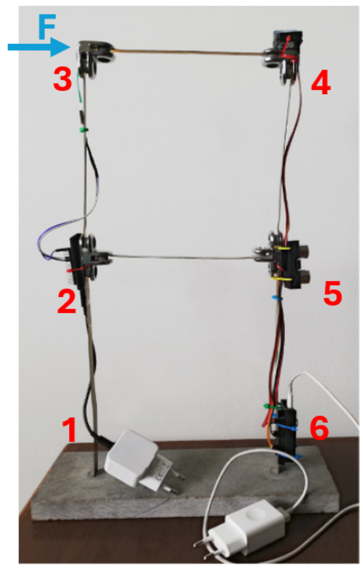


Figure B.10: 2D tower with Force representation

B.7.2 Data

The data involved in the case study is delineated as follows.

u: input parameters (measured displacement d, number of samples or simulations N)

θ: model parameters (stiffness coefficient K)

e: external variables: (force F, unknown and inferred)

B.7.3 Output records

In the following report (Figure B.11) it can be seen the data recorded: Execution date, Parameter inferred (in this case the force F), Values given to the system or Input parameters (the threshold displacement, the displacement measured, and the stiffness of the tower), the Inferred value by the module, the Accuate rate as the α parameter in the M-H algorithm (see Section 7.2), and the standard deviation involved in the inverse problem.

Bayesian historical

Execution date	Parameter	Input parameters	Inferred value	Accurate rate	Standard deviation	Actions
2023-09-29 17:44:46.014229	F	base dist = 1.10000000e-1 m disp = 5.60000000e-2 m stiffness = 9.00000000e+8 N/m^2	2.31276316e+ N	0.326	2.48593554e-1 N	
2023-09-29 17:42:50.856564	F	base dist = 1.10000000e-1 m disp = 7.97000000e-1 m stiffness = 9.00000000e+8 N/m^2	4.99912250e+ N	1	1.44363353e+ N	
2023-09-29 14:21:19.601053	F	base dist = 1.10000000e-1 m disp = 1.00000000e-3 m stiffness = 9.00000000e+8 N/m^2	1.54518864e-1 N	0.242	2.41525563e-1 N	
2023-09-29 14:19:46.416307	F	base dist = 1.10000000e-1 m disp = 7.20000000e-2 m stiffness = 9.00000000e+8 N/m^2	2.99734612e+ N	0.288	2.29698883e-1 N	
2023-09-27 11:02:07.718923	F	base dist = 1.10000000e-1 m disp = 2.08500000e+0 m stiffness = 9.00000000e+8 N/m^2	4.97245977e+ N	1	1.13131225e+ N	
2023-09-27 10:53:45.932141	F	base dist = 1.10000000e-1 m disp = 2.60000000e-2 m stiffness = 9.00000000e+8 N/m^2	1.15040962e+ N	0.33	1.35101250e-1 N	
2023-09-27 10:52:24.970289	F	base dist = 1.10000000e-1 m disp = 2.08100000e+0 m stiffness = 9.00000000e+8 N/m^2	4.99834035e+ N	1	1.16165380e+ N	
2023-02-16 09:15:10.630047	F	base dist = 1.10000000e-1 m disp = 3.40000000e-2 m stiffness = 9.00000000e+8 N/m^2	1.46823882e+ N	0.322	4.66417887e-1 N	
2023-02-16 09:14:35.720841	F	base dist = 1.10000000e-1 m disp = 6.90000000e-2 m stiffness = 9.00000000e+8 N/m^2	2.87441719e+ N	0.3	2.18449205e-1 N	
2023-02-15 09:28:50.474246	F	base dist = 1.10000000e-1 m disp = 4.00000000e-2 m stiffness = 9.00000000e+8 N/m^2	1.66358843e+ N	0.3	2.48803442e-1 N	

Here are shown the last 10 calculations of the Bayesian module. Please download the historical records to get more.

Figure B.11: Example of a report for the 2D tower

B.8 Case Study 2: 3D Tower

B.8.1 Methodology

The test conducted for the second case study (Figure B.12) consisted of a battery of forces applied to the tower in the middle point of its sixth floor, while measuring the displacement in the opposite middle point of the six floors. The system triggers the 'Damage Assessment module' according to the policy configured in the Petri net, evaluating the damage detection, location, quantification and RUL prediction if damage is encountered. The data generated was stored in the database at the same time that actions were performed by the actuator according to the Petri net policy.

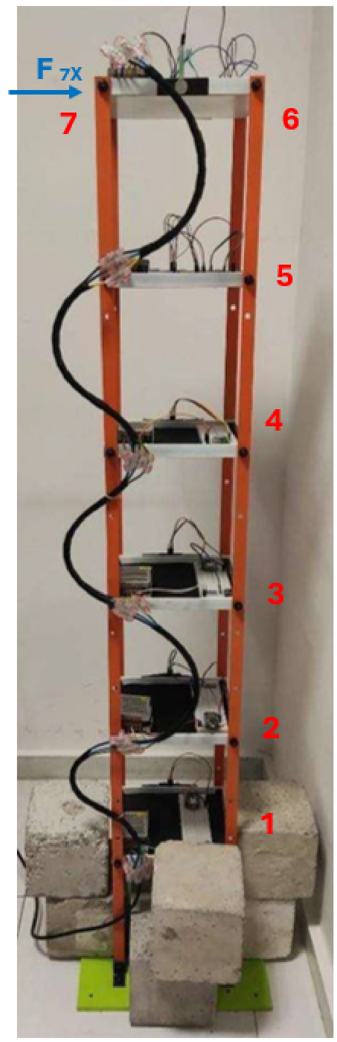


Figure B.12: 3D tower with Force representation

B.8.2 Data

The data involved in the present case study is as follows.

u: input parameters (measured displacement d)

θ: model parameters (stiffness coefficient K)

e: external variables: (force F)

B.8.3 Output records

iPMLab - Structural monitoring platform

Device report #60 (50:02:91:5F:FF:9E) for sensor #61 (6th floor Distanciometer) in date range 09-06-2022 - 09-06-2022

#	Log date	Device	Sensor	Value	Measurement unit	Conversion value	Conversion unit	Transmission delay
663621	2022-06-09 10:11:51.877108	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	743ms
663629	2022-06-09 10:12:00.967001	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	728ms
663633	2022-06-09 10:12:02.227010	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	948ms
663634	2022-06-09 10:12:03.633210	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	317ms
663644	2022-06-09 10:12:06.128714	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	739ms
663646	2022-06-09 10:12:07.209576	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	742ms
663647	2022-06-09 10:12:08.248697	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	733ms
663649	2022-06-09 10:12:10.303333	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	742ms
663653	2022-06-09 10:12:12.350034	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	740ms
663659	2022-06-09 10:12:14.397781	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	743ms
663664	2022-06-09 10:12:16.442409	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	740ms
663669	2022-06-09 10:12:18.489583	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	741ms
663671	2022-06-09 10:12:20.535424	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	732ms
663681	2022-06-09 10:12:25.619609	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	738ms
663685	2022-06-09 10:12:26.705282	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	743ms
663703	2022-06-09 10:12:34.789648	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	742ms
663705	2022-06-09 10:12:36.829155	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	743ms
663709	2022-06-09 10:12:38.888357	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	743ms
663712	2022-06-09 10:12:39.969106	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	740ms
663716	2022-06-09 10:12:40.102940	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	835ms
663719	2022-06-09 10:12:41.502750	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	195ms
663725	2022-06-09 10:12:49.185267	50:02:91:5F:FF:9E	6th floor Distanciometer	395	mm	--	--	746ms
663726	2022-06-09 10:12:50.221460	50:02:91:5F:FF:9E	6th floor Distanciometer	390	mm	--	--	744ms
663770	2022-06-09 10:13:23.520460	50:02:91:5F:FF:9E	6th floor Distanciometer	386	mm	--	--	739ms
663774	2022-06-09 10:13:24.605347	50:02:91:5F:FF:9E	6th floor Distanciometer	383	mm	--	--	738ms
663783	2022-06-09 10:13:26.656408	50:02:91:5F:FF:9E	6th floor Distanciometer	399	mm	--	--	741ms
663790	2022-06-09 10:13:28.705873	50:02:91:5F:FF:9E	6th floor Distanciometer	393	mm	--	--	734ms

Figure B.13: Example of a report for the displacement in the 6th floor in the 3D tower

Part VI

References

Bibliography

[1] Klaas Van Breugel. Societal burden and engineering challenges of ageing infrastructure. *Procedia Engineering*, 171:53–63, 2017.

[2] Mauricio Sánchez-Silva, Dan M Frangopol, Jamie Padgett, and Mohamed Soliman. Maintenance and operation of infrastructure systems. *Journal of Structural Engineering*, 142(9):F4016004, 2016.

[3] Co2 emissions from buildings and construction hit new high, leaving sector off track to decarbonize by 2050. Technical report, UN Environmental Programme, 2022.

[4] B Danette Allen. Digital twins and living models at nasa. In *Digital Twin Summit*, 2021.

[5] Michael Grieves and John Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary perspectives on complex systems: New findings and approaches*, pages 85–113, 2017.

[6] Eric J Tuegel, Anthony R Ingraffea, Thomas G Eason, S Michael Spottswood, et al. Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering*, 2011, 2011.

[7] Edward Glaessgen and David Stargel. The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, page 1818, 2012.

[8] Ulf T Tygesen, Michael S Jepsen, Jonas Vestermark, Niels Dollerup, and Anne Pedersen. The true digital twin concept for fatigue re-assessment of marine structures. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 51203, page V001T01A021. American Society of Mechanical Engineers, 2018.

[9] Peter ED Love and Jane Matthews. The 'how'of benefits management for digital technology: From engineering to asset management. *Automation in Construction*, 107:102930, 2019.

[10] Frank Schnicke, Thomas Kuhn, and Pablo Oliveira Antonino. Enabling industry 4.0 service-oriented architecture through digital twins. In *Software Architecture: 14th European Conference, ECSA*

2020 Tracks and Workshops, L'Aquila, Italy, September 14–18, 2020, Proceedings 14, pages 490–503. Springer, 2020.

[11] Laura Pellegrini, Stefano Campi, Mirko Locatelli, Giulia Pattini, Giuseppe Martino Di Giuda, and Lavinia Chiara Tagliabue. Digital transition and waste management in architecture, engineering, construction, and operations industry. *Frontiers in Energy Research*, 8:576462, 2020.

[12] Erwin Rauch and Eugenio Vinante. Three dimensional technology radar model to evaluate emerging industry 4.0 technologies. In *Design, Simulation, Manufacturing: The Innovation Exchange*, pages 233–242. Springer, 2021.

[13] Oray Kulac, Banu Y Ekren, and A Ozgur Toy. Digital twins for decision making in supply chains. In *Global Joint Conference on Industrial Engineering and Its Application Areas*, pages 86–96. Springer, 2022.

[14] Elisa Negri, Luca Fumagalli, and Marco Macchi. A review of the roles of digital twin in cps-based production systems. *Procedia manufacturing*, 11:939–948, 2017.

[15] Ahmet Anil Sezer, Micael Thunberg, and Brian Wernicke. Digitalization index: Developing a model for assessing the degree of digitalization of construction projects. *Journal of construction engineering and management*, 147(10):04021119, 2021.

[16] Christian U Grosse. Monitoring and inspection techniques supporting a digital twin concept in civil engineering. In *Proc. 5th International Conference on Sustainable Construction Materials and Technologies (SCMT5): In honour of Professor Christian Grosse*, 2019.

[17] C Arcadius Tokognon, Bin Gao, Gui Yun Tian, and Yan Yan. Structural health monitoring framework based on internet of things: A survey. *IEEE Internet of Things Journal*, 4(3):619–635, 2017.

[18] Cong Ye, Liam Butler, Bartek Calka, Marat Iangurazov, Qiuchen Lu, Alastair Gregory, Mark Girolami, and Campbell Middleton. A digital twin of bridges for structural health monitoring. 2019.

[19] General Assembly. Sustainable development goals. *SDGs Transform Our World*, 2030:6–28, 2015.

[20] Un general assembly, transforming our world the 2030 agenda for sustainable development. Technical report, UN General Assembly, 2015.

[21] Michael Grieves. Product lifecycle management: driving the next generation of lean thinking. *(No Title)*, 2006.

[22] DJ Wagg, Keith Worden, RJ Barthorpe, and Paul Gardner. Digital twins: state-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 6(3):030901, 2020.

[23] M Pregnolato, S Gunner, E Voyagaki, R De Risi, N Carhart, G Gavriel, P Tully, T Tryfonas, J Macdonald, and C Taylor. Towards civil engineering 4.0: Concept, workflow and application of digital twins for existing infrastructure. *Automation in Construction*, 141:104421, 2022.

[24] Davide D'Amico, John Ekoyuncu, Sri Addepalli, Christopher Smith, Ed Keedwell, Jim Sibson, and Steven Penver. Conceptual framework of a digital twin to evaluate the degradation status of complex engineering systems. *Procedia CIRP*, 86, 2019. pp: 61-67.

[25] Feng Jiang, Ling Ma, Tim Broyd, and Ke Chen. Digital twin and its implementations in the civil engineering sector. *Automation in Construction*, 130:103838, 2021.

[26] TG Ritto and FA Rochinha. Digital twin, physics-based model, and machine learning applied to damage detection in structures. *Mechanical Systems and Signal Processing*, 155:107614, 2021.

[27] NgocSon Dang, HwiRang Kang, Sokanya Lon, and ChangSu Shim. 3d digital twin models for bridge maintenance. In *Proceedings of 10th International Conference on Short and Medium Span Bridges, Quebec city, Quebec, Canada*, 2018.

[28] Ruodan Lu and Ioannis Brilakis. Digital twinning of existing reinforced concrete bridges from labelled point clusters. *Automation in construction*, 105:102837, 2019.

[29] CS Shim, HR Kang, and NS Dang. Digital twin models for maintenance of cable-supported bridges. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*, pages 737–742. ICE Publishing, 2019.

[30] Chang-Su Shim, Ngoc-Son Dang, Sokanya Lon, and Chi-Ho Jeon. Development of a bridge maintenance system for prestressed concrete bridges using 3d digital twin model. *Structure and Infrastructure Engineering*, 15(10):1319–1332, 2019.

[31] Cong Ye, Liam Butler, C Bartek, Marat Iangurazov, Qiuchen Lu, Alastair Gregory, Mark Girolami, and Campbell Middleton. A digital twin of bridges for structural health monitoring. In *12th International Workshop on Structural Health Monitoring 2019*. Stanford University, 2019.

[32] Chang-Su Shim, Ngoc-Son Dang, Sokanya Lon, and Chi-Ho Jeon. Development of a bridge maintenance system for prestressed concrete bridges using 3d digital twin model. *Structure and Infrastructure Engineering*, 15(10), 2019. pp: 1319-1332.

[33] Faham Tahmasebinia, Daniel Fogerty, Lang Oliver Wu, Zhichao Li, Saleh Sepasgozar, Kai Zhang, Samad Sepasgozar, and Fernando Alonso Marroquin. Numerical analysis of the creep and shrinkage experienced in the sydney opera house and the rise of digital twin as future monitoring technology. *Buildings*, 9(6):pp1–18, 2019.

[34] Chanakya Boddupalli, Ayan Sadhu, Ehsan Rezazadeh Azar, and Scott Pattyson. Improved visualization of infrastructure monitoring data using building information modeling. *Structure and Infrastructure Engineering*, 15(9), 2019. pp: 1247-1263.

[35] Zhansheng Liu, Wenyan Bai, Xiuli Du, Anshan Zhang, Zezhong Xing, and Antong Jiang. Digital twin-based safety evaluation of prestressed steel structure. *Advances in Civil Engineering*, 2020, 2020.

[36] Paul Gardner, Mattia Dal Borgo, Valentina Ruffini, Aidan J Hughes, Yichen Zhu, and David J Wagg. Towards the development of an operational digital twin. *Vibration*, 3(3):235–265, 2020.

[37] Grigor Angjeliu, Dario Coronelli, and Giuliana Cardani. Development of the simulation model for digital twin applications in historical masonry buildings: The integration between numerical and experimental reality. *Computers & Structures*, 238:pp:106282, 2020.

[38] Shi Ye, Xiangang Lai, Ivan Bartoli, and A Emin Aktan. Technology for condition and performance evaluation of highway bridges. *Journal of Civil Structural Health Monitoring*, 10(4), 2020. pp: 573-594.

[39] Qiuchen Lu, Xiang Xie, Ajith Kumar Parlikad, and Jennifer Mary Schooling. Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance. *Automation in Construction*, 118:pp:103277, 2020.

[40] Kaiqi Lin, You-Lin Xu, Xinzheng Lu, Zhongguo Guan, and Jianzhong Li. Digital twin-based collapse fragility assessment of a long-span cable-stayed bridge under strong earthquakes. *Automation in Construction*, 123:pp:103547, 2021.

[41] Sakdirat Kaewunruen, Jessada Sresakoolchai, Wentao Ma, and Olisa Phil-Ebosie. Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions. *Sustainability*, 13(4):pp:2051, 2021.

[42] Mengmeng Wang, Chengye Wang, Anna Hnydiuk-Stefan, Shizhe Feng, Incecik Atilla, and Zhixiong Li. Recent progress on reliability analysis of offshore wind turbine support structures considering digital twin solutions. *Ocean Engineering*, 232:pp:109168, 2021.

[43] Danhui Dan, Yufeng Ying, and Liangfu Ge. Digital twin system of bridges group based on machine vision fusion monitoring of bridge traffic load. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[44] Zhansheng Liu, Antong Jiang, Anshan Zhang, Zezhong Xing, and Xiuli Du. Intelligent prediction method for operation and maintenance safety of prestressed steel structure based on digital twin technology. *Advances in Civil Engineering*, 2021, 2021.

[45] Shuangpin Li, Zuqiang Liu, Min Zheng, and Yonghua Li. Research on intelligent monitoring system of hydraulic engineering based on digital twin. In *2021 7th International Conference on Hydraulic and Civil Engineering & Smart Water Conservancy and Intelligent Disaster Reduction Forum (ICHCE & SWIDR)*, pages 224–231. IEEE, 2021.

[46] Manuel Chiachío, María Megía, Juan Chiachío, Juan Fernandez, and María L Jalón. Structural digital twin framework: Formulation and technology integration. *Automation in Construction*, 140:104333, 2022.

[47] Mengmeng Wang, Shizhe Feng, Atilla Incecik, Grzegorz Królczyk, and Zhixiong Li. Structural fatigue life prediction considering model uncertainties through a novel digital twin-driven approach. *Computer Methods in Applied Mechanics and Engineering*, 391:pp: 114512, 2022.

[48] Eky Febrianto, Liam Butler, Mark Girolami, and Fehmi Cirak. Digital twinning of self-sensing structures using the statistical finite element method. *Data-Centric Engineering*, 3:e31, 2022.

[49] Amirhosein Shabani, Margarita Skamantzari, Sevasti Tapinaki, Andreas Georgopoulos, Vagelis Plevris, and Mahdi Kioumarsi. 3d simulation models for developing digital twins of heritage structures: challenges and strategies. *Procedia structural integrity*, 37:314–320, 2022.

[50] Mojtaba Mahmoodian, Farham Shahrivar, Sujeeva Setunge, and Sam Mazaheri. Development of digital twin for intelligent maintenance of civil infrastructure. *Sustainability*, 14(14):8664, 2022.

[51] Haidar Hosamo Hosamo and Mohsen Hosamo Hosamo. Digital twin technology for bridge maintenance using 3d laser scanning: A review. *Advances in Civil Engineering*, 2022, 2022.

[52] Norberto Rojas-Mercedes, Kalil Erazo, and Luigi Di Sarno. Seismic fragility curves for a concrete bridge using structural health monitoring and digital twins. *Earthquake and Structures*, 22(5):503–515, 2022.

[53] Didem Gürdür Broo, Miguel Bravo-Haro, and Jennifer Schooling. Design and implementation of a smart infrastructure digital twin. *Automation in construction*, 136:104171, 2022.

[54] Ashwin Agrawal, Martin Fischer, and Vishal Singh. Digital twin: From concept to practice. *Journal of Management in Engineering*, 38(3):06022001, 2022.

[55] Hua Zhao, Chengjun Tan, Eugene J OBrien, Bin Zhang, Nasim Uddin, and Hongjie Guo. Developing digital twins to characterize bridge behavior using measurements taken under random traffic. *Journal of Bridge Engineering*, 27(1):04021101, 2022.

[56] Xiaonan Lai, Xiwang He, Shuo Wang, Xiaobang Wang, Wei Sun, and Xueguan Song. Building a lightweight digital twin of a crane boom for structural safety monitoring based on a multifidelity surrogate model. *Journal of Mechanical Design*, 144(6):064502, 2022.

[57] Sheng Yu, Dongsheng Li, and Jinping Ou. Digital twin-based structure health hybrid monitoring and fatigue evaluation of orthotropic steel deck in cable-stayed bridge. *Structural control and health monitoring*, 29(8):e2976, 2022.

[58] Amaury Chabod. Digital twin for fatigue analysis. *Procedia Structural Integrity*, 38:382–392, 2022.

[59] Wynand JvdM Steyn and André Broekman. Development of a digital twin of a local road network: a case study. *Journal of Testing and Evaluation*, 50(6):2901–2915, 2022.

[60] Jiulin Li, Qingquan Li, Qingzhou Mao, and Hao Xu. Digital-twin-based high-precision assembly of a steel bridge tower. *Buildings*, 13(1):257, 2023.

[61] T Zinke, S Reymer, S Kosse, P Hagedorn, M König, F Wedel, S Schneider, S Marx, S Nieborowski, and S Windmann. Digital twins for bridges–concept of a modular digital twin based on the linked data approach. In *Life-Cycle of Structures and Infrastructure Systems*, pages 1745–1752. CRC Press, 2023.

[62] Yumei Ye, Qiang Yang, Jingang Zhang, Songhe Meng, Jun Wang, and Xia Tang. A reconfigurable dynamic bayesian network for digital twin modeling of structures with multiple damage modes. *Theoretical and Applied Mechanics Letters*, 13(4):100440, 2023.

[63] Daniele Sivori, Laura Ierimonti, Ilaria Venanzi, Filippo Ubertini, and Serena Cattari. An equivalent frame digital twin for the seismic monitoring of historic structures: A case study on the consoli palace in gubbio, italy. *Buildings*, 13(7):1840, 2023.

[64] Zeqiang Wang, Mingming Li, Zhansheng Liu, Majid Dezhkam, Yifeng Zhao, and Yang Hu. Continuous dynamic analysis method and case verification of cable structure based on digital twin. *Sustainability*, 15(22):16125, 2023.

[65] Yumei Ye, Qiang Yang, Jingang Zhang, Songhe Meng, and Jun Wang. A dynamic data driven reliability prognosis method for structural digital twin and experimental validation. *Reliability Engineering & System Safety*, 240:109543, 2023.

[66] Annalaura Vuoto, Marco Francesco Funari, and Paulo B Lourenço. On the use of the digital twin concept for the structural integrity protection of architectural heritage. *Infrastructures*, 8(5):86, 2023.

[67] Xiangxiong Kong and Ronny Garrett Hucks. Preserving our heritage: A photogrammetry-based digital twin framework for monitoring deteriorations of historic structures. *Automation in Construction*, 152:104928, 2023.

[68] Asseel Za'al Ode Al-Hijazeen, Muhammad Fawad, Michael Gerges, Kálmán Koris, and Marek Salamak. Implementation of digital twin and support vector machine in structural health monitoring of bridges. *Archives of Civil Engineering*, pages 31–47, 2023.

[69] Vanni Nicoletti, Riccardo Martini, Sandro Carbonari, and Fabrizio Gara. Operational modal analysis as a support for the development of digital twin models of bridges. *Infrastructures*, 8(2):24, 2023.

[70] Jinghai Xu, Xin Shu, Peng Qiao, Shanshan Li, and Jigang Xu. Developing a digital twin model for monitoring building structural health by combining a building information model and a real-scene 3d model. *Measurement*, 217:112955, 2023.

[71] Shuai Teng, Xuedi Chen, Gongfa Chen, and Li Cheng. Structural damage detection based on transfer learning strategy using digital twins of bridges. *Mechanical Systems and Signal Processing*, 191:110160, 2023.

[72] T Hielscher, S Khalil, N Virgona, and SA Hadigheh. A neural network based digital twin model for the structural health monitoring of reinforced concrete bridges. In *Structures*, volume 57, page 105248. Elsevier, 2023.

[73] Dimitrios Loverdos and Vasilis Sarhosis. Geometrical digital twins of masonry structures for documentation and structural assessment using machine learning. *Engineering Structures*, 275:115256, 2023.

[74] Tapas Tripura, Aarya Sheetal Desai, Sondipon Adhikari, and Souvik Chakraborty. Probabilistic machine learning based predictive and interpretable digital twin for dynamical systems. *Computers & Structures*, 281:107008, 2023.

[75] Longxuan Wang, Hongbo Liu, Zhihua Chen, Fan Zhang, and Liulu Guo. Combined digital twin and hierarchical deep learning approach for intelligent damage identification in cable dome structure. *Engineering Structures*, 274:115172, 2023.

[76] Lara Edington, Nikolaos Dervilis, Anis Ben Abdessalem, and David Wagg. A time-evolving digital twin tool for engineering dynamics applications. *Mechanical Systems and Signal Processing*, 188:109971, 2023.

[77] Zimo Zhu, Jian Zhang, Songye Zhu, and Jun Yang. Digital twin technology for wind turbine towers based on joint load–response estimation: A laboratory experimental study. *Applied Energy*, 352:121953, 2023.

[78] Fardad Mokhtari and Ali Imanpour. A digital twin-based framework for multi-element seismic hybrid simulation of structures. *Mechanical Systems and Signal Processing*, 186:109909, 2023.

[79] Bo Wang, Zengcong Li, Ziyu Xu, Zhiyong Sun, and Kuo Tian. Digital twin modeling for structural strength monitoring via transfer learning-based multi-source data fusion. *Mechanical Systems and Signal Processing*, 200:110625, 2023.

[80] Huaitao Shi, Zelong Song, Xiaotian Bai, Yunjian Hu, Tao Li, and Ke Zhang. A novel digital twin model for dynamical updating and real-time mapping of local defect extension in rolling bearings. *Mechanical Systems and Signal Processing*, 193:110255, 2023.

[81] Kaiqi Lin, You-Lin Xu, Xinzheng Lu, Zhongguo Guan, and Jianzhong Li. Digital twin-based life-cycle seismic performance assessment of a long-span cable-stayed bridge. *Bulletin of Earthquake Engineering*, 21(2):1203–1227, 2023.

[82] Esteban Bernal, Qing Wu, Maksym Spiryagin, and Colin Cole. Augmented digital twin for railway systems. *Vehicle System Dynamics*, 62(1):67–83, 2024.

[83] Shen Li and Feargal Brennan. Digital twin enabled structural integrity management: Critical review and framework development. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, page 14750902241227254, 2024.

[84] Rolando Chacón, Hector Posada, Carlos Ramonell, Manuel Jungmann, Timo Hartmann, Rehan Khan, and Rahul Tomar. Digital twinning of building construction processes. case study: A reinforced concrete cast-in structure. *Journal of Building Engineering*, 84:108522, 2024.

[85] Matteo Torzoni, Marco Tezzele, Stefano Mariani, Andrea Manzoni, and Karen E Willcox. A digital twin framework for civil engineering structures. *Computer Methods in Applied Mechanics and Engineering*, 418:116584, 2024.

[86] Charles R Farrar and Nick AJ Lieven. Damage prognosis: the future of structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):623–632, 2007.

[87] Dominika Ziaja and Piotr Nazarko. Shm system for anomaly detection of bolted joints in engineering structures. In *Structures*, volume 33, pages 3877–3884. Elsevier, 2021.

[88] Bernard Friedland. *Control system design: an introduction to state-space methods*. Courier Corporation, 2012.

[89] David R Wright. Finite state machines. *Carolina State University*, 203, 2005.

[90] Bernhard Kaiser. *State event fault trees: a safety and reliability analysis technique for software controlled systems*. Verlag Dr. Hut, 2006.

[91] Bart Peeters. System identification and damage detection in civil engeneering. 2000.

[92] Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*, volume 12. Siam, 2013.

[93] Carl Adam Petri. Kommunikation mit automaten. 1962.

[94] John Andrews, Darren Prescott, and Florian De Rozières. A stochastic model for railway track asset management. *Reliability Engineering & System Safety*, 130, 2014. pp: 76-84.

[95] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4), 1989. pp: 541-580.

[96] Kurt Jensen and Grzegorz Rozenberg. *High-level Petri nets: theory and application*. Springer Science & Business Media, 2012. ISBN: 978-3-642-84524-6.

[97] Eva Grafahrend-Belau, Falk Schreiber, Monika Heiner, Andrea Sackmann, Björn H Junker, Stefanie Grunwald, Astrid Speer, Katja Winder, and Ina Koch. Modularization of biochemical networks based on classification of petri net t-invariants. *BMC bioinformatics*, 9:1–17, 2008.

[98] Zhen Wang, Jian-Min Gao, Rong-Xi Wang, Kun Chen, Zhi-Yong Gao, and Wei Zheng. Failure mode and effects analysis by using the house of reliability-based rough vikor approach. *IEEE Transactions on Reliability*, 67(1):230–248, 2017.

[99] Charles R Farrar, Scott W Doebling, and David A Nix. Vibration–based structural damage identification. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 359(1778):131–149, 2001.

[100] Irem Y Tumer. System design and analysis methods. *System Health Management: With Aerospace Applications*, pages 129–143, 2011.

[101] Arman Malekloo, Ekin Ozer, Mohammad AlHamaydeh, and Mark Girolami. Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, 21(4):1906–1955, 2022.

[102] Eloi Figueiredo and Adam Santos. Machine learning algorithms for damage detection. In *Vibration-based techniques for damage detection and localization in engineering structures*, pages 1–39. World Scientific, 2018.

[103] Anders Rytter. Vibrational based inspection of civil engineering structures. 1993.

[104] Keith Worden, Charles R Farrar, Graeme Manson, and Gyuhae Park. The fundamental axioms of structural health monitoring. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2082):1639–1664, 2007.

[105] T Stepinski, T Uhl, and W Staszewski. Advanced structural damage detection-from theory to engineering applications, john wiley and sons, 2013.

[106] Charles R Farrar and Keith Worden. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[107] Rongshuai Li, Akira Mita, and Jin Zhou. Hybrid methodology for structural health monitoring based on immune algorithms and symbolic time series analysis. *Journal of Intelligent Learning Systems and Applications*, 5(01):48–56, 2013.

[108] Juan Chiachío, Manuel Chiachío, Shankar Sankararaman, Abhinav Saxena, and Kai Goebel. Condition-based prediction of time-dependent reliability in composites. *Reliability Engineering & System Safety*, 142:134–147, 2015.

[109] Jinming Zou, Yi Han, and Sung-Sau So. Overview of artificial neural networks. *Artificial neural networks: methods and applications*, pages 14–22, 2009.

[110] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[111] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[112] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[113] Stephen Grossberg. Recurrent neural networks. *Scholarpedia*, 8(2):1888, 2013.

[114] Francisco De Leon, Pablo Gómez, Juan A Martinez-Velasco, and Michel Rioual. Transformers. In *Power system transients*, pages 177–250. CRC Press, 2017.

[115] Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, and Andrea Mechelli. Autoencoders. In *Machine learning*, pages 193–208. Elsevier, 2020.

[116] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[117] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[118] Juan Fernández, Juan Chiachío, Manuel Chiachío, José Barros, and Matteo Corbetta. Physics-guided bayesian neural networks by abc-ss: Application to reinforced concrete columns. *Engineering Applications of Artificial Intelligence*, 119:105790, 2023.

[119] Wei Zhang, Gaoliang Peng, Chuanhao Li, Yuanhang Chen, and Zhujun Zhang. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, 17(2):425, 2017.

[120] Weining Lu, Bin Liang, Yu Cheng, Deshan Meng, Jun Yang, and Tao Zhang. Deep model based domain adaptation for fault diagnosis. *IEEE Transactions on Industrial Electronics*, 64(3):2296–2305, 2016.

[121] Chenhui Jiang, Qifeng Zhou, Jiayan Lei, and Xinhong Wang. A two-stage structural damage detection method based on 1d-cnn and svm. *Applied Sciences*, 12(20):10394, 2022.

[122] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151:107398, 2021.

[123] Hikaru Mizutani. The back propagation method for cnn. In *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, volume 6, pages 463–466. IEEE, 1994.

[124] Jakub M Tomczak. Why deep generative modeling? In *Deep Generative Modeling*, pages 1–12. Springer, 2021.

[125] Minkai Xu. Towards generalized implementation of wasserstein distance in gans. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10514–10522, 2021.

[126] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[127] Furkan Luleci and F Necati Catbas. A brief introductory review to deep generative models for civil structural health monitoring. *AI in Civil Engineering*, 2(1):9, 2023.

[128] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-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*, pages 1–6, 2019.

[129] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[130] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. Recent progress on generative adversarial networks (gans): A survey. *IEEE access*, 7:36322–36333, 2019.

[131] Toby RF Phillips, Claire E Heaney, Ellyess Benmoufok, Qingyang Li, Lily Hua, Alexandra E Porter, Kian Fan Chung, and Christopher C Pain. Multi-output regression with generative adversarial networks (mor-gans). *Applied Sciences*, 12(18):9209, 2022.

[132] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[133] George Tsialiamanis, David J Wagg, Nikolaos Dervilis, and Keith Worden. On generating parametrised structural data using conditional generative adversarial networks. In *Data Science in Engineering, Volume 9: Proceedings of the 39th IMAC, A Conference and Exposition on Structural Dynamics 2021*, pages 35–46. Springer, 2022.

[134] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[135] Gartner says advanced analytics is a top business priority. Technical report, Gartner INC, 2014.

[136] Juan Chiachío-Ruano, Manuel Chiachio-Ruano, and Shankar Sankararaman. *Bayesian Inverse Problems: Fundamentals and Engineering Applications*. CRC Press, 2021.

[137] James L Beck. Bayesian system identification based on probability logic. *Structural Control and Health Monitoring*, 17(7):825–847, 2010.

[138] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[139] James L Beck and Yong Huang. Bayesian uncertainty quantification and sparse bayesian learning for model updating in structural health monitoring.

[140] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

[141] Christian P Robert, George Casella, Christian P Robert, and George Casella. The metropolis—hastings algorithm. *Monte Carlo statistical methods*, pages 267–320, 2004.

[142] Thomas Clemen, Nima Ahmady-Moghaddam, Ulfia A Lenfers, Florian Ocker, Daniel Osterholz, Jonathan Ströbele, and Daniel Glake. Multi-agent systems and digital twins for smarter cities. In *Proceedings of the 2021 ACM SIGSIM conference on principles of advanced discrete simulation*, pages 45–55, 2021.

[143] Ameer Alaasam, Gleb Radchenko, Andrei Tchernykh, Kirill Borodulin, and Alexey Podkorytov. Scientific micro-workflows: where event-driven approach meets workflows to support digital twins. In , pages 489–495, 2018.

[144] Stefanie James, Chris Harbron, Janice Branson, and Mimmi Sundler. Synthetic data use: exploring use cases to optimise data utility. *Discover Artificial Intelligence*, 1(1):15, 2021.

[145] Nicholas E Silionis and Konstantinos N Anyfantis. Data-driven probabilistic quantification and assessment of the prediction error model in damage detection applications. *Probabilistic Engineering Mechanics*, page 103412, 2023.

[146] Debejyo Chakraborty, Narayan Kovvali, Bhavana Chakraborty, Antonia Papandreou-Suppappola, and Aditi Chattopadhyay. Structural damage detection with insufficient data using transfer learning

techniques. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2011*, volume 7981, pages 1175–1183. SPIE, 2011.

[147] Kyle Dunphy, Mohammad Navid Fekri, Katarina Grolinger, and Ayan Sadhu. Data augmentation for deep-learning-based multiclass structural damage detection using limited information. *Sensors*, 22(16):6193, 2022.

[148] Furkan Luleci, F Necati Catbas, and Onur Avci. A literature review: Generative adversarial networks for civil structural health monitoring. *Front. Built Environ. Struct. Sens. Control Asset Manag*, 8:1027379, 2022.

[149] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, AS Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H Al-Timemy, et al. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10(1):46, 2023.

[150] George Tsialiamanis, David J Wagg, Nikolaos Dervilis, and Keith Worden. On generative models as the basis for digital twins. *Data-Centric Engineering*, 2:e11, 2021.

[151] Hansoo Lee, Jonggeun Kim, Eun Kyeong Kim, and Sungshin Kim. Wasserstein generative adversarial networks based data augmentation for radar data analysis. *Applied Sciences*, 10(4):1449, 2020.

[152] Yizhou Zhuang, Jiacheng Qin, Bin Chen, Chuanzhi Dong, Chenbo Xue, and Said M Easa. Data loss reconstruction method for a bridge weigh-in-motion system using generative adversarial networks. *Sensors*, 22(3):858, 2022.

[153] Michael G Kapteyn and Karen 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, USA, October 2-4, 2020, Proceedings 3*, pages 3–7. Springer, 2020.

[154] Furkan Luleci, F Necati Catbas, and Onur Avci. Generative adversarial networks for labeled acceleration data augmentation for structural damage detection. *Journal of Civil Structural Health Monitoring*, 13(1):181–198, 2023.

[155] Thomas J. Gibbons, Stephen Gareth Pierce, Keith Worden, and Ifigeneia Antoniadou. Convolutional neural networks for the detection of damaged fasteners in engineering structures. 2018.

[156] Haichen Shi, Keith Worden, and Elizabeth 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*, 120:16–31, 2019.

[157] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[158] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[159] Azad M Madni, Carla C Madni, and Scott D Lucero. Leveraging digital twin technology in model-based systems engineering. *Systems*, 7(1):7, 2019.

[160] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52, 2020.

[161] Mengmeng Wang, Shizhe Feng, Atilla Incecik, Grzegorz Królczyk, and Zhixiong Li. Structural fatigue life prediction considering model uncertainties through a novel digital twin-driven approach. *Computer Methods in Applied Mechanics and Engineering*, 391:114512, 2022.

[162] Travis Goodwin, Jie Xu, Nurcin Celik, and Chun-Hung Chen. Real-time digital twin-based optimization with predictive simulation learning. *Journal of Simulation*, pages 1–18, 2022.

[163] Qinglin Qi, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang, and AYC Nee. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 58:3–21, 2021.

[164] Souvik Chakraborty, Sondipon Adhikari, and Ranjan Ganguli. The role of surrogate models in the development of digital twins of dynamic systems. *arXiv preprint arXiv:2001.09292*, 2020.

[165] Sofiat O Abioye, Lukumon O Oyedele, Lukman Akanbi, Anuoluwapo Ajayi, Juan Manuel Davila Delgado, Muhammad Bilal, Olugbenga O Akinade, and Ashraf Ahmed. Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges. *Journal of Building Engineering*, 44:103299, 2021.

[166] Gozde Basak Ozturk. Interoperability in building information modeling for aeco/fm industry. *Automation in Construction*, 113:103122, 2020.

[167] Andrew White. By 2024, 60 per cent of the data used for the development of ai and analytics projects will be synthetically generated. Technical report, Gartner consultancy, 2021.

[168] Kosmas Alexopoulos, Nikolaos Nikolakis, and George Chryssolouris. Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. *International Journal of Computer Integrated Manufacturing*, 33(5):429–439, 2020.

[169] Stefan Mihai, Mahnoor Yaqoob, Dang V Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V Prasad, et al. Digital twins: a survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys & Tutorials*, 2022.

[170] Sergio Cantero-Chinchilla, James L. Beck, Manuel Chiachío, Juan Chiachío, Dimitrios Chronopoulos, and Arthur Jones. Optimal sensor and actuator placement for structural health monitoring via an efficient convex cost-benefit optimization. *Mechanical Systems and Signal Processing*, 144:106901, 2020.

[171] C. Arcadius Tokognon, B. Gao, Gui Yun Tian, and Yan Yan. Structural health monitoring framework based on internet of things: A survey. *IEEE Internet of Things Journal*, 4(3), 2017. pp: 619-635.

[172] Kenneth M Zemrowski. Open systems interconnection. *Handbook of Networking & Connectivity*, pages 3–60, 1994.

[173] Pramod Pandya. Transmission control protocol/internet protocol packet analysis. In *Computer and Information Security Handbook*, pages e205–e218. Elsevier, 2013.

[174] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. Rfc2616: Hypertext transfer protocol–http/1.1, 1999.

[175] MA Simarro, Paola Guzmán, MA Rodríguez, Pau Arce, Gema Piñero, Alberto Gonzalez, and Juan Carlos Guerri. Open standards for the internet of things. Waves, 2018.

[176] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California Irvine, 2000.

[177] Jani Hietala. Real-time two-way data transfer with a digital twin via web interface. Master's thesis, Aalto University, 2020.

[178] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6), 1953. pp: 1087-1092.

[179] Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling for various metropolis-hastings algorithms. *Statistical science*, 16(4), 2001. pp: 351-367.

[180] Minjie Zhu, Frank McKenna, and Michael H Scott. Openseespy: Python library for the opensees finite element framework. *SoftwareX*, 7:6–11, 2018.

[181] Abdul Jabbar, Xi Li, and Bourahla Omar. A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys (CSUR)*, 54(8):1–49, 2021.

[182] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[183] David Bowes, Tracy Hall, and David 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*, pages 109–118, 2012.

[184] Luma Omar and Ioannis Ivrissimtzis. Using theoretical roc curves for analysing machine learning binary classifiers. *Pattern Recognition Letters*, 128:447–451, 2019.

[185] Mithat Gönen et al. Receiver operating characteristic (roc) curves. *SAS Users Group International (SUGI)*, 31:210–231, 2006.