Artificial Intelligence for Real Time Applications in Digital Cultural Heritage

Carmine Valentino

UNIVERSITY OF SALERNO

DEPARTMENT OF INDUSTRIAL ENGINEERING

Ph.D. Course in Industrial Engineering
Curriculum in Electronic Engineering
XXXIII Cycle

Artificial Intelligence for Real Time Applications in Digital Cultural Heritage

Supervisor
Prof. Francesco Colace
Prof. Dajana Conte

Ph.D. student
Carmine Valentino

Ph.D. Course Coordinator
Prof. Massimo De Santo

Scientific Production

Contributions to journals

[1] F. Colace, D. Conte, M. De Santo, M. Lombardi, D. Santaniello, C. Valentino. *A content-based recommendation approach based on singular value decomposition*. Connect. Sci., vol. 34, n. 1, pp. 2158 – 2176, 2022.
DOI:10.1080/09540091.2022.2106943

[2] F. Colace, D. Conte, M. De Santo, M. Lombardi, B. Paternoster, D. Santaniello, C. Valentino. *Recommender systems: a novel approach based on singular value decomposition*. Int. J. Electr. Comput. Eng., vol. 12, n. 6, pp. 6513 – 6521, 2022.
DOI:10.11591/ijece.v12i6.pp6513-6521

[3] M. Casillo, B. B. Gupta, M. Lombardi, A. Lorusso, D. Santaniello, C. Valentino. *Context Aware Recommender Systems: A Novel Approach Based on Matrix Factorization and Contextual Bias*. Electron. (Switz.), vol. 11, n. 7, 2022.
DOI:10.3390/electronics11071003

[4] G. Frasca-Caccia, C. Valentino, F. Colace, D. Conte. *An overview of differential models for corrosion of cultural heritage artefacts*. Math. Model. Nat. Phenom., vol. 18, 2023.
DOI:10.1051/mmnp/2023031

[5] M. Casillo, F. Colace, D. Conte, M. Lombardi, D. Santaniello, C. Valentino. *Context-aware recommender systems and cultural heritage: a survey*. J. Ambient Intell. Humaniz. Comput., vol. 14, n. 4, pp. 3109 – 3127, 2023.
DOI:10.1007/s12652-021-03438-9

[6] D. Conte, E. Cuesta, C. Valentino. *Non-stationary wave relaxation methods for general linear systems of Volterra equations: convergence and parallel GPU implementation*. Numer. Algorithms, vol. 95, n. 1, pp. 149 – 180, 2024.
DOI:10.1007/s11075-023-01567-0

[7] V. Arya, R. W. Attar, A. Alhomoud, M. Casillo, F. Colace, D. Conte, M. Lombardi, D. Santaniello, C. Valentino. *FANE: A FAke NEws Detector Based on Syntactic, Semantic, and Social Features Bayesian Analysis*. Int. J. Semant. Web Inf. Syst., vol. 20, n. 1, 2024.
DOI:10.4018/IJSWIS.360785

[8] M. Casillo, F. Colace, R. Gaeta, A. Lorusso, D. Santaniello, C. Valentino. *Revolutionizing cultural heritage preservation: an innovative IoT-based framework for protecting historical buildings*. Evol. Intell., vol. 17, n. 5-6, pp. 3815 – 3831, 2024.
DOI:10.1007/s12065-024-00959-y

[9] M. Casillo, F. Colace, A. Lorusso, D. Santaniello, C. Valentino. *A multilevel graph approach for IoT-based complex scenario management through situation awareness and*

semantic approaches. J. Reliab. Intell. Environ., vol. 10, n. 4, pp. 395 – 411, 2024.
DOI:10.1007/s40860-024-00224-0

[10] F. Colace, D. Conte, G. Pagano, B. Paternoster, C. Valentino. *PHYSICS-INFORMED NEURAL NETWORKS FOR A LITHIUM-ION BATTERIES MODEL: A CASE OF STUDY*. Adv. comput. sci. eng., vol. 2, n. 4, pp. 354 – 367, 2024.
DOI:10.3934/acse.2024018

[11] M. Casillo, F. Colace, A. Lorusso, D. Santaniello, C. Valentino. *Integrating Physical and Virtual Experiences in Cultural Tourism: An Adaptive Multimodal Recommender System*. IEEE Access, vol. 13, 2025.
DOI:10.1109/ACCESS.2025.3539205

[12] C. Valentino, G. Pagano, D. Conte, B. Paternoster, F. Colace, M. Casillo. *Step-by-step time discrete Physics-Informed Neural Networks with application to a sustainability PDE model*. Math. Comput. Simul., vol. 230, 2025.
DOI:10.1016/j.matcom.2024.10.043

[13] M. Casillo, F. Colace, M. Lombardi, A. Lorusso, D. Santaniello, C. Valentino. *ARTU: An Approach for Recommending Trips to Users*. IEEE Access, vol. 13, 2025.
DOI:10.1109/ACCESS.2025.3627105

[14] M. Casillo, F. Colace, D. Conte, M. Lombardi, D. Santaniello, C. Valentino. *A context-aware recommender system-based framework for improving cultural experiences*. User Model. User-Adapt. Interact., vol. 36:1, 2026.
DOI:10.1007/s11257-025-09437-1

Contributions to book chapters

[16] M. Casillo, F. Colace, B. B. Gupta, A. Lorusso, D. Santaniello, C. Valentino. *The role of AI in improving interaction with cultural heritage: An overview*. Handbook of Research on AI and ML for Intelligent Machines and Systems, 2023.
DOI:10.4018/978-1-6684-9999-3.ch006

[17] F. Colace, B. B. Gupta, A. Lorusso, A. Troiano, D. Santaniello, C. Valentino. *Unsupervised learning techniques for vibration-based structural health monitoring systems driven by data: A general overview*. Handbook of Research on AI and ML for Intelligent Machines and Systems, 2023.
DOI:10.4018/978-1-6684-9999-3.ch013

Contributions to conferences

[18] M. Casillo, F. Colace, B. B. Gupta, D. Santaniello, C. Valentino. *Fake News Detection Using LDA Topic Modelling and K-Nearest Neighbor Classifier*. Lect. Notes Comput. Sci., vol. 13116 LNCS, 2021.
DOI:10.1007/978-3-030-91434-9_29

[19] M. Casillo, D. Conte, M. Lombardi, D. Santaniello, C. Valentino. *Recommender System for Digital Storytelling: A Novel Approach to Enhance Cultural Heritage*. Lect. Notes Comput. Sci., vol. 12667 LNCS, 2021.
DOI:10.1007/978-3-030-68787-8_22

[20] M. Carbone, F. Colace, M. Lombardi, F. Marongiu, D. Santaniello, C. Valentino. *An Adaptive Learning Path Builder based on a Context Aware Recommender System*. Proceedings - FIE, vol. 2021-October, 2021.

DOI:10.1109/FIE49875.2021.9637465

[21] M. Casillo, M. D. Santo, M. Lombardi, R. Mosca, D. Santaniello, C. Valentino. *Recommender Systems and Digital Storytelling to Enhance Tourism Experience in Cultural Heritage Sites*. Proceedings - SMARTCOMP 2021, 2021.
DOI:10.1109/SMARTCOMP52413.2021.00067

[22] M. Casillo, D. Conte, M. Lombardi, D. Santaniello, A. Troiano, C. Valentino. *A Content-Based Recommender System for Hidden Cultural Heritage Sites Enhancing*. Lect. Notes Netw. Syst., vol. 217, 2022.
DOI:10.1007/978-981-16-2102-4_9

[23] E. Landolfi, A. Lorusso, F. Marongiu, D. Santaniello, A. Troiano, C. Valentino. *A Multilevel Approach for Smart Buildings Management*. Proceedings - SMARTCOMP 2022, 2022.
DOI:10.1109/SMARTCOMP55677.2022.00072

[24] M. Casillo, M. Lombardi, A. Lorusso, F. Marongiu, D. Santaniello, C. Valentino. *Sentiment Analysis and Recurrent Radial Basis Function Network for Bitcoin Price Prediction*. MELECON 2022, Proceedings, 2022.
DOI:10.1109/MELECON53508.2022.9842889

[25] M. Casillo, F. Colace, B. B. Gupta, A. Lorusso, F. Marongiu, D. Santaniello, C. Valentino. *A Situation Awareness Approach for Smart Home Management*. ISMODE 2021, 2022.
DOI:10.1109/ISMODE53584.2022.9742901

[26] M. Carratù, F. Colace, A. Lorusso, A. Pietrosanto, D. Santaniello, C. Valentino. *Data Mining Techniques for Intrusion Detection on the Internet of Things Field*. Lect. Notes Netw. Syst., vol. 599 LNNS, 2023.
DOI:10.1007/978-3-031-22018-0_1

[27] M. Casillo, L. Cecere, F. Colace, A. Lorusso, D. Santaniello, C. Valentino. *Exhibition spaces in the metaverse: A novel design approach*. HISTELCON 2023, Proceedings, 2023.
DOI:10.1109/HISTELCON56357.2023.10365847

[28] F. Colace, D. Conte, G. Frasca-Caccia, A. Lorusso, D. Santaniello, C. Valentino. *An IoT-based framework for the enjoyment and protection of Cultural Heritage Artifacts*. Proceedings - WoWMoM 2023, 2023.
DOI:10.1109/WoWMoM57956.2023.00085

[29] F. Colace, M. P. D'Arienzo, A. Lorusso, M. Lombardi, D. Santaniello, C. Valentino. *A Novel Context Aware Paths Recommendation Approach for the Cultural Heritage Enhancement*. Proceedings - SMARTCOMP 2023, 2023.
DOI:10.1109/SMARTCOMP58114.2023.00071

[30] M. Casillo, M. De Santo, M. Lombardi, R. Mosca, D. Santaniello, C. Valentino. *An IoT Architecture to Enhance Monitoring and Predictive Maintenance for Cultural Heritage Buildings*. Lect. Notes Netw. Syst., vol. 447, 2023.
DOI:10.1007/978-981-19-1607-6_41

[31] F. Colace, D. Conte, M. P. D'Arienzo, D. Santaniello, A. Troiano, C. Valentino. *A Path Recommender System for Enjoyment Improvement of the Cultural Heritage*. Lect. Notes Netw. Syst., vol. 694 LNNS, 2023.
DOI:10.1007/978-981-99-3091-3_88

[32] F. Colace, D. Conte, B. Gupta, D. Santaniello, A. Troiano, C. Valentino. *A Novel Context-Aware Recommendation Approach Based on Tensor Decomposition*. Lect. Notes Netw. Syst., vol. 448, 2023.
DOI:10.1007/978-981-19-1610-6_39

[33] L. Cecere, F. Colace, M. Lombardi, A. Lorusso, D. Santaniello, C. Valentino. *Cultural Heritage Enhancement through Digital Storytelling and Context-Aware Recommender System*. ACM International Conference Proceeding Series, 2023.

DOI:10.1145/3617233.3617241

[34] M. Casillo, F. Colace, F. Marongiu, D. Santaniello, C. Valentino. *Gamification in Cultural Heritage: When History Becomes SmART*. Lect. Notes Comput. Sci., vol. 14366, 2024. DOI:10.1007/978-3-031-51026-7_33

[35] F. Colace, R. Gaeta, M. Lombardi, A. Lorusso, D. Santaniello, C. Valentino. *An Architecture for Cultural Heritage enhancement based on the Internet of Things and a Hybrid Context-Aware Recommender System*. ICA-ACCA 2024, 2024. DOI:10.1109/ICA-ACCA62622.2024.10766774

[36] M. Casillo, L. Cecere, F. Colace, A. Lorusso, D. Santaniello, C. Valentino. *Digital Twin and Metaverse Supporting Smart Cities: New Perspectives and Potentials*. Lect. Notes Netw. Syst., vol. 828, 2024. DOI:10.1007/978-981-99-8111-3_11

[37] M. Casillo, L. Cecere, S. P. Dembele, A. Lorusso, D. Santaniello, C. Valentino. *The Metaverse and Revolutionary Perspectives for the Smart Cities of the Future*. Lect. Notes Netw. Syst., vol. 1004 LNNS, 2024. DOI:10.1007/978-981-97-3305-7_17

[38] L. Cecere, F. Colace, M. De Santo, A. Lorusso, D. Santaniello, C. Valentino. *Overview of Cultural Heritage Education and Emerging Technologies*. IHTC 2024, 2024. DOI:10.1109/IHTC61819.2024.10855153

[39] T. Y. Melesse, F. Colace, S. P. Dembele, A. Lorusso, D. Santaniello, C. Valentino. *Digital Twin for Predictive Monitoring of Crops: State of the Art*. Lect. Notes Netw. Syst., vol. 695 LNNS, 2024. DOI:10.1007/978-981-99-3043-2_85

[40] L. Cecere, F. Colace, B. Gupta, A. Lorusso, B. Messina, C. Valentino. *Metaverse and Museum: A Case Study*. Procedia Struct. Integr., vol. 64, 2024. DOI:10.1016/j.prostr.2024.09.336

[41] F. Colace, R. Gaeta, D. Santaniello, A. Troiano, C. Valentino. *A Framework Based on Internet of Things and Recommender Systems for Thermal Facilities*. Lect. Notes Netw. Syst., vol. 1054 LNNS, 2025. DOI:10.1007/978-981-97-5035-1_8

[42] M. Casillo, F. Colace, A. Lorusso, D. Santaniello, C. Valentino. *Improving Enjoyment of Cultural Heritage Through Recommender Systems, Virtual Tour, and Digital Storytelling*. International Conference on Pattern Recognition Applications and Methods, vol. 1, 2025. DOI:10.5220/0013176000003905

[43] M. Casillo, F. Colace, A. Lorusso, D. Santaniello, C. Valentino. *A Multilevel Graph-Based Recommender System for Personalized Learning Paths in Archaeological Parks: Leveraging IoT and Situation Awareness*. IoTBDS - Proceedings, 2025. DOI:10.5220/0013134500003944

[44] M. Casillo, L. Cecere, F. Colace, A. Lorusso, D. Santaniello, C. Valentino. *A Framework to Improve Cultural Experience through Metaverse and Recommender Systems*. Proceedings - VRW 2025, 2025. DOI:10.1109/VRW66409.2025.00123

[45] F. Colace, S. P. Dembele, R. Gaeta, A. Lorusso, D. Santaniello, C. Valentino. *Enhancing Agricultural Practices Through Iot And Decision Tree Analytics: A Case Study On Autonomous Irrigation Management*. Lect. Notes Netw. Syst., vol. 1178, 2025. DOI:10.1007/978-981-97-9559-8_20

[46] M. Casillo, R. Gaeta, A. Lorusso, F. Marongiu, D. Santaniello, C. Valentino. *A Novel Architecture for Enhancing Museum Visits Through Recommender Systems, Digital Storytelling, and NFT*. Lect. Notes Netw. Syst., vol. 1054 LNNS, 2025. DOI:10.1007/978-981-97-5035-1_9

[47] M. Casillo, F. Colace, M. C. De Simone, A. Lorusso, D. Santaniello, C. Valentino. *A Machine Learning-Based Architecture for Structural Health Monitoring*. Lect. Notes Netw. Syst., vol. 1144, 2025.
DOI:10.1007/978-981-97-7839-3_30

[48] D. Battista, F. Colace, A. Lorusso, D. Santaniello, A. Tucker, C. Valentino. *Scalable Strategy for Fake News Detection in Smart Cities Combining AI and Semantic Web Technologies*. Lect. Notes Netw. Syst., vol. 1444 LNNS, 2025.
DOI: 10.1007/978-981-96-6932-5_33

[49] M. Casillo, F. Colace, R. Gaeta, M. Lombardi, D. Santaniello, C. Valentino. *An Architecture Based on Recommender Systems and Context Awareness to Suggest Personalized Cultural Experiences*. Lect. Notes Netw. Syst., vol. 1444 LNNS, 2025.
DOI:10.1007/978-981-96-6932-5_18

Contents

List of Figures

List of Tables

Abstract

The digital cultural heritage (CH) domain is rapidly evolving through the adoption of digital infrastructures, immersive media, and data-centric technologies. These advances enable new forms of cultural engagement, such as adaptive narratives, personalised itineraries, and hybrid physical–digital experiences, while also supporting conservation activities through quantitative monitoring, structural interpretation, and preventive planning. Despite this potential, CH remains a challenging application field due to heterogeneous, often sparse data, non-standardised documentation, ethical constraints, and the need to protect fragile, irreplaceable assets. Within this context, two research directions are both crucial and commonly addressed in isolation:

- enhancing cultural heritage and improving users' enjoyment;
- guaranteeing predictive maintenance and preservation of cultural assets.

This dissertation addresses these challenges by proposing a unified framework that integrates the Internet of Things (IoT), Artificial Intelligence (AI), and physically-grounded modelling to support both user-facing enhancement services and expert-facing conservation workflows. The framework is designed as a modular architecture composed of four functional layers: a Data Acquisition Layer for collecting heterogeneous data streams and digital models; a Knowledge-Base Layer for organising, storing, and reusing information across tasks; an Inference Engine Layer for executing learning- and physics-based pipelines; and an Application Layer for visualisation, interaction, and decision support. A core contribution is enabling the processing of complex 3D geometries via Blender APIs, allowing digital replicas of cultural assets to be automatically preprocessed and converted into simulation-ready representations for Scientific Machine Learning (SciML) workflows. Within the inference layer, the dissertation focuses on Physics-Informed Neural Networks (PINNs) and Reduced Order Models (ROMs) as complementary tools to handle data scarcity, enforce physical consistency, and reduce computational costs in high-dimensional differential problems.

The unified framework is assessed through two application scenarios that instantiate the same layered architecture to address both visitor-facing enhancement and expert-facing conservation requirements. On the enhancement side,

the framework integrates a context-aware recommendation pipeline coupled with conversational interaction and non-linear digital storytelling. User context is represented through a novel modality, defined as embedded context, and exploited to generate personalised cultural suggestions that reflect both user preferences and situational constraints. This workflow is evaluated through offline accuracy analysis and an in situ study conducted during real cultural visits, where user satisfaction supports its feasibility in realistic operating conditions. On the conservation side, the framework supports predictive maintenance workflows grounded in physics-based learning. The experimental validation investigates the combined use of Physics-Informed Neural Networks (PINNs) and Reduced Order Models (ROMs) for parameterised differential problems, highlighting how reduced models can accelerate simulations and enable efficient exploration of physical parameters related to degradation and structural dynamics. It also demonstrates PINNs' ability to solve direct problems by integrating observational data with governing equations under incomplete or noisy measurements. In addition, the dissertation introduces a methodological contribution to time-dependent differential problems: a time-discrete, step-by-step PINN strategy that embeds classical numerical time-integration schemes into the training objective, thereby strengthening temporal causality and improving robustness for simulations required for degradation forecasting.
The dissertation is structured to reflect the dual focus on enhancement and conservation within a shared digital ecosystem. Chapter 1 introduces the motivations and objectives. Chapter 2 presents the proposed architecture and its integration of data acquisition, AI, and physics-based methods. Chapter 3 focuses on cultural experience enhancement through embedded context and a novel AI-based context-aware recommender system. Chapter 4 develops the predictive maintenance task, showing how complex 3D assets are integrated into PINN-based simulations through automated geometry processing, together with network identification and validation. In addition, a real scenario application is described. Chapter 5 addresses limitations of continuous-in-time PINNs for evolutionary problems and introduces the step-by-step, time-discrete PINN approach, validated on a non-linear benchmark selected to emphasise temporal causality and long-horizon stability. Chapter 6 summarises the contributions and presents the conclusions.

Chapter 1

Introduction

In recent years, Artificial Intelligence (AI) has undergone a rapid expansion, moving from specialized industrial and academic sectors into the daily routines of non-expert users. Contemporary search engines, such as Google, leverage large language models like Gemini to provide concise, synthesized overviews of query results[1]. Similarly, conversational agents such as ChatGPT, by OpenAI, are now commonly adopted in research workflows and educational settings [148], although their use does not always follow best practices or appropriate methodological standards [49]. Furthermore, enterprises are integrating AI solutions to support decision-making and enhance employee productivity, while consumer devices such as smartwatches and smartphones employ machine learning–based predictive models to estimate sleep quality and other health-related parameters [113]. This ubiquity of AI, coupled with the heterogeneity of underlying techniques and goals, often leads to conceptual confusion and terminological inconsistency, even within the expert community. Moreover, AI applications can range from classical approaches such as expert systems [88, 143, 193] to recent advances in computer vision and large language models [66, 148, 162, 172]. Choosing the appropriate approach depends on several factors, including the level of control required over the model, the amount of available data, and the computational resources at hand. The target application domain also plays a crucial role in this choice [109].

However, any AI project must begin with a fundamental step: defining the problem [142]. This means that, before selecting a model or specifying technical requirements, we need to clearly understand the problem to be solved and the expected outcomes. After defining the problem, following the scheme shown in Figure 1.1, we can select an analytical strategy to address it, formalize the requirements for applying this strategy, define the corresponding AI solution, and evaluate its behaviour [65]. If the evaluation is satisfactory,

[1] https://www.nytimes.com/2025/11/18/technology/google-gemini-3.html

the process proceeds to model deployment, which should then be monitored and the outcomes appropriately communicated. Otherwise, the process returns to the analytical stage, where the problem resolution strategy and the chosen approach are revised in order to achieve the expected results.

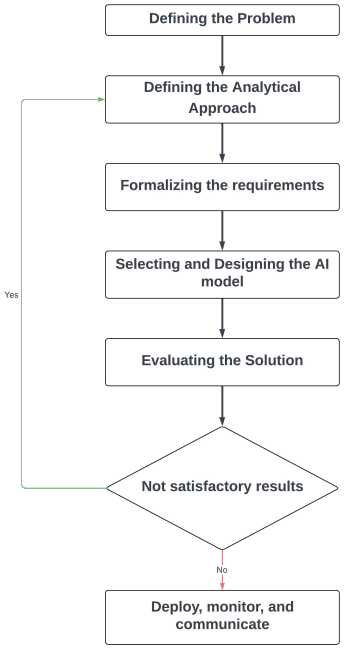


Figure 1.1. Problem-driven AI life-cycle.

This scheme will serve as the guiding framework for the present work. Specifically, this dissertation investigates how Artificial Intelligence can be systematically applied to address two complementary classes of problems in the Cultural Heritage domain through a unified, modular framework. The first concerns cultural experience and asks how AI-based, context-aware services can enrich visits by improving access to content, tailoring narratives to individual visitors, and supporting inclusive, user-centred fruition of sites and collections. The second focuses on conservation and management, exploring how predictive, model-based strategies can anticipate degradation phenomena, integrate heterogeneous monitoring data, and support risk-informed decision-making by heritage professionals. The following sections provide the methodological foundations for these two strands, introducing the main concepts of AI, Machine Learning, Deep Learning, and Scientific Machine Learning that support the unified and modular framework, as well as the case studies developed in the remainder of this dissertation.

1.1 Fundamentals of AI

The term Artificial Intelligence (AI) is closely tied to what we mean by *intelligence*. However, attempting to define intelligence directly quickly leads to abstract debates about whether and how a machine could *think*. A landmark step towards an operational perspective was Alan Turing's 1950 paper *Computing Machinery and Intelligence*, in which he introduced the *imitation game* [166], now known as the Turing test. In this test, a machine is said to be intelligent if its conversational behaviour cannot be reliably distinguished from that of a human interlocutor. Turing's proposal thus shifts the focus from metaphysical questions about the nature of thought to the observable behaviour of computing machines.

Building on this operational perspective, subsequent work in AI has broadened and refined the notion of machine intelligence. Russell and Norvig characterise AI as the study and design of agents that perceive their environment and act to achieve their goals, often formalised as rational agents that select actions expected to maximise a given performance measure [141]. In their terminology, an *agent* is any entity that perceives its environment through sensors and acts upon that environment through actuators. In this agent-based view, intelligence is not defined by the system's internal substrate, but by how appropriately it behaves in relation to its environment and objectives, thereby maintaining a direct connection to Turing's original behavioural criterion.

Within this general view, classical AI systems typically maintain an internal model of the world, expressed in a logic-like or rule-based form, and manipulate these symbols using inference procedures to derive conclusions and decide how to act. Among these approaches, expert systems are among the most influential instantiations of classical AI. They are computer programs designed to emulate the decision-making ability of a human expert in a narrowly defined domain. Expert systems separate domain knowledge from control: a knowledge base encodes facts and heuristics, while an inference engine applies the corresponding rules to the current case in order to derive recommendations or decisions [70, 79]. From an agent perspective, an expert system can be seen as a particular kind of AI agent whose sensors are the input facts provided by the user or by other software systems, whose internal state is the evolving knowledge base, and whose actions are the decisions or advisories that it outputs.

Following the era of classical AI, the widespread diffusion of the web and the Internet of Things (IoT) has created ecosystems in which billions of devices, services, and users continuously generate and exchange information. On the web, user interactions with on-line platforms, search queries, social networks, and digital content leave behind rich behavioural traces in the form of logs, click-streams, textual content, images, and videos [104]. At the same time, the IoT extends this data-rich landscape into the physical world: sensors embedded

in buildings, vehicles, industrial machinery, or mobile devices continuously measure temperature, vibration, energy consumption, location, and many other quantities [6, 14]. Together, the web and IoT create an environment where data is abundant, high-dimensional, and constantly evolving [152].
In this context, the limitations of purely symbolic, knowledge-based approaches become evident, and AI has increasingly shifted towards data-driven methods [114]. In AI, the term Machine Learning (ML) refers to the family of approaches in which a system learns its behaviour from data. Specifically, ML models infer patterns and regularities from data and use them to make predictions, make decisions, or generate new content [114].
Consequently, the remainder of this section is organised as follows: we first outline the core concepts of Machine Learning, and then introduce two major sub-fields that build on this paradigm, namely Deep Learning and Scientific Machine Learning. Deep Learning (DL) is a subfield of ML based on neural networks that automatically learn hierarchical representations from data and is particularly effective for high-dimensional, unstructured inputs such as images, text, audio, and complex sensor time series [92]. Scientific Machine Learning (SciML), in turn, combines ML techniques with physical knowledge [130].

1.1.1 Machine Learning

The rapid diffusion of IoT and web-based services has led to a growth in the volume, variety, and velocity of data, giving rise to the so-called Big Data problem [43]. Traditional storage and analysis solutions based on relational databases and sequential processing are no longer sufficient, motivating the adoption of new data management paradigms, advanced data mining and Artificial Intelligence techniques [69], and high-performance computing architectures such as GPU-based parallel processing [118]. Within this ecosystem, Machine Learning plays a central role, providing scalable methods to extract patterns and knowledge from large, complex datasets and to support descriptive, predictive, and prescriptive analytics.
Machine Learning (ML) techniques differ from classical AI approaches because they do not require explicit programming to solve a specific problem. ML can both simplify solution implementation and often achieve higher accuracy than conventional methods.
Arthur Samuel (1959) defined Machine Learning as a field of study that gives computers the ability to learn tasks for which they have not been explicitly programmed [146]. Later, Tom Mitchell (1997) formalized this idea by stating that a learning algorithm can be said to learn from experience E, with respect to some class of tasks T and a performance measure P, if its performance on tasks T, as measured by P, improves with experience E [111]. The main advantages of ML algorithms include their ability to tackle complex problems for which

traditional solution methods are unavailable or computationally prohibitive, their suitability for rapidly changing environments, where models must adapt to evolving data and conditions, and their capability to handle large volumes of data and extract meaningful patterns and information. In addition, ML methods can be classified accordingly to three main dimensions, as shown in the Figure 1.2: the degree of human supervision, the mode of learning over time, and the strategy adopted for prediction.

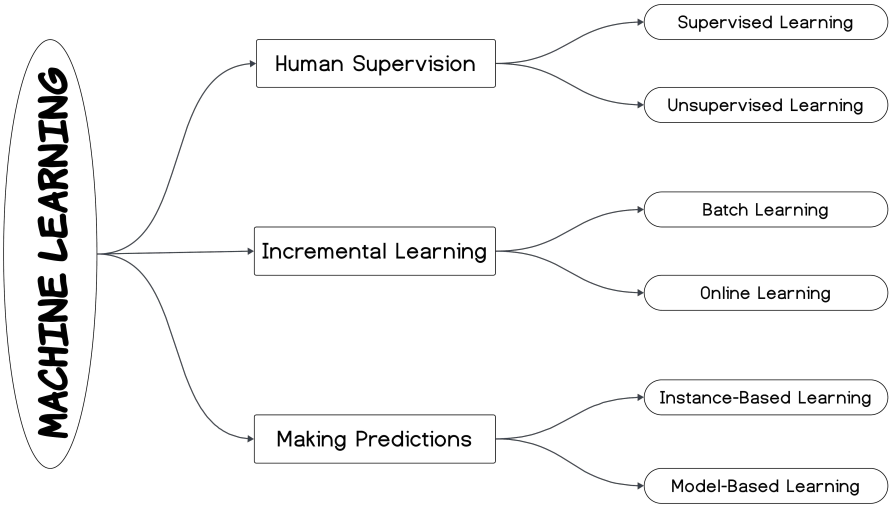


Figure 1.2. Scheme of machine learning methods classified by level of human supervision, learning modality, and prediction generation.

The first categorization is based on the type of human supervision required during training:

- Supervised Learning: the algorithm learns from labelled data, where each input instance is associated with a target label. The goal is to learn a mapping from input features to output labels that generalizes to unseen data. Supervised learning methods can be further divided into:
  - Classification algorithms: the label denotes membership of an instance in one of a finite set of classes.
  - Regression algorithms: the label is a numerical target value predicted from the input features, which are often referred to as predictors in this context.
- Unsupervised Learning: the training data are unlabelled, and the method must discover patterns without explicit guidance. In this setting, algorithms can:
  - Automatically learn association rules among variables;
  - Be employed for data visualization and dimensionality reduction,

as in Principal Component Analysis (PCA);
- Perform data clustering by grouping instances based on feature similarity. This requires adopting suitable distance or similarity measures, such as Euclidean distance, cosine similarity, or Minkowski distance.

The second dimension concerns how frequently the model is updated as new data becomes available:

- Batch Learning: the model is trained once on the entire available dataset. Because training may be computationally expensive, retraining is performed infrequently, typically only after substantial new data has accumulated;
- Online Learning: the model is updated incrementally as data arrives, often using small subsets of instances known as mini-batches, allowing rapid adaptation to new data and making it easier to track changes in the underlying data distribution.

The third classification relates to how algorithms generalize from the training data to make predictions:

- Instance-Based Learning: new instances are compared to previously observed ones using similarity measures. Predictions are derived from the labels of the most similar stored instances, typically via neighborhood-based reasoning;
- Model-Based Learning: the algorithm learns an explicit mathematical model of the relationship between inputs and outputs from the training data. This model is then used to predict the outputs for new, unseen inputs.

Within this theoretical framework, a broad spectrum of algorithms has been developed. Instance-based methods, such as k-Nearest Neighbours, implement the instance-based prediction strategy by assigning labels based on feature-space similarity [176]. Linear models with regularization, exemplified by Regularized Least Squares, offer simple yet effective estimators [97], while early neural models such as the Perceptron and Adaline introduce gradient-based learning rules for binary classification. Margin-based approaches, such as Support Vector Machines, further refine these ideas by maximizing the separation between classes in a suitably transformed feature space [41].
However, despite their success, these algorithms generally have limited capacity to capture highly non-linear patterns in the data. This limitation motivates the transition to deep learning, where multi-layer neural architectures, trained by gradient-based optimization, extend the principles of linear models and early neural classifiers to a far more expressive and scalable framework for modern Big Data applications.

1.1.2 *Deep Learning*

Deep learning is a subfield of Machine Learning that builds on Neural Networks (NNs). Unlike rule-based systems, NNs are function estimators that can learn patterns from data to perform tasks [72]. This data-driven paradigm has proved highly successful in a wide range of applications, including computer vision, speech recognition, video games, and medical diagnosis. From a modelling viewpoint, an NN is a non-linear function approximator composed of interconnected neurons organized in layers. The neuron represents the fundamental element of the NN and aims to send signals (real numbers) through the network. The connections between neurons define the network structure and how signals propagate to fit the data. Each neuron receives inputs from other neurons or directly from the data, combines them through a weighted sum, and then applies a non-linear activation function. Connections between neurons are represented as directed edges, each with a weight learned during training that controls the strength and sign of the transmitted signal. Groups of neurons are arranged into layers: an input layer that receives the features, one or more hidden layers that perform intermediate transformations, and an output layer that produces the final prediction. The strength of NNs lies precisely in this layered, nonlinear structure, which allows them to represent complex input–output relationships.
For simplicity, from now on in this subsection we refer to the neural network represented in Figure 1.3; it consists of an input layer with three neurons, an output layer with two neurons, and two hidden layers with four and five neurons, respectively. The network output f consists of the composition of the so-called activation functions (named g^ℓ in the figure), as follows:

$$f(x) = \left(g^3 \circ g^2 \circ g^1\right)(x) \in \mathbb{R}^2, \qquad x \in \mathbb{R}^3. \tag{1.1}$$

The g^ℓ functions are usually expressed as follows:

$$g^\ell(z) = \sigma\left(W^{(\ell)} z + b^{(\ell)}\right). \tag{1.2}$$

Here, the $W^{(\ell)}$ matrices contain the so-called weights of the network, and the $b^{(\ell)}$ vectors are named biases. They contain parameters to be set appropriately, as explained later. Referring to the example in Figure 1.3, we have:

$$\begin{array}{llllll} \text{for} & \ell = 1, & z \in \mathbb{R}^3, & W^{(1)} \in \mathbb{R}^{4\times 3}, & b^{(1)} \in \mathbb{R}^4; \\ \text{for} & \ell = 2, & z \in \mathbb{R}^4, & W^{(2)} \in \mathbb{R}^{5\times 4}, & b^{(2)} \in \mathbb{R}^5; \\ \text{for} & \ell = 3, & z \in \mathbb{R}^5, & W^{(3)} \in \mathbb{R}^{2\times 5}, & b^{(3)} \in \mathbb{R}^2. \end{array} \tag{1.3}$$

Consistently with the nomenclature introduced above, the function σ is known as activation function [72]; for NNs, different types of activation functions

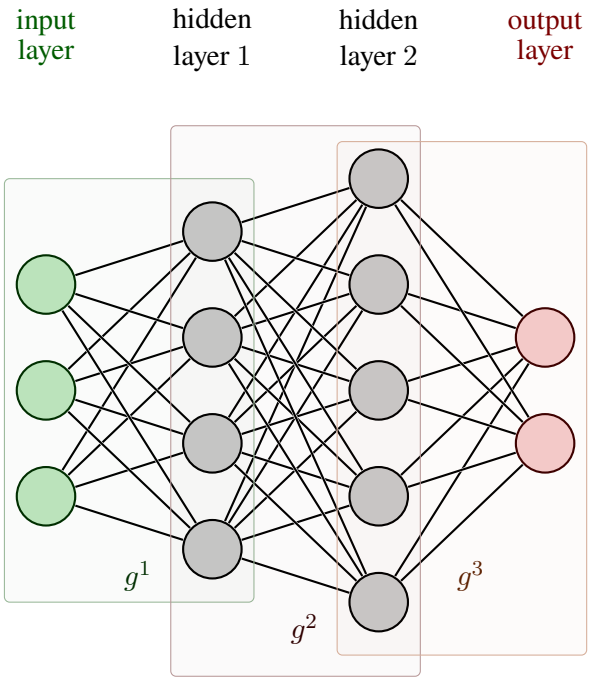


Figure 1.3. Example of NN with an input layer of three neurons, an output layer of two neurons, and two hidden layers of four and five neurons, respectively; the network output is the composition of the g^ℓ functions, see Eq. (1.1).

are usually considered, such as sigmoid, hyperbolic tangent, ReLU (REctified Linear Unit) [161] (see Figure 1.4).
Suppose we have to deal with a supervised learning problem based on the network in Figure 1.3. The classical application of NNs involves a data-driven approach based on a dataset

$$D = \left\{ \left(x^i, y^i \right) : x^i \in \mathbb{R}^3,\ y^i \in \mathbb{R}^2,\ i = 1,\ \ldots,\ M \right\}. \tag{1.4}$$

Here, $x^1,\ \ldots,\ x^M$ constitute the feature vectors, and $y^1,\ \ldots,\ y^M$ are the corresponding labels. The objective of the NN learning phase consists of identifying adequate weights and biases [72], see Eq. (1.3), by minimizing the so-called loss function, which in this example can be defined as follows:

$$\text{LOSS}\left(W^{(1)}, W^{(2)}, W^{(3)}, b^{(1)}, b^{(2)}, b^{(3)} \right) = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2} \| y^i - f\left(x^i \right) \|^2. \tag{1.5}$$

Therefore, the idea is to determine the parameters of the network so that the continuous model f is accurate on the training data. However, once trained, the model thus constructed can be used to make predictions on any data, naturally even those not belonging to the dataset D.

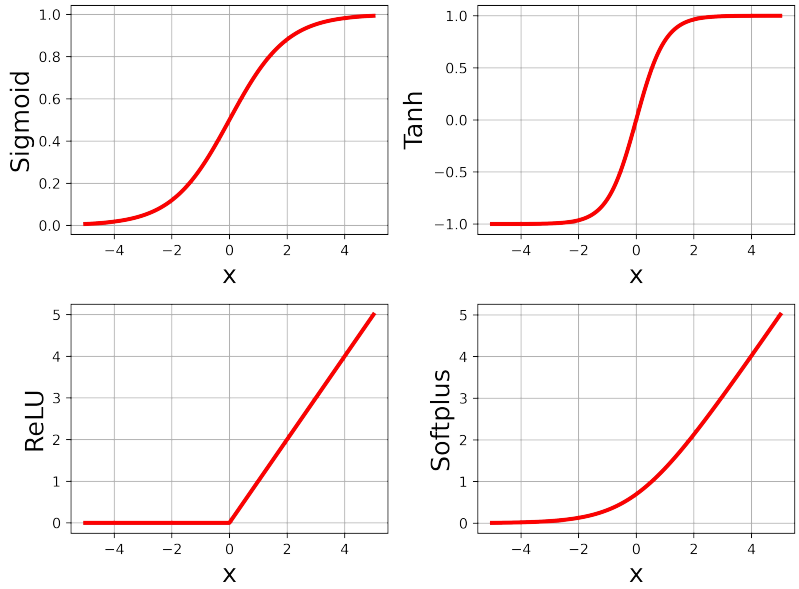


Figure 1.4. Examples of activation functions.

1.1.3 Scientific Machine Learning

Scientific machine learning (SciML) has emerged at the intersection of two complementary paradigms: traditional physics-based modelling and data-driven learning. On the one hand, classical scientific computing starts from a mechanistic understanding of the phenomenon, formulates governing principles as mathematical models, often as systems of Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs), and then approximates these models numerically. This workflow, from physical principles to mathematical formulation, discretisation, and algorithmic implementation, supports the standard notion of digital or computational modelling and has been thoroughly developed within numerical analysis and high-performance computing [130, 155]. On the other hand, modern machine learning, driven by the availability of large data sets, affordable accelerated hardware, and increasingly sophisticated learning architectures, seeks to infer input–output relations directly from data without explicitly invoking an underlying causality principle [92]. In this paradigm, Neural Networks (NNs) and related models are trained on example data to approximate complex, often highly non-linear mappings, and have proven particularly effective in settings where explicit models are incomplete or unavailable [72]. SciML does not replace physics-based models with machine learning surrogates; it integrates the two viewpoints into a unified computational framework, as shown by Figure 1.5.

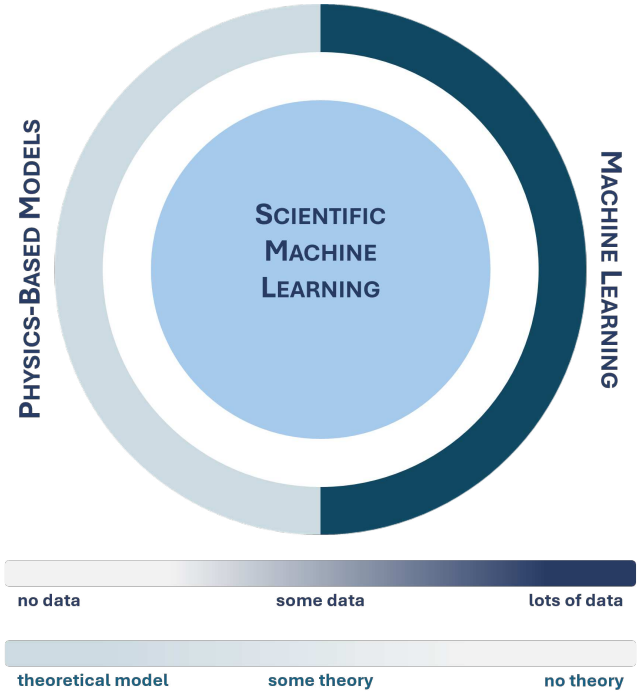


Figure 1.5. Scheme representing the main idea of Scientific Machine Learning: integrating physical knowledge and data-driven approaches.

The central idea is to exploit the expressive power of learning algorithms while systematically embedding physical structure into them. Domain knowledge can be incorporated through tailored loss functions, architectural design, or suitable parameterisations, thereby restricting the hypothesis space to model classes compatible with the underlying physics [16].
Within this general framework, several methodological directions have gained prominence. A first line of research concerns the construction of surrogate models for high-fidelity numerical solvers. Here, NNs or other learning architectures are trained on data generated by accurate but computationally intensive simulations to reproduce their input–output behaviour at a dramatically reduced cost. Such surrogates are particularly valuable for tasks that require repeated evaluations of the model, and they naturally complement reduced-order modelling techniques developed in numerical analysis [130]. A second, conceptually distinctive direction is physics-informed learning. Physics-Informed Neural Networks (PINNs) and their variational extensions incorporate the governing differential equations directly into the training objective: the network is penalised not only for discrepancies with observed data, but also for violations of the PDE residual or its weak form, allowing the learned solution to satisfy

the physical laws in an approximate sense even when observational data are scarce, noisy, or incomplete [81, 130, 135]. A third, rapidly developing strand of SciML aims at operator learning. Instead of approximating a single solution map, these methods learn non-linear operators that map coefficients, initial data, or forcing terms to solutions, observables, or reduced representations [130, 175]. Deep operator networks, neural operators, and related architectures are designed to generalise across varying geometries, boundary conditions, and parameter ranges, thereby moving beyond the traditional focus on fixed problem instances [130].

Category	**Main idea**	**Data Requirements**	**Advantages**
SURROGATE MODELLING	Fast approximation of high-fidelity numerical solvers through learned input–output mappings.	Moderate to large sets of simulated data.	Reduction in computational cost; suitable for many-query tasks.
PHYSICS-INFORMED LEARNING	Incorporation of PDEs, constraints, or variational principles directly into the neural network loss function.	Operates effectively with scarce or noisy data combined with physics-based residuals.	Respects physical laws; robust interpolation between sparse measurements; improved generalisation via model structure.
OPERATOR LEARNING	Learning non-linear operators that map functions to functions (e.g., coefficients or initial conditions to solution fields).	Sets of functional input–output pairs, typically obtained from simulations.	Generalises across geometries, boundary conditions, and parameter ranges; reusable across entire families of PDE problems.

Table 1.1. Summary of three representative categories of Scientific Machine Learning.

From this perspective, SciML provides a natural computational backbone for digital twins [175], in which high-fidelity numerical models and continuously updated data streams are coupled within a closed feedback loop. Learning-based surrogates and operator networks can drastically reduce the cost of large-scale simulations, enabling real-time state estimation, on-line calibration of material and constitutive parameters, and prediction of unobserved or future system responses. These features are practical in domains where direct experimentation is limited, and assets are unique or non-replaceable, as in the field of cultural heritage. Digital twins of historic buildings or cultural artefacts can

integrate structural and environmental models with monitoring data from heterogeneous sensor networks, allowing SciML methods to identify degradation mechanisms and support scenario testing for preventive conservation and risk assessment.

1.2 MOTIVATION AND GOALS

The ongoing digital transformation of the Cultural Heritage (CH) domain is generating significant opportunities for both visitor engagement and conservation practice [34, 46]. Digital infrastructures, immersive media, and data-driven tools can enrich visitors' experiences through adaptive narratives, personalised itineraries, and hybrid physical–digital interactions [36]. At the same time, these technological advances enable more systematic monitoring of material degradation, environmental conditions, and structural behaviour, fostering data-informed conservation strategies [46]. Within this scenario, two research directions have emerged as particularly critical:

- enhancing cultural enjoyment through contextualised, user-centred experiences;
- enabling predictive maintenance and preservation of cultural assets.

Building on the AI foundations discussed in Section 1.1, this thesis operationalises these two directions through complementary methodological paradigms: Machine Learning–driven personalisation for visitor-facing services (Context-Aware Recommender Systems and interactive storytelling) and Scientific Machine Learning for expert-facing analyses (physics-based and physics-informed modelling for monitoring and prediction). The core hypothesis is that both paradigms can be implemented as specialised inference workflows within the same digital infrastructure, thus promoting reuse of data, models, and evaluation assets across services.

Despite their apparent complementarity, these directions are often pursued independently. Visitor-oriented systems typically focus on interaction, personalisation, and engagement, without exploiting knowledge derived from monitoring streams and predictive models. Conversely, conservation-oriented approaches are designed for expert users and are rarely integrated with visitor-facing applications. This separation leads to fragmented solutions, duplicated efforts, and limited reuse of data and models across different services.

The first motivation for this dissertation is to address the growing need for cultural experiences that are not only digitally enhanced but also contextual, adaptive, and user-centred. In real-world heritage sites and museums, cultural content is extensive, heterogeneous, and semantically layered, while audiences exhibit diverse interests, prior knowledge, mobility constraints, and accessibility needs [12, 124]. In this setting, non-personalised or static information delivery can easily lead to cognitive overload, fragmented understanding, or dis-

engagement. Context-Aware Recommender Systems (CARS) offer a paradigm for addressing these issues by leveraging user behaviour, contextual variables, and environmental data to produce tailored recommendations [36]. However, several open challenges remain:

- how to model context in a way that is both expressive and manageable;
- how to integrate recommendations with digital storytelling to support coherent interpretative paths;
- how to evaluate the impact of contextual recommendations on visitors' learning, satisfaction, and overall cultural experience.

These open questions motivate the design, implementation, and assessment of novel CARS specifically tailored to CH scenarios.

The second motivation concerns the critical need for more robust and proactive strategies for the conservation and management of CH artefacts and structures. Degradation processes in historical materials are typically non-linear, multi-physics, and influenced by external factors that are difficult to measure or predict over long timescales [50]. Traditional numerical simulations, while powerful, often require simplifying assumptions and high-quality data, whereas purely data-driven models can struggle with scarce or noisy measurements. Recent advances in Scientific Machine Learning (SciML), and in particular Physics-Informed Neural Networks (PINNs), offer a middle path by incorporating physical laws directly into the learning process, allowing models to respect governing equations while adapting to observational data [50, 51]. However, there is a need to investigate how PINNs can be calibrated and validated on heritage artefacts, how they can interact with sparse sensor networks or monitoring data, and how they can be embedded within broader digital twin architectures to support predictive maintenance.

A further overarching motivation is to bring these two directions together within a single AI-enabled framework. Both contextual recommendation and predictive maintenance rely on structured digital representations of heritage assets, integration of multimodal sensor data, and the ability to reason over this information in a dynamic, context-aware manner [5, 12, 50]. Rather than developing separate ad hoc systems for visitors and conservation experts, this thesis advocates a unified framework that supports multiple AI-driven services, each tailored to specific objectives and user groups. To this end, the dissertation adopts a modular four-layer architecture comprising a Data Acquisition Layer, a Knowledge-Base Layer, an Inference Engine Layer, and an Application Layer. The Data Acquisition Layer ingests and harmonises heterogeneous streams from sensors and monitoring devices, user interactions, and external repositories. The Knowledge-Base Layer organises these heterogeneous inputs into semantically rich representations of assets, monitoring signals, and contextual information, enabling joint exploitation for both recommendation and maintenance tasks. On top of this, the Inference Engine Layer hosts specialised,

replaceable workflows that exploit the shared knowledge base while remaining functionally independent.
In Chapter 2, these workflows are instantiated as two complementary modules-built on the same shared data and knowledge backbone: a Cultural Experience Engine and a Predictive Maintenance Engine. Finally, the Application Layer exposes the resulting capabilities through personalised user services and expert-oriented decision-support tools. This modular design reconciles the need for a shared digital infrastructure with the flexibility of specialised, goal-oriented modules, enabling incremental extension and reuse of data and models across services.
Based on these considerations, this dissertation pursues three main objectives:

- *Unified framework for AI-enabled CH services*: to design and validate a modular four-layer architecture that supports multiple AI-driven functionalities, such as recommendation, prediction, and decision support, on top of a shared data and knowledge infrastructure for Cultural Heritage.
- *Enhancement of cultural enjoyment through contextual AI services*: to improve the way users access and experience Cultural Heritage by designing and implementing context-aware, user-centred recommender systems. These systems leverage the proposed framework to deliver adaptive cultural paths, personalised narratives, and context-sensitive suggestions that respect the specific constraints of CH environments.
- *Predictive maintenance of cultural assets through physics-based AI*: to support the long-term preservation and risk-informed management of CH artefacts and structures by developing and testing Physics-Informed Neural Network models on representative case studies. Within the proposed architecture, these models operate as inference modules that combine monitoring data with physical knowledge of degradation and structural behaviour, enabling more reliable predictive maintenance strategies within a unified digital pipeline.

Taken together, these objectives reflect the need to move beyond fragmented, single-purpose digital solutions towards an integrated, modular framework in which AI is deployed in a functional, goal-oriented manner to support both cultural enjoyment and conservation.

1.3 THE STRUCTURE OF THE THESIS

This dissertation is structured into six chapters that jointly support the main thesis of this work: Cultural Heritage requires a unified digital infrastructure able to serve heterogeneous stakeholders, from visitors seeking adaptive and engaging experiences to conservation experts requiring reliable, physically grounded predictive tools.
This first Chapter has framed the research context and motivates the two com-

plementary directions addressed in this dissertation: enhancement-oriented services for cultural users and expert-oriented workflows for predictive maintenance. It has also introduced the methodological background in AI, Machine Learning, Deep Learning, and Scientific Machine Learning, which will be instantiated throughout the remainder of the thesis.
Chapter 2 translates these motivations into a concrete, modular architecture. It presents the proposed four-layer framework comprising Data Acquisition, Knowledge-Base, Inference Engine, and Application, and clarifies how heterogeneous streams (sensor observations, user interactions, content descriptors, open data, and digital replicas) are harmonised and made reusable across services. The chapter also details the main engines and the services exposed to different user profiles, explicitly supporting both cultural experience and predictive maintenance within the same backbone.
Chapter 3 focuses on visitor-oriented enhancement and formalises the recommendation pipeline supported by the framework. After introducing the background and related work on recommender systems and context-aware recommendation, this chapter presents the notion of embedded context and the proposed Contextual Biases Matrix Factorization (CBMF) model. The chapter then introduces the Path Elaboration Module, which converts predicted preferences into feasible on-site itineraries, and reports an experimental assessment that includes both an offline evaluation and an in situ study during real Cultural Heritage visits.
Chapter 4 addresses expert-oriented predictive maintenance. It first discusses related work and introduces the two methodological pillars used in this thesis, Reduced Order Models and Physics-Informed Neural Networks, then describes how the framework processes complex 3D geometries through the 3D Model Module (via Blender APIs) to generate simulation-ready representations. The chapter reports numerical results for both inverse/offline-online components and direct-problem PINN workflows, and then validates the overall pipeline on the real case study of *Furia Selvaggia*, showing how geometry processing, network identification, and data acquisition from external sources support physically grounded corrosion simulations and qualitative comparison with real observations.
Chapter 5 isolates a key methodological limitation emerging in time-dependent physics-informed learning, the lack of explicit temporal causality enforcement in standard continuous-in-time PINNs, and introduces a step-by-step, time-discrete formulation. The proposed approach embeds classical one-step time-integration schemes (e.g., the Implicit Euler and Crank-Nicolson methods) into the training objective, thereby improving stability and long-horizon robustness. The chapter validates the method on a non-linear benchmark designed to highlight causality and temporal propagation effects.
Finally, Chapter 6 summarises the main contributions and discusses limitations

and future research directions. In particular, it highlights how the proposed framework enables reuse of data and digital replicas across enhancement and conservation services, and outlines extensions toward model discovery in the absence of established governing equations and toward Structural Health Monitoring workflows for historic buildings.

Chapter 2

The designed AI-based framework to the enhancement and maintenance of Cultural Heritage

Building on the motivations outlined in Chapter 1, this dissertation argues that Cultural Heritage requires digital solutions able to serve heterogeneous stakeholders, ranging from visitors seeking meaningful and adaptive experiences to conservation experts demanding reliable predictive tools [34, 35, 90]. The goal of this work is to support these user groups without fragmenting data, models, or development efforts. To address this need, we propose a unified AI-based framework that integrates contextual personalisation and predictive maintenance within a shared digital infrastructure, as illustrated in Figure 2.1. The framework is conceived as a modular four-layer architecture that acquires and harmonises heterogeneous data streams, organises them into semantically structured knowledge representations, and deploys replaceable inference modules to support multiple services [123]. In particular, the proposed design aims to:

- promote reuse of data and models across visitor- and conservation-oriented applications;
- ensure extensibility, so that new AI components and data sources can be added incrementally;
- provide a clear separation of concerns across data management, knowledge representation, inference, and user-facing functionalities.

This chapter first introduces the framework at a high level and then details its layers, their responsibilities, and the interfaces that enable interoperability

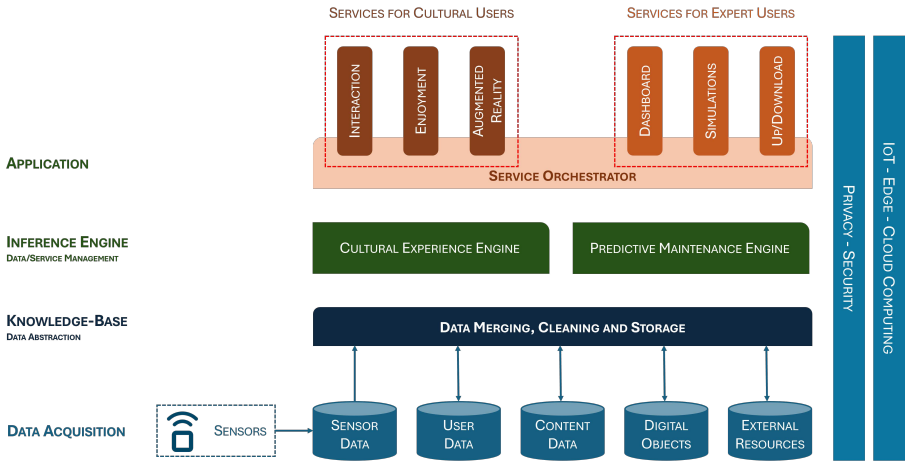


Figure 2.1. A high-level overview of the proposed framework for enhancing cultural heritage user engagement and supporting predictive maintenance of cultural artefacts.

among components. The Data Acquisition Layer is presented as the entry point for sensor streams, user interactions, cultural content data, digital objects, and external resources. The Knowledge-Base Layer is then described as the storage and integration backbone, organising assets, context, and time-dependent observations into structured representations that can be reused across services. Next, the Inference Engine Layer is presented as a container for AI and reasoning modules, including both context-aware recommendation and physics-informed predictive models. Finally, the Application Layer is discussed in terms of the services exposed to different user profiles, including adaptive visitor experiences and expert-oriented decision-support tools.

A key advantage of this architecture is its modularity, which supports independent and specialised processing pipelines while preserving a shared view of the underlying data and knowledge. Inference modules can be developed, tested, and maintained in isolation without requiring changes to the other layers, reducing coupling and facilitating iterative improvements. Moreover, the framework does not assume that all components must be active at all times; instead, modules can be selectively activated based on the operational context, available data, and the goals of the current user session. This on-demand activation improves computational efficiency and scalability by allocating resources only to the analyses required at a given time. At the same time, hosting separate inference pipelines within the same architectural backbone encourages reciprocal exchange between services [57], reconciling specialisation with reuse.

To make these design choices explicit, the next section moves from the conceptual overview to a concrete architectural description of the framework.

2.1 The proposed framework

This section provides a concrete description of the proposed framework, clarifying how its components collaborate to deliver two families of services, context-aware visitor support and expert-oriented predictive maintenance, on a shared digital infrastructure. Rather than separating these goals into independent systems, the framework is designed to maximise reuse of data, knowledge representations, and AI modules, while clearly separating responsibilities and keeping components replaceable.

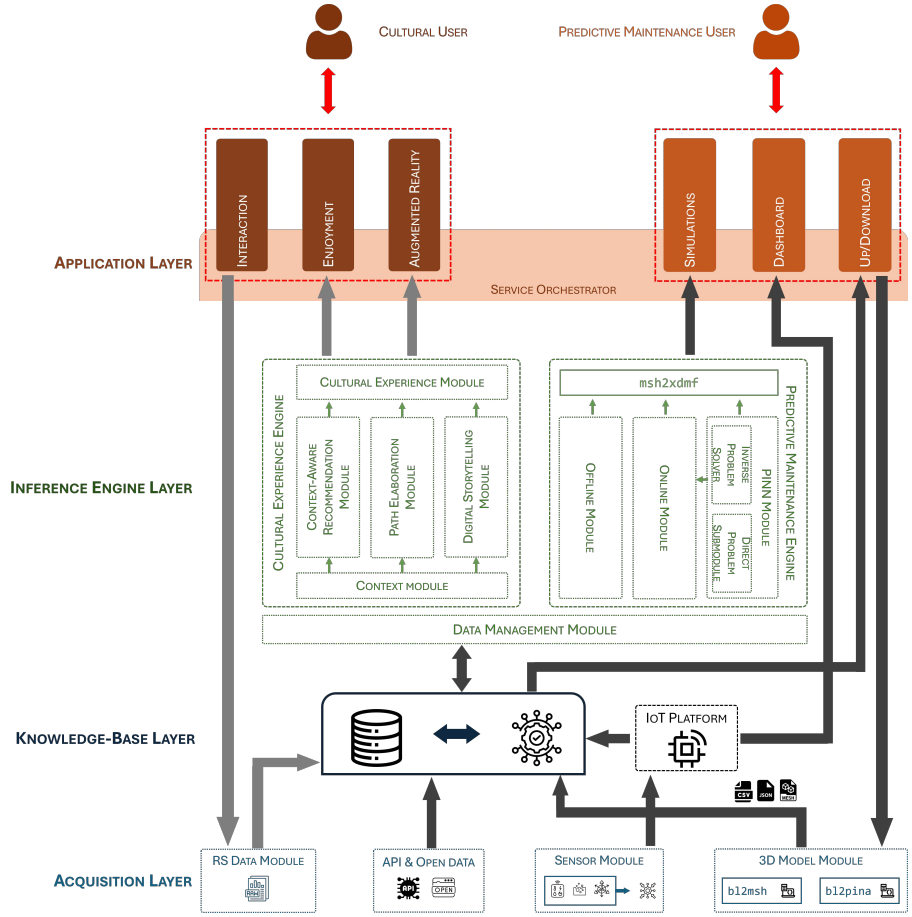


Figure 2.2. The proposed framework to enhance cultural users' experiences and support conservation and restoration experts through AI-based predictive maintenance.

The framework, shown in Figure 2.2, is articulated into four layers: the Data Acquisition Layer, the Knowledge-Base Layer, the Inference Engine Layer, and the Application Layer. Each layer exposes well-defined interfaces to the others, enabling heterogeneous inputs to be harmonised into a common representation

and then exploited by specialised inference modules. This layered organisation ensures extensibility, allowing new sources, models, or services to be added incrementally, and supports interoperability across the full stack, from data governance to user-facing functionality [57].

To enhance cultural experiences, the framework provides context-aware guidance before and during the visit through conversational and immersive modalities [76, 124]. Recommendations can be adapted to real-time signals such as location, weather conditions, and crowding, and combined with cultural content and user preferences to generate tailored routes and narrative experiences [59, 184]. Importantly, this personalisation logic does not operate on a separate dataset: it relies on the same knowledge core (i.e., the Knowledge-Base Layer) used by expert services, ensuring consistency of content, provenance, and contextual metadata.

From a predictive maintenance perspective, the same infrastructure supports the ingestion of digital replicas of cultural artefacts and time-dependent monitoring data [112]. Within this setting, experts can use physics-based and physics-informed analyses to assess the condition of cultural artefacts and forecast their evolution under alternative scenarios [103]. Outputs are made available as inspection-oriented results, including structured reports and interactive 3D visual analytics, supporting reproducible workflows and controlled data exchange for collaboration and refinement.

Two cross-cutting mechanisms contribute to making these services reliable and scalable. First, orchestration governs authentication, permissions, data access, and execution flows across layers (implemented at the application level through a dedicated Service Orchestrator), keeping user-facing interactions responsive while allowing expert computations to scale from lightweight online surrogates to more intensive offline training and calibration procedures. Second, feedback signals, such as user interaction traces and expert validations, are treated as first-class inputs: they are filtered and aggregated into the knowledge core to update user models, validate assumptions, and progressively improve both recommendation strategies and maintenance pipelines.

In the remainder of this section, the framework is described layer by layer. We first present the Data Acquisition Layer as the entry point for heterogeneous sources, then the Knowledge-Base Layer as the organising backbone for assets, context, and observations. Next, we detail the Inference Engine Layer as a container for replaceable AI and reasoning modules, and finally, the Application Layer in terms of the services exposed to different user profiles and interaction modalities.

2.1.1 *Data Acquisition Layer*

The Data Acquisition Layer is the entry point of the framework and is responsible for collecting, preparing, and forwarding heterogeneous inputs to both the cultural experience and predictive maintenance pipelines. Its main objective is to make diverse sources interoperable early in the stack, so that the remaining components can access consistent and timely information without duplicating ingestion logic. The layer is organised into four modules, Sensor Module, RS Data Module, Open Data & API Module, and 3D Model Module, which operate as parallel acquisition channels converging into the Knowledge-Base Layer. There is no strict hierarchy among these modules: they are complementary sources whose outputs are integrated downstream. In particular, the Knowledge-Base Layer receives:

- time-dependent observations from the Sensor Module (via the IoT platform and REST APIs),
- entities and content descriptors from the RS Data Module,
- external contextual variables from the Open Data & API Module,
- analysis-ready geometry descriptors and derived artefacts from the 3D Model Module.

This organisation keeps ingestion concerns separate while ensuring that both visitor-facing and expert-facing services operate on a shared, consistent representation.

Sensor Module

The Sensor Module manages the acquisition of real-time measurements through IoT devices deployed in the environment and/or on cultural assets. It is event-oriented: data are produced as updates triggered by sensor readings and are transmitted by local gateways (centralisers) that synchronise signals, apply lightweight preprocessing when needed, and forward observations to an IoT platform (e.g., ThingsBoard) in a key-value form [95, 115]. On the experience side, sensor data support context-aware personalisation. Measurements such as weather conditions, proximity to points of interest, and crowding levels directly affect the suitability of recommendations and the feasibility of suggested routes. In practice, crowding can be estimated using different sensing strategies, depending on the site (e.g., camera-based counting for outdoor installations or entry/exit sensing for indoor environments). User presence and proximity can be captured to infer location-dependent context and recurrent attendance patterns. When computationally feasible, edge devices can perform initial preprocessing to reduce noise and bandwidth usage and derive higher-level indicators. An example of the sensor topology integrated with local gateways is reported in Figure 2.3.

On the predictive maintenance side, the same acquisition logic supports mon-

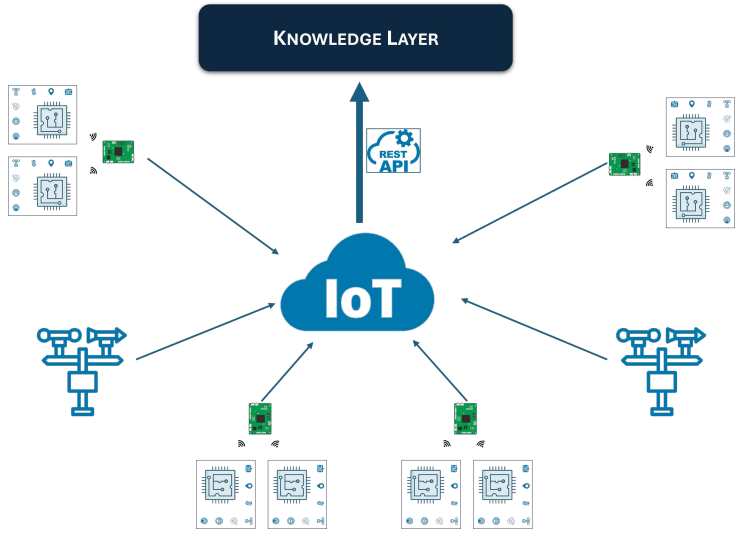


Figure 2.3. Sensor topology and gateways used to acquire environmental data and forward them to the IoT platform.

itoring environmental conditions and physical processes relevant to asset conservation. The resulting time-stamped observations are persisted and made available to inference components for assessment and forecasting. In both cases, the Sensor Module outputs a stream of harmonised observations to the IoT platform, which exposes them to the Knowledge-Base Layer through REST APIs, enabling storage, visualisation, and downstream consumption [60].

RS DATA MODULE

The RS Data Module collects and maintains the information needed by recommendation and personalisation services, including user profiles and preferences, metadata for points of interest, and curated digital content associated with each POI [124]. Unlike the Sensor Module, which focuses on real-time signals, this module primarily manages structured and editorial data, as well as user-generated data derived from interactions [85]. Its output is a consistent set of entities and relations (users-POIs-content) that can be combined with real-time context to generate tailored cultural paths and narrative experiences [34].

OPEN DATA & API

The Open Data & API integrates information from external providers through REST services [60]. Its role is to enrich both pipelines with contextual variables that may not be directly measured on site or that are better sourced from authoritative services [33]. This module communicates directly with the Knowledge-Base Layer, contributing additional context that can:

- improve recommendation relevance,

- support conservation analyses by providing data related to the analysed physical phenomena.

3D Model Module

The 3D Model Module manages digital replicas of cultural assets uploaded by authorised users and prepares them for computational analysis. It performs ingestion, validation, and preprocessing steps that transform a raw 3D model into analysis-ready artefacts consumed by the Knowledge-Base Layer and, indirectly, by the Inference Engine [159]. Concretely, the module extracts geometry-related descriptors and derived assets for SciML workflows (e.g., mesh-related data and sampling-support structures), and can also generate visualisation-ready outputs to support inspection and reporting. While this module is central to the predictive maintenance pipeline, it also provides a geometry backbone that can be reused by experience-oriented applications whenever 3D assets are leveraged for immersive visualisation. A detailed description and validation of this module is provided in Section 4.3.

2.1.2 Knowledge-Base Layer

The Knowledge-Base Layer is the framework's core for storage and integration. It provides the common substrate on which both service families operate: cultural experience personalisation and predictive maintenance. Its responsibility is not limited to persistence; it also performs integration and quality control to ensure consistency, traceability, and readiness for downstream inference.
The layer receives inputs through four coordinated ingestion paths. The first path concerns IoT measurements: sensor observations are collected by local centralisers and published to the IoT platform (ThingsBoard), which manages device-level streaming and exposes REST endpoints. The Knowledge-Base Layer ingests these measurements via REST API calls, storing time-stamped observations together with the metadata required for interpretation. The second path targets external contextual information: data acquired by the Open Data & API component is ingested through dedicated REST connectors. The third path concerns digital replicas and analysis assets: the 3D Model Module provides geometry-related descriptors and derived artefacts that enable computational use of the model. The fourth path aggregates experience-related information: structured and semi-structured data describing users, POIs, curated cultural content, and interaction traces are persisted so that visitor-facing services can combine real-time context with stable content and evolving user models.
To ensure reliability, the layer includes a pre-processing module that acts as a quality gate before storage. This module checks the completeness and consistency of incoming streams, detects missing data and time-series gaps, and harmonises fields across sources. When feasible, it applies lightweight imputation

or aggregation policies; however, its primary goal is to produce well-formed, explicitly annotated data so that subsequent modules can operate on validated inputs.
In addition, the Knowledge-Base Layer supports the export of selected datasets and artefact models for expert users through the Upload/Download services exposed at the application level.

2.1.3 Inference Engine Layer

The Inference Engine Layer is the computational backbone of the framework. It transforms the integrated, quality-controlled information stored in the Knowledge-Base into actionable outputs for two categories of users:

- cultural users, who require adaptive guidance and engaging narrative experiences;
- restoration experts, who need reliable simulations and predictive insights to support assessment and decision-making.

In operational terms, the layer consumes context streams and user–POI interactions, curated cultural content, monitoring data acquired through the IoT platform and stored in the Knowledge-Base, and digital replicas and derived geometric artefacts produced by the 3D Model Module. It produces ranked recommendations and adaptive paths, narrative structures and content packages for delivery via the application services, and simulation/prediction outputs, identified parameters, and inspection-ready visual products over the asset geometry.
To ensure robust interoperability, a Data Management Module serves as a mediator between the Knowledge-Base and the engines described below. This module standardizes data access and formatting, selects the appropriate datasets and features for each task, and manages the retrieval and persistence of computational artefacts. In this way, the two engines operate on consistent inputs and generate outputs that remain traceable and reusable across sessions and experiments.
The layer is organized into two main subsystems: the Cultural Experience Engine and the Predictive Maintenance Engine.

Cultural Experience Engine

The Cultural Experience Engine, reported in Figure 2.4, makes operational adaptive cultural support by combining real-time context, user-related signals, and curated heritage content.
Its objective is not only to recommend points of interest (POIs), but also to assemble a coherent, feasible experience that can be delivered through user-facing services. The engine is articulated into four modules, each addressing a specific stage of the experience generation pipeline.

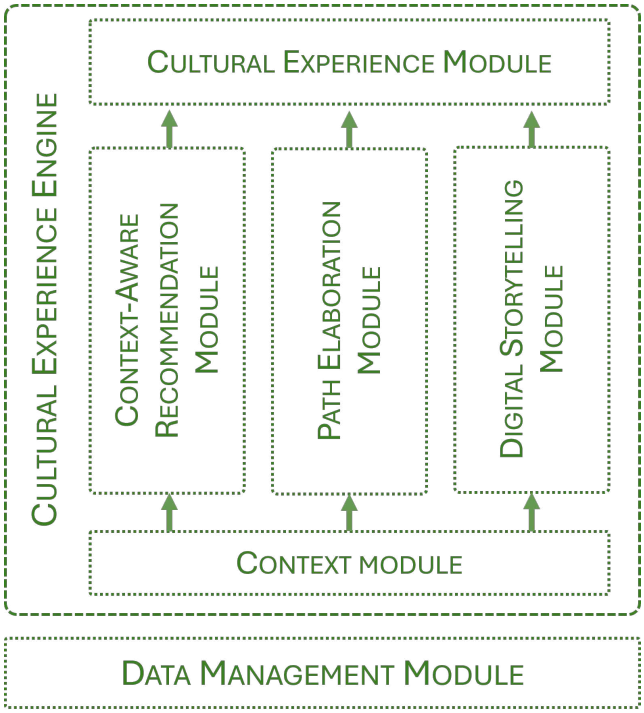


Figure 2.4. Focus on the Cultural Experience Engine of the proposed framework introduced in Figure 2.2.

Context Module

The Context Module integrates and prepares contextual signals required by the downstream modules. It receives heterogeneous context streams through the Data Management Module and applies normalisation and discretisation policies to obtain well-defined contextual values. Then, it constructs the embedded context identifier by combining the concurrent contextual values into a single discrete contextual state (i.e., a tuple-based identifier representing the current visit conditions). When contextual conditions change during the visit, the module updates the embedded context accordingly and triggers downstream recomputation when required.

Specifically, the management of the context leverages the Context Dimension Tree (CDT) [30], a directed acyclic graph that allows representing and identifying context from data. The Context Dimension Tree G_{CDT} consists of the nodes' set N_{CDT}, the edges' set E_{CDT}, and the root r_{CDT}

$$G_{CDT} =< N_{CDT}, E_{CDT}, r_{CDT} >, \tag{2.1}$$

where the nodes' set N_{CDT} is the union of three disjointed sets

$$N_{CDT} = N_d \cup N_c \cup N_p, \qquad N_d \cap N_c, N_d \cap N_p, N_c \cap N_p = \emptyset. \tag{2.2}$$

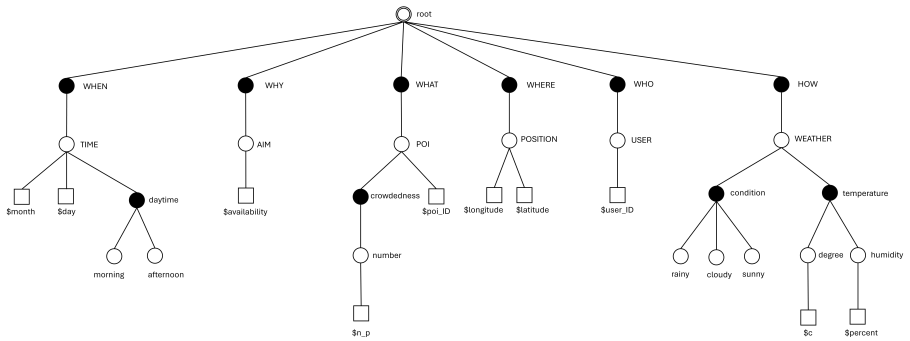


Figure 2.5. The Context Dimension Tree (CDT) designed for the analysed case study of Chapter 3.

The sets N_d, N_c, and N_p represent the dimension nodes, concept nodes, and parameters, respectively. Specifically, the dimension nodes describe the context according to the 5W+1H model [93]. The 5W+1H model enables the structuring of contextual data along six key dimensions: *who* is involved, *what* is the object of interaction,*where* and *when* the interaction takes place, *why* it occurs, and *how* it unfolds [93]. These dimensions collectively contribute to the development of a rich, multi-faceted representation of context. Specifically, in the case study related to Chapter 3, the employed CDT is shown in Figure 2.5. In the Figure, the dimension nodes are represented as black circular nodes, the concept nodes as white circular nodes, and the parameters as squared white nodes. The dimension nodes of the first level in the tree represent the 5W+1H model. The *when* dimension is related to the time context (the concept node TIME), specifying the month and the day of the visit, and the dimension node daytime, to which morning and afternoon are associated. The *why* dimension is related to the visit aim (the concept node AIM), which affects the user's visit modality. Specifically, based on the objective and motivation behind the visit, the user's presence in the archaeological park will have a defined duration. The *what* dimension node is related to the POI concept node, to which the poi_ID is associated. Additionally, we consider the dimension of node crowdedness in terms of the number of people present at the POI. The *where* dimension node is connected to the concept node POSITION, returning the parameters $longitude and $latitude. The *who* dimension node is related to the concept node USER, which is related to the user_ID. Finally, the *how* dimension node has the concept node WEATHER, which is related to two dimension nodes: condition and temperature. The first returns the weather conditions (rainy, cloudy, or sunny), and the second returns the temperature degree and humidity. Through the CDT, the Context Module can identify the context, then provides the embedded context value to elaborate the recommendation in the Context-

Aware Recommendation Module.

Context-Aware Recommendation Module

This module implements a machine-learning–driven context-aware recommender system that estimates the relevance of candidate POIs for a given user under the current visit conditions. The model learns latent user and item representations from historical interaction data, while explicitly adapting predictions to context. Rather than treating each contextual dimension independently, the module adopts an embedded context strategy. In this formulation, multiple contextual data (weather, crowding, and distance) are combined into a single discrete contextual state, i.e., an embedded context identifier representing a tuple of concurrent conditions. The learning process then estimates user–POI affinity in the latent space and adjusts it through context-dependent terms learned over these embedded states.
Operationally, the module receives, via the Data Management Module, the user–POI interaction history, POI descriptors, and the current contextual information; it outputs a ranked list of POIs, together with relevance scores calibrated to the prevailing situation. This design preserves the expressiveness of multi-dimensional context while maintaining computational efficiency, enabling recommendations to be updated at runtime as contextual conditions evolve. Further methodological details of the proposed context-aware recommendation approach are discussed in Section 3.3.

Path Elaboration Module

Recommendations are then converted into an adaptive itinerary. The Path Elaboration Module constructs the visit as an incremental decision process, selecting the next POI step by step based on the user's current position, estimated travel time, remaining visit duration, and dynamic context updates. This component, therefore, enforces feasibility while preserving personalization, ensuring that the suggested experience remains compatible with real-world constraints by employing a Linear Integer Programming Model. A detailed description of the adopted path optimization model is provided in Section 3.4.

Digital Storytelling Module

Given the POIs selected and ordered by the preceding modules, the Digital Storytelling Module orchestrates narrative assets and multimedia content associated with each POI. It supports non-linear storytelling by selecting and sequencing story fragments based on thematic coherence, user interests, and context, thereby increasing engagement without disconnecting narration from the underlying cultural knowledge [129]. The resulting storytelling package can be exposed through conversational interaction and immersive AR experiences at the application level.

Cultural Experience Module

The Cultural Experience Module combines the outputs of the previous three modules into a single, user-ready experience. It combines the recommended POI set, adaptive path, and narrative content plan into a single proposal for the Application Layer. This module resolves any inconsistencies between storytelling and path feasibility. It produces the final visit representation, such as an ordered POI itinerary with storytelling and interaction hooks. The module also allows updates when the context changes during the visit.

Predictive Maintenance Engine

The Predictive Maintenance Engine, shown in Figure 2.6 and composed of four modules, enables expert-oriented analyses by integrating monitoring data, digital replicas, and physical knowledge of the underlying phenomena.

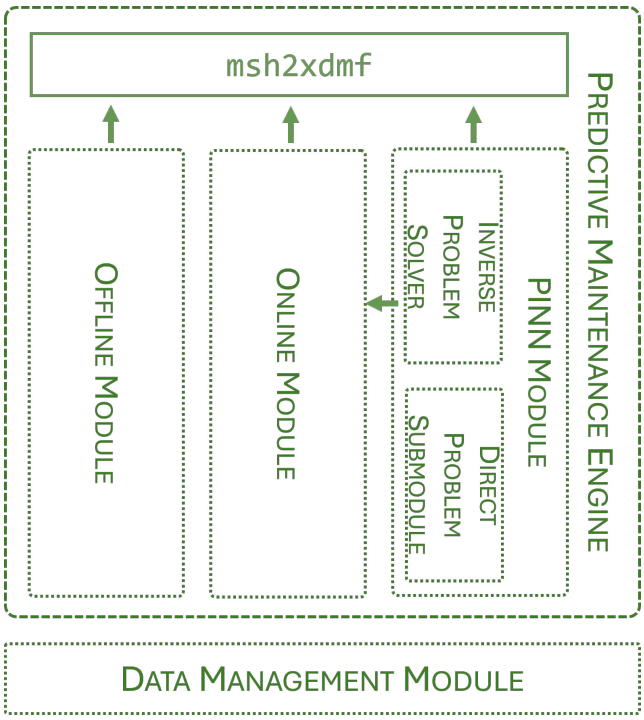


Figure 2.6. Focus on the Predictive Maintenance Engine of the proposed framework introduced in Figure 2.2.

Its objective is to provide results that are both scientifically grounded and operationally usable, balancing accuracy and computational efficiency through a structured combination of offline computation, online rapid evaluation, and physics-informed learning. A comprehensive discussion of the behaviour of the Predictive Maintenance Engine is provided in Chapter 4.

Offline Module

This module executes the offline phase required by reduced-order modeling workflows. It includes a FEM Submodule, which computes full-order solutions of the governing differential problem for selected parameter configurations (snapshots) and extracts the reduced basis, via Proper Orthogonal Decomposition. It also includes a ROM-Saving component that stores the reduced basis, snapshot solutions, and associated metadata in the Knowledge-Base through the Data Management Module, ensuring reusability, traceability, and reproducibility across analyses and enabling the online phase.

Online Module

The Online Module provides rapid responses after building the reduced basis. The ROM Loading component retrieves the needed reduced artefacts from the Data Management Module. The Online Solver Submodule performs the reduced solve and reconstructs the solution in the original space. This stage allows interactive or near-real-time scenario exploration when repeated full-order simulations are not possible.

PINN Module

The PINN Module draws on outputs from the Online Module and integrates physical constraints with observed data to address both inverse and direct tasks. The Inverse Problem Solver estimates unknown parameters provided to the Online Module by fitting the model to monitoring data under physics-informed constraints. The Direct Problem Submodule (Direct Problem) solve the forward problem, producing predictions consistent with both physics and data.

Msh2xdmf

This component connects predicted fields to the asset geometry. It combines the processed 3D model with outputs from ROM and PINN components. The result is visualization-ready products, such as field mappings in formats suitable for inspection and reporting. This lets experts directly interpret predictive outputs through interactive 3D analytics and inspection-oriented views.

2.1.4 Application Layer

The Application Layer provides the operational interface between the framework and its end users. It exposes user-facing services, manages interaction flows, and delivers outputs generated by the Inference Engine in usable forms. In addition, it captures interaction traces and explicit preference signals and forwards them to the Knowledge-Base Layer and the RS Data Module to progressively refine user models and personalisation strategies.
A key architectural element of this layer is the Service Orchestrator, which

coordinates service execution and mediates the interaction with the Inference Engine. The orchestrator manages authentication and permissions, routes requests to the appropriate engine (Cultural Experience Engine or Predictive Maintenance Engine), sequences multi-step workflows, and supports differentiated execution modes. In this way, user-facing interactions remain responsive while computation-intensive tasks can be scheduled and controlled without exposing internal complexity to the end user. Moreover, orchestration supports traceability of requests and results by maintaining a consistent association between executed services, consumed data, and produced outputs.
The services exposed by the Application Layer are organised into two target user groups: Cultural Users and Predictive Maintenance Users.

Services for Cultural Users

Services for Cultural Users are designed to support visitors before and during the visit by delivering adaptive guidance and engaging narrative experiences. They are implemented through three complementary services: Interaction, Enjoyment, and Augmented Reality.

INTERACTION SERVICE

The Interaction Service provides the primary access point to the system through conversational and guided interfaces [38]. It enables users to request contextualised information, obtain explanations of suggested POIs, and receive navigational support to the next step in the recommended path. Preference elicitation and cold-start mitigation are integrated directly into this service: the system can collect explicit signals via short questionnaires and implicit signals from interaction traces. These signals are persisted in the Knowledge-Base Layer and made available through the RS Data Module, enabling continuous refinement of the user profile and improved subsequent recommendations.

ENJOYMENT SERVICE

The Enjoyment Service focuses on delivering the assembled cultural experience produced upstream by the Cultural Experience Engine. It presents the recommended itinerary and the associated storytelling plan in a user-consumable form, ensuring coherence across POIs, narrative units, and visit constraints. This service exploits non-linear digital storytelling by selecting and presenting multimedia fragments, thematic connections, and contextual explanations that enhance engagement while remaining consistent with the curated knowledge stored in the Knowledge-Base Layer.

AUGMENTED REALITY SERVICE

The Augmented Reality (AR) Service provides immersive, in situ delivery of selected storytelling assets when the visit context and available content make AR appropriate. AR acts as a high-engagement channel that complements the

Interaction and Enjoyment services, grounding narrative content in the physical environment and increasing interpretability [27].

Services for Predictive Maintenance Users

Services for Predictive Maintenance Users support conservation experts in monitoring, analysis, and data governance tasks. They are organised into Simulation, Dashboard, and Upload/Download services, which collectively expose the Predictive Maintenance Engine's outputs and enable controlled workflows for digital assets.

Simulation Service

The Simulation Service provides access to prediction and simulation products generated by the Predictive Maintenance Engine. It supports the visualisation and inspection of computed fields over the asset geometry, as produced by the engine post-processing pipeline, and enables the exploration of alternative scenarios when multiple runs are available. The service is designed to deliver results in an inspection-ready form, suitable for expert interpretation and reporting.

Dashboard Service

The Dashboard Service offers an operational overview of monitored assets by exposing sensor streams and derived indicators through the IoT platform. It enables experts to explore trends, filter by asset and time range, and identify conditions that warrant deeper analysis.

Upload/Download Service

The Upload/Download Service provides a governed channel for exchanging inputs and outputs with the framework. It allows experts to upload digital replicas (3D models) and associated resources for preprocessing and analysis, and to download selected datasets as well as models or meshes of artefacts stored in the Knowledge-Base Layer.

Chapter 3

Enhancing Cultural Heritage and improving cultural experiences

This chapter shifts the focus from the architectural perspective introduced in Chapter 2 to the methodological core of the Cultural Heritage enhancement pipeline, with specific emphasis on improving cultural users' experiences. In particular, it explains how contextual signals and user feedback are transformed into actionable outputs that support adaptive visits. The goal is to make the recommendation workflow explicit: from modelling context and preferences, to estimating personalised relevance, and finally to generating feasible visit paths that remain consistent with real-world constraints.

The starting point is the observation that Cultural Heritage visits are inherently context-dependent [108]. The same Point of Interest (POI) may be perceived differently depending on time, weather, crowd levels, and mobility needs [23, 53, 125, 147]. Moreover, visitors are heterogeneous: some seek highlights and short routes, whereas others prefer thematic depth, quiet areas, or accessibility-aware itineraries. Addressing this variability requires recommendation models that treat context not as an optional refinement, but as a core component of the decision process [85].

To this end, the chapter introduces the notion of *embedded context*, a representation strategy that models context as a coherent configuration rather than as independent variables evaluated one at a time. Instead of modelling *weather*, *crowdedness*, and *distance* separately, the embedded context encodes the specific combination of these factors that co-occur during the user's experience. Building on this concept, the chapter presents Contextual Biases Matrix Factorisation (CBMF), a context-aware recommendation approach that extends matrix factorisation by integrating contextual biases while preserving a compact parametrisation. This design supports fast updates and makes the method

practical in settings where recommendations must adapt quickly to changing conditions and continuously arriving data.
The methodological discussion is complemented by a quantitative evaluation. The chapter reports numerical experiments aimed at validating the predictive accuracy and robustness of CBMF over several literature datasets. In addition, using a dataset of user evaluations of POIs collected in real Cultural Heritage scenarios, the experiments compare CBMF against representative baselines from the state of the art and analyse how embedded context modelling affects both accuracy and stability across contexts [37]. The evaluation is designed not only to measure performance, but also to clarify which contextual configurations most influence user judgement and where the proposed modelling choice provides measurable advantages.
Accurate relevance estimation, however, is only one part of the problem: cultural recommendations must be transformed into on-site visit routes. For this reason, the chapter describes the Path Elaboration Module introduced in the previous chapter, which converts predicted preferences into personalised paths by combining recommendation scores with spatial and operational constraints. The module enables the system to propose routes that are both meaningful and realistically actionable within the dynamics of archaeological parks and museum spaces.
Finally, the chapter presents the prototype implementation of the proposed approach and discusses the results of an in-situ experimentation conducted with a large number of visitors in real contexts.

3.1 BACKGROUND

Recommender Systems (RSs) are information filtering and decision-support tools that provide users with personalised suggestions of relevant items.
The main entities involved in an RS are users, items (e.g., services, objects, or points of interest), and transactions, which represent interactions between users and items. A common form of transaction is an explicit rating, i.e., a user's judgment about a specific item. Ratings can be expressed on an ordinal scale (e.g., strongly agree, agree, neutral, disagree, strongly disagree), on a numerical scale (e.g., 1-5 or 1-10), or as binary feedback (like/dislike). In some settings, feedback is unary (one-class): only positive interactions are observed, while the remaining user–item pairs are treated as unknown.
Formally, let U be the set of users and I the set of items. Let $K \subseteq U \times I$ denote the set of observed user-item pairs. The rating or utility function is defined as a mapping $r : K \to \mathbb{R}$ that assigns a score r_{ui} to each observed pair $(u, i) \in K$:

$$r : (u, i) \in K \longmapsto r(u, i) = r_{ui} \in \mathbb{R}. \tag{3.1}$$

Since r is not observed for all pairs in $U \times I$, one of the main goals of a

recommender system is to estimate the missing values, producing predicted ratings $\hat{r}_{ui}$ for pairs $(u, i) \in (U \times I) \setminus K$.

3.1.1 Classification of Recommender Systems

The ability to generate reliable rating forecasts is a key factor that distinguishes effective recommender systems from ineffective ones, and it is therefore central when evaluating recommendation techniques.
RSs can be classified according to how forecasts are generated. The three most common strategies are:

- **Content-Based RS**: this technique exploits the feature vector $\mathbf{y} = (y_1, \ldots, y_d) \in \mathbb{R}^d$ associated with an item i (Content Analyzer) and creates the feature vector $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ representing user preferences for user u (Profile Learner) [55, 117]. To assess whether item i is suitable for user u, the Filtering Component applies similarity metrics between $\mathbf{x}$ and $\mathbf{y}$, such as cosine similarity [55, 117]:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\sum_{t=1}^{d} x_t y_t}{\sqrt{\sum_{t=1}^{d} x_t^2}\sqrt{\sum_{t=1}^{d} y_t^2}} \tag{3.2}$$

- **Collaborative Filtering RS:** this strategy generates rating forecasts by exploiting collective information from user-item interactions, e.g., leveraging the opinions/ratings of similar users or items [45]. Collaborative systems are commonly divided into two groups:
  - **Memory-Based Collaborative Filtering:** users or items are grouped into neighbourhoods in order to generate rating predictions through similarity coefficients and known ratings [87];
  - **Model-Based Collaborative Filtering:** a predictive model is learned from known ratings, often through matrix factorisation [87]. The most commonly used decomposition techniques include:
    - Principal Component Analysis [29];
    - Probabilistic Matrix Factorization [29, 144];
    - Non-Negative Matrix Factorization [29];
    - Singular Value Decomposition [62, 132];
- **Hybrid RS:** hybrid approaches combine two or more techniques to improve prediction quality. The techniques can be applied separately and then combined, or integrated within a single model that merges the properties of the chosen strategies [137].

3.1.2 The limits of Recommender Systems

The main problems faced by recommender systems are [29, 137, 164]:

- **Scalability:** the system's ability to cope with an increasing amount of data (e.g., users, items, and interactions);
- **Sparsity:** the limited number of known ratings, which may negatively affect the quality of the generated forecasts;
- **Cold Start:** the difficulty in producing forecasts for new users and/or new items with insufficient interaction history.

Table 3.1 provides a detailed description of the advantages and disadvantages of the recommendation techniques discussed above.

Table 3.1. Limits of Recommendation techniques.

Recommendation Technique		**Advantages**	**Limitations**
CONTENT-BASED		Easiness in suggesting new item Easiness of implementation	Cold start (new user) Diversity
COLLABORATIVE FILTERING	Memory-Based	Easiness of data updating Easiness of implementation	Cold start (new user /new item) Sparsity Scalability
	Model-Based	Compares well with sparcity e scalability The resulting prediction performance is better	Cold start (new user/new item) Loss of information because of the use of factorization techniques
HYBRID RS		Provides better suggestions Overcomes the limitation of individual techniques	Complexity Model Development Cost

3.1.3 CONTEXT-AWARE RECOMMENDER SYSTEMS

Over the years, Context-Aware Recommender Systems (CARS) have been introduced to produce more reliable rating forecasts by explicitly accounting for the conditions under which interactions occur. The term *context* does not admit a single definition, as it is used across several research areas. For instance [4]:

- **Data Mining**, where context denotes events or states that characterise a user and may influence preference variations;

- **E-commerce Personalisation**, where context helps identify user behaviour patterns related to purchasing intentions;
- **Mobile Context-Aware Systems**, where context includes information accessible via mobile devices (e.g., location and nearby people or places of interest);
- **Marketing and Management**, where contextual information supports situation-aware strategies tailored to customer needs.

The difficulty of finding a specific purpose is expressed by the words of Bazire and Brezillon [22, 4]:

... it is difficult to find a relevant definition satisfying in any discipline. Is context a frame for a given object? Is it the set of elements that have any influence on the object? Is it possible to define context a priori or state the effects of a posteriori? Is it something static or dynamic?

To clarify the concept, some representative definitions adopted in the recommender systems literature are reported below:

- **C. K. Prahalad:** the ability to reach customers anywhere and at any time requires delivering real-time experiences shaped by customer context [4];
- **Abowd et al.:** any information useful to characterise the situation of an entity that can affect the way users interact with systems [1];
- **Schilit et al.:** location, nearby people and things, and the changes that occur to them [154];
- **Dey:** examples include emotion, focus of attention, location, adaptation, date and time, nearby things, and nearby people [154, 56].

In the RS domain, contextual information can therefore be viewed as modulating user-item interactions, potentially changing the observed rating depending on the situation. As contextual conditions vary, the same user may evaluate the same item differently. Introducing context into the recommendation process naturally extends the utility function.

Definition 3.1. *Let U be the set of users, I the set of items, and C the set of possible contextual states (or contextual configurations). Let $K \subseteq U \times I \times C$ denote the set of observed triples. The rating (utility) function for CARS is defined as a mapping $f : K \to \mathbb{R}$ that assigns a score f_{uic} to each observed triple $(u, i, c) \in K$:*

$$f : (u, i, c) \in K \longmapsto f(u, i, c) = f_{uic} \in \mathbb{R}. \tag{3.3}$$

The introduction of CARS highlights the importance of acquiring contextual information. Context can be obtained in different ways:

- **explicitly**: by directly asking the user (e.g., questionnaires) or through other explicit input mechanisms [137];

- **implicitly**: via devices and sensors (e.g., location, temporal data, weather data) or by observing changes in the user's environmental conditions [137];
- **through inference**: by statistical methods or data mining techniques that derive context from available signals.

Classification of Contextual Information

The acquisition of contextual information necessitates schematisation and management strategies. Equally important is the selection of relevant contextual variables: only context that meaningfully affects ratings is expected to improve recommendation performance [3].
A comprehensive classification is provided by Villegas et al. [170]. Here, the term *entity* refers to the main elements of CARS, namely users and items. Contextual information can be classified as follows:

- **Individual Context:** contextual information associated with independent system entities sharing common characteristics. It can be further subdivided into:
  - **Natural:** characteristics acquired without human intervention (e.g., atmospheric conditions);
  - **Human:** information associated with a user's behaviour or preferences;
  - **Artificial:** information derived from human actions or technical processes describing entities;
  - **Groups of Entities:** information that aggregates entities through shared features.
- **Location Context:** information associated with the location of an entity, which can be:
  - **Physical:** geographical coordinates and spatial positioning;
  - **Virtual:** digital location information (e.g., IP address).
- **Time Context:** information related to physical time associated with an entity's activity, which can be:
  - **Defined:** when the time interval is known through explicit start and end points;
  - **Indefinite:** when the occurrence is relative to other events or the duration is not specified.
- **Activity Context:** information related to activities performed by system entities, which can support prediction of future preferences or behaviours;
- **Relational Context:** information related to relationships developed between entities under contingent circumstances. It can be further subdivided into:
  - **Social:** interpersonal relationships, affiliations, associations, etc.;
  - **Functional:** usage relationships among entities (how some entities

make use of others).

Classification of Context-Aware Recommender Systems: integration of the context in RS

Once contextual information has been acquired and classified, it must be integrated into the recommendation process.

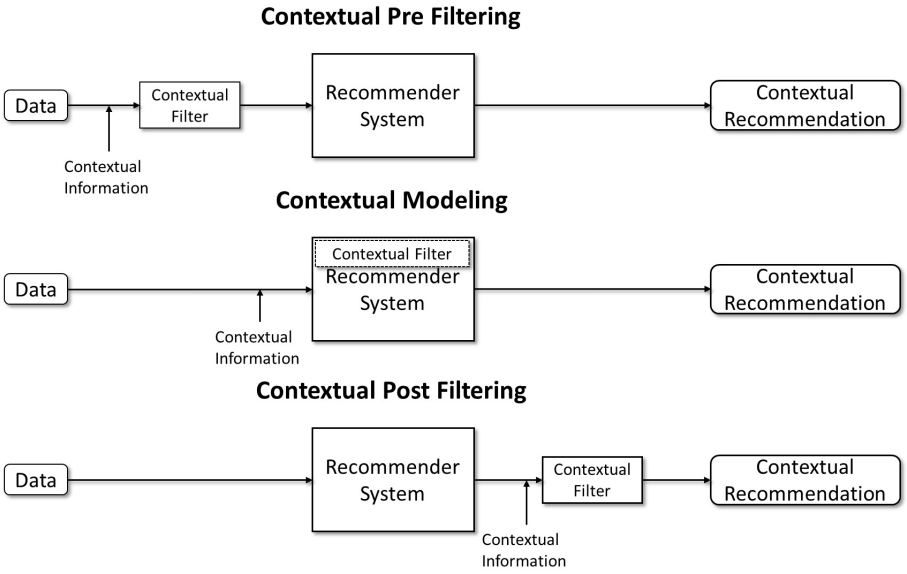


Figure 3.1. Approaches to integrate context within the recommendation phase of CARS [4].

The main ways to incorporate contextual information into recommender systems can be grouped into three approaches (Figure 3.1):

- **Contextual Pre-Filtering:** contextual information is used to filter or select the data before generating recommendations. Conceptually, context can be modelled as an operator that selects a context-specific subset of the user–item space:

$$c : (u, i) \in U \times I \longmapsto (\bar{u}, \bar{i}) \in U \times I \tag{3.4}$$

  so that the utility function r is computed on the context-filtered representation:

$$r : (\bar{u}, \bar{i}) \in U \times I \longmapsto r_{\bar{u}\bar{i}} \in \mathbb{R}. \tag{3.5}$$

  In practice, only data relevant to the current context are used to train or query the recommender.
- **Contextual Post-Filtering:** contextual information is ignored during the initial recommendation step (classic RS), and then used to re-rank or

filter the produced recommendations. Context can be seen as a function that modifies the predicted utility:

$$c : r_{ui} \in \mathbb{R} \longmapsto \bar{r}_{ui} \in \mathbb{R}, \tag{3.6}$$

where the original scores are adjusted according to contextual information.

- **Contextual Modelling:** contextual information is integrated directly into the recommendation model, i.e., into the utility function used to compute recommendations.

The integration of context within two common recommendation strategies can be summarised as follows:

- **Content-Based CARS**
  - **Pre-Filtering:** context is incorporated when building user and/or item profiles, often by maintaining context-specific profiles (typically user-centred);
  - **Post-Filtering:** context is used to filter or re-rank the recommendations generated by the content-based method; this introduces additional computations proportional to the number of recommended items and contextual conditions;
  - **Modelling:** contextual information is integrated into the similarity computation between user and item profiles (often via heuristic formulations).
- **Collaborative Filtering CARS**
  - **Pre-Filtering:** the rating data can be split by context, producing context-specific matrices (or expanded representations) used for prediction; this may reduce modelling complexity but requires richer data acquisition;
  - **Post-Filtering:** context is used to filter or re-rank predicted ratings, similarly to the content-based case;
  - **Modelling:** contextual information is embedded in the predictive model. Common options include:
    - **Heuristic-based:** context affects neighbourhood construction or similarity computations;
    - **Model-based:** context is incorporated as additional dimensions, e.g., through tensor factorisation.

    In addition, techniques such as Context-Aware Matrix Factorisation (CAMF) belong to the modelling family, as they integrate contextual information directly into matrix-factorisation-based prediction.

3.2 Related works

The necessity to personalize the users' cultural experience requires the employment of Recommender Systems (RSs), which have the ability to analyze and filter data in order to provide suggestions. The available data affects the RS strategy for elaborating predictions and allows their classification, as shown in the previous Section [120, 139, 158]. In recent years, recommender systems have benefited from the integration of complementary approaches, including graph models, ontologies, and machine learning techniques [53, 67, 134, 145]. Among these, context-awareness has proven particularly effective by incorporating environmental and situational data into the recommendation process, ultimately leading to the development of Context-Aware Recommender Systems (CARSs) [106, 119]. Enabled by the Internet of Things, CARSs adapt suggestions based on dynamic user contexts [119]. Unlike traditional systems that rely on user-item interactions, CARSs operate on a multi-dimensional model that includes contextual variables [2]. Contextual information can be integrated into recommender systems through three main strategies, as shown in Figure 3.1 [2, 106, 119]. Applying context to improve recommendations adds value by generating more elaborate suggestions, but it requires selecting the most relevant contextual data. Huang et al [68] propose a model that enhances personalised recommendation systems through dynamic user interest modelling and context-aware learning, where the integrated contextual information is adaptively adjusted to the specific application. Another crucial tool for CARS is the Knowledge Graph (KG). Goswami et al. [64] exploit it in a context-aware collaborative recommendation model to improve accuracy and efficiency. By incorporating contextual parameters of users and items, the model reduces excessive propagation within KGs, leading to more precise recommendations. Song et al. [156] propose a recommendation model that explicitly captures feature interactions across scenario–task pairs. The model introduces a scenario–task aware attention mechanism that adaptively weighs feature interactions according to these contextual factors, while sparsity-aware functions filter out irrelevant ones.

Several CARSs in the literature address multidimensionality by employing tensorial factorization [190], which is hybridized with Machine Learning and Deep Learning approaches [167, 178]. To improve the computational cost associated with the employment of tensorial factorizations, several CARSs utilize matrix factorizations by approximating the available contextual information. The first example consists of splitting methods that analyze interactions between user and context (User Splitting), item and context (Item Splitting), or both (User-Item Splitting). These methods work by increasing the dimension of the rating matrix in a two-dimensional recommendation method [18, 191]. Instead, the Context-Aware Matrix Factorization (CAMF) methods integrate biases that de-

pend on the interaction between context and users or items [17, 191] . Then, these methods select relevant contextual information and analyse it individually. The interest in CARSs encompasses several application fields, specifically tourism [119], where the possibility of suggesting appropriate Points of Interest (POIs) [145] based on context became crucial [40, 53, 85]. Specifically, the employment of recommender systems still represents a focus point concerning cultural heritage enhancement. Data scarcity is a significant problem in this field [34], and employing machine learning techniques based on matrix factorization represents an additional option to the usual approaches for overcoming this problem [44]. In [86], the authors introduce a mobile recommendation system designed for cultural heritage visitors, which leverages a multi-label classification approach. Rather than relying on demographic or location-based data, the proposed RS focuses on visiting preferences expressed through an intuitive icon-based interface. Users select visual cues representing their interests. Then, the system generates personalized POI recommendations proportionally to the assigned profile scores. In [175], the author proposes a group recommender system for promoting intangible cultural heritage, using k-means clustering and Non-negative Matrix Factorization. User preferences and item features are modelled in latent space, and group recommendations are generated by weighting individual user profiles based on inferred domain expertise. In [39], the authors present an architecture for enhancing cultural heritage by integrating Digital Storytelling techniques with a Context-Aware Recommender System (CARS). The goal is to offer personalized cultural experiences by adapting multimedia content to user preferences and contextual factors. The architecture exploits a collaborative filtering recommendation module and a non-linear Digital Storytelling strategy. In [74], the authors present a recommendation system based on matrix factoring that considers cultural factors such as the country of origin and the associated culture of the users. The work aims to show how the valorization of cultural aptitude is strongly influenced by cross-cultural contextualization. In [181], the author proposes a collaborative filtering-based recommender system to support the transmission of traditional ceramic culture within a digital learning platform. The system collects user behaviour data, including browsing activity, discussion participation, and interaction frequency, and recommends personalized learning content, as well as suggests peers with similar interests. In [121], the authors present a multi-layer personalization framework designed to support tangible and embodied interaction with cultural heritage. The system combines customization, context-awareness, and adaptivity, enabling visitors to interact with digital content through smart objects, location-based sensors, and responsive interfaces. Personalization is integrated into both content selection and interaction design, empowering visitors to shape their own experiences while curators retain complete control over content authoring. In [165], the authors explore the use

of Large Language Models (LLMs) as natural language-based recommender systems in museum settings. The methodology focuses on embedding LLMs within conversational agents to deliver personalized content suggestions in response to open-ended user queries. The system integrates three main components: a content database containing curated museum-related information, a dialogue interface powered by a fine-tuned LLM, and an interaction tracking mechanism that maps user inputs to their preferences and topics of interest. The model leverages semantic understanding and generative capabilities to provide real-time, context-sensitive recommendations. Massimo et al. [105] propose a three-step approach to next-POI recommendation. First, tourist visit trajectories are clustered according to contextual and content-related features of POIs. Second, Inverse Reinforcement Learning is applied to infer latent reward functions that capture the experiential value underlying tourist sequential choices. Finally, recommendation strategies are designed to balance predictive accuracy with novelty and satisfaction, aiming to suggest POIs that users may not know yet but are likely to perceive as rewarding. Finally, in [163], the authors propose a culture-based tourism recommendation system supported by a mobile application developed through community collaboration. The system leverages local knowledge to curate relevant cultural attractions, services, and products, integrating them into a user-centered digital platform. The application includes a location-aware route recommendation engine, a database of local cultural points of interest, and a rating system. Recommendations are generated by considering the user's current location and previously expressed interests.

This literature analysis testifies that the employment of matrix-factorization-based recommender systems is still widespread in the cultural heritage field. However, the analyzed works present an experimental phase that relies on datasets from the literature or on in-situ evaluations involving a limited number of users. In contrast, this work not only aims to define a framework to enhance user cultural experiences but also tests its effectiveness on a sample of nearly 2000 users.

In addition, recent developments extend traditional recommender systems by integrating chatbots and immersive technologies to enhance cultural heritage experiences. Chatbot-based recommenders enable more natural and conversational interactions, allowing users to express preferences through dialogue [7]. Sperli [157] proposes a deep learning-based framework with a GRU-powered chatbot that personalizes suggestions of tangible and intangible heritage through a conversational interface. Instead, [48] propose a loosely coupled integration of a cultural heritage web guide with external Large Language Models, relying on prompt engineering to enhance user interaction. The approach extends the system's local knowledge by leveraging the question–answering capabilities of the chatbot, combining free-text queries with context-dependent question

generation to assist lay users in accessing relevant information, and tailoring answers to the content already explored. [171] improve Conversational Recommender Systems by exploiting contextual and temporal signals within dialogues, reducing dependence on domain-specific knowledge. Similarly, [180] introduce a framework that integrates LSTM networks with Large Language Models to dynamically capture user intent, addressing cold-start issues and enabling more accurate, context-aware recommendations. Furthermore, Augmented and Virtual Reality (AR/VR) environments enriched with Virtual Humans enable narrative-driven and emotionally adaptive interactions, enhancing visitor engagement and learning [99]. Solutions like 3VR map user experience to dynamically adapt virtual narratives and environments [169], while end-user development approaches support the customization of smart interactive experiences through user-friendly authoring tools [13]. Integrating chatbots and AR/VR with recommender systems enables increased involvement from cultural users and an enhanced cultural user experience.

3.3 CONTEXTUAL BIASES MATRIX FACTORIZATION

This Section aims to introduce a novel Contextual Modelling recommendation method that exploits matrix factorization and contextual biases. The proposed approach evaluates context as tensor representation does, then all contexts are considered as ordered tuples and not individually. We define this context representation as *embedded context* that aims to consider all the contextual information simultaneously. For instance, a user can watch a movie 'alone' and 'at home' as shown in Figure 3.2. In this case, the individual evaluation of the companion context value alone can take into account all the values of location context and not only the value 'at home'; likewise, the singular evaluation of the location context 'at home' evaluates all possible values of the companion context and not only the value 'alone'. Instead, the embedded context allows elaborating the location and companion contexts only for the proper values.
Therefore, the Section first focuses on the idea of embedded context compared with two literature methods. Consequently, the proposed context-aware recommender system is described.

3.3.1 AN ANALYSIS OF CONTEXT ELABORATION IN LITERATURE

To introduce the concept of embedded context, we first provide an analysis of context-aware recommender systems based on matrix factorisation and machine learning approaches in the literature.
As discussed in Section 3.2, Baltrunas et al. [18, 19] propose a context-aware collaborative filtering recommender system based on a Contextual Pre-Filtering strategy. This method, referred to as *Item Splitting*, partitions the available

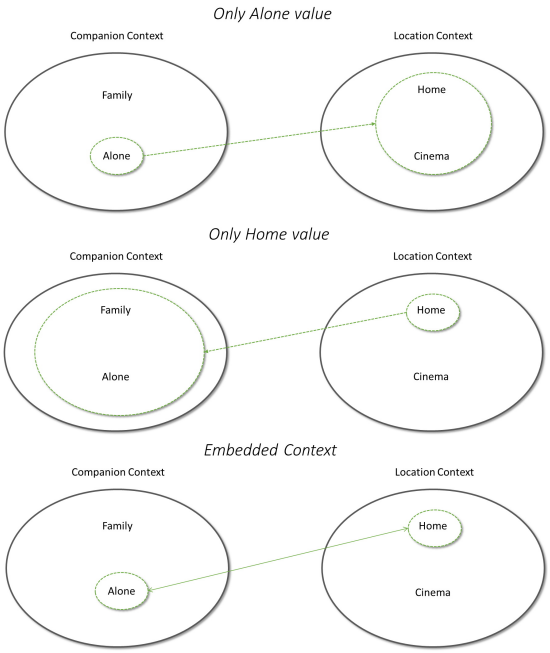


Figure 3.2. Example of embedded context with two contextual domains.

ratings for an item based on the contextual values under which users provided their ratings. In this way, only ratings relevant to the current context are selected, and the recommendation technique operates on the context-appropriate subset of the data.
Applying the same idea to the rows of the rating matrix (instead of the columns) leads to *User Splitting* [191]. Finally, splitting both rows and columns yields *User–Item Splitting* [191]. To limit computational cost, User Splitting and Item Splitting typically consider a single contextual dimension, whereas User–Item Splitting can incorporate two contextual dimensions, one applied to rows and the other to columns.
To clarify these approaches, Table 3.2 introduces two contextual domains and their respective values.

Table 3.2. Contextual domains and relative values considered to introduce the idea of embedded context and compare it with literature methods.

Location Context	Companion Context
Home (H)	Alone (A)
Cinema (C)	Family (F)

For instance, the User–Item Splitting method may consider the *companion*

context for users and the *location* context for items. Each user u is then represented as different context-specific users (e.g., u in `Alone` and u in `Family`), and analogously each item becomes context-specific (e.g., i in `Home` and i in `Cinema`). Table 3.3 shows an example of the rating matrix produced by the User–Item Splitting method.

Table 3.3. Example of User-Item Splitting method on the example of Table 3.2.

	i_1-Home	i_1-Cinema	...
u_1-Alone	$\hat{r}_{u_1,i_1,A,H}$	$\hat{r}_{u_1,i_1,A,C}$	...
u_1-Family	$\hat{r}_{u_1,i_1,F,H}$	$\hat{r}_{u_1,i_1,F,C}$	...
⋮	⋮	⋮	⋱

Baltrunas et al. [17] propose a Contextual Modelling approach known as Context-Aware Matrix Factorisation (CAMF). This approach computes the rating forecast using k contextual dimensions according to:

$$\hat{r}_{uic_1\ldots c_k} = \bar{r} + p_u \cdot q_i + b_u + \sum_{z=1}^{k} B_{izc_z}, \tag{3.7}$$

where $\bar{r}$ denotes the global mean rating, $p_u \cdot q_i$ expresses user–item affinity in the latent space, b_u is the decontextualised user bias, and B_{izc_z} denotes the bias associated with contextual dimension z.
Different CAMF variants can be defined depending on how contextual dimensions are selected:

- CAMF-C: a single contextual dimension is considered ($k = 1$);
- CAMF-CI (Context–Item): contextual information is associated with items, as in Eq. (3.7). This increases the number of parameters to be estimated;
- CAMF-CC: a selected subset of contextual dimensions is used to reduce computational cost in rating forecast computation;
- CAMF-CU (Context–User) [191]: contextual information is associated with users rather than items;
- CAMF-CUCI [191]: contextual information is associated with both users and items.

For instance, the CAMF-CI approach applied to the example in Table 3.2 computes rating forecasts as:

$$\hat{r}_{uic_1c_2} = \bar{r} + p_u q_i + b_u + Bi, 1, c_1 + B_{i,2,c_2} \qquad c_1 \in \text{H,C}\ c_2 \in \text{A,F} \tag{3.8}$$

Overall, the approaches discussed above treat contextual dimensions *independently*: each domain contributes to the rating prediction separately.

3.3.2 The embedded context

Classical recommender systems that do not consider context typically exploit the rating matrix

$$R = (r_{ui})_{u \in U,, i \in I} \in \mathbb{R}^{m \times n} \qquad m = |U|, \;\; n = |I| \tag{3.9}$$

to represent known ratings, where the two dimensions correspond to the cardinalities of the user set U and the item set I.
In a context-aware setting, a two-dimensional representation is insufficient to encode the additional contextual dimensions. A common formulation is therefore a tensor:

$$\mathcal{R} = (r_{uic_1 \ldots c_k})_{u \in U, i \in I, c_1 \in C_1, \ldots, c_k \in C_k} \in \mathbb{R}^{m \times n \times w_1 \times \cdots \times w_k}, \tag{3.10}$$

$$m = |U|, \quad n = |I|, \quad w_j = |C_j| \quad \text{for } j = 1, \ldots, k.$$

Here, $r_{uic_1 \ldots c_k}$ denotes the rating provided by user u for item i under contextual conditions $c_1 \in C_1, \ldots, c_k \in C_k$. In real datasets, ratings are only observed for a subset of the tensor entries; unobserved entries represent missing information, often stored as zeros.
This tensorial view motivates the embedded context concept exploited by the proposed approach, Contextual Biases Matrix Factorization (CBMF). Since tensor decompositions can be computationally expensive and memory-demanding, CBMF retains matrix factorisation while modelling context through a single *embedded* contextual state that represents the joint configuration of all contextual dimensions.
To clarify the embedded context concept, Table 3.4 introduces the embedded context derived from the contextual information in Table 3.2 and highlights how CBMF differs from other approaches, such as User–Item Splitting (Table 3.3) and CAMF-CI (Eq. (3.8)).

Table 3.4. Example of Embedded Context on the example of Table 3.2.

Embedded Context
(Home, Alone)
(Home, Family)
(Cinema, Alone)
(Cinema, Family)

In the example, the embedded context can be interpreted as a single contextual variable whose values correspond to all tuples formed by combining the values of each contextual dimension. Accordingly, a tensor with two contextual

dimensions (each with two values) can be re-indexed into a three-dimensional tensor with four embedded context values, rather than keeping the original four-dimensional representation.
To further illustrate the concept, consider the case in Figure 3.2. The tensor has four dimensions:

$$\bar{\mathcal{R}} = \left(\bar{\mathcal{R}}_{uic_1c_2}\right) \in \mathbb{R}^{m\times n\times 2\times 2} \qquad m = |U|,\ \ n = |I|. \tag{3.11}$$

The embedded context allows defining an equivalent three-dimensional tensor:

$$\tilde{\mathcal{R}} = \left(\tilde{\mathcal{R}}_{uic}\right) \in \mathbb{R}^{m\times n\times 4}, \tag{3.12}$$

where the single contextual dimension enumerates all combinations of the two original contextual variables (see Table 3.4).
If recommendations are computed using only one contextual variable (e.g., `Alone` in the companion context), the unconsidered contextual dimension is effectively averaged out. For example, the rating forecast for `Alone` may be written as:

$$\hat{\mathcal{R}}_{u,i,\texttt{alone}} = \frac{1}{2}\left(\bar{\mathcal{R}}_{u,i,\texttt{alone},\texttt{home}} + \bar{\mathcal{R}}_{u,i,\texttt{alone},\texttt{cinema}}\right). \tag{3.13}$$

The same reasoning applies when conditioning only on a location context value. Embedded context, instead, preserves the dependency between contextual dimensions by modelling their joint configuration, thus avoiding the loss of information due to marginalisation over unobserved contextual factors. When more contextual values exist in the unconsidered dimension, the averaging would involve more terms.
Table 3.5 summarises the typologies of rating forecast computation based on the contextual information considered.

Table 3.5. Summary of the typologies of rating predictions based on the example of Table 3.2.

Context	Contextual Information	Rating Forecast
Companion	*Alone*	$\frac{1}{2}\left(\bar{\mathcal{R}}_{u,i,alone,home} + \bar{\mathcal{R}}_{u,i,alone,cinema}\right)$
Location	*Home*	$\frac{1}{2}\left(\bar{\mathcal{R}}_{u,i,alone,home} + \bar{\mathcal{R}}_{u,i,family,home}\right)$
Embedded Context	*Alone,Home*	$\tilde{\mathcal{R}}_{u,i,alone,home}$

This introductory example underscores the objective of the proposed approach: to jointly evaluate contextual information to avoid losing information about context combinations.

3.3.3 The proposed context aware recommender system

The proposed Context-Aware Recommender System, defined as Contextual Biases Matrix Factorization (CBMF), falls into the Collaborative Filtering Model-Based category and integrates context according to the Contextual Modelling strategy.
In particular, the developed CARS exploits the Singular Value Decomposition (SVD) and achieves context awareness through contextualized biases learned to adapt suggestions based on contextual data. The peculiarity of the recommendation approach consists of matrix factorization employment without losing the integration among all contextual information. Usually, CARSs take advantage of matrix factorizations evaluating part of contextual information selected based on relevancy or exploit tensor factorizations to assess all contextual information increasing the computation cost. Instead, this CARS aims to exploit SVD to evaluate the affinity between user and item by conserving the tensorial representation of context to adapt rating forecasts.
In the application of SVD in classical RSs, the matrix $R \in \mathbb{R}^{m \times n}$ considers interactions among m users and n items and takes advantage of several possible decomposition models, such as Principal Component Analysis (PCA), Non-Negative Matrix Factorization (NMF), Probabilistic Matrix Factorization (PMF), and, as mentioned before, Singular Value Decomposition (SVD) [37]. In particular, the SVD factorizes the rating matrix (3.9) into the product of three matrices

$$R = UDV^t = \sum_{i=1}^{r} \sigma_i u_i v_i^t \tag{3.14}$$

where $U = (u_1, \ldots, u_m) \in \mathbb{R}^{m \times m}$ and $V = (v_1, \ldots, v_n) \in \mathbb{R}^{n \times n}$ are the matrices of left and right singular vectors, respectively, v_i^t is the transpose of the column vector v_i, $i = 1, \ldots, n$, $D = \text{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n}$ is a diagonal matrix with $p = \min\{m, n\}$, and $r =$rank(R) is the rank of the matrix R. In particular, singular values $\sigma_1, \ldots, \sigma_p$ satisfy the following property

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0. \tag{3.15}$$

For application purposes, the use of the factorization requires an approximation process in which the matrix $U_k = (u_1, \ldots, u_k) \in \mathbb{R}^{m \times k}$, obtained from firsts $k < r$ left singular vectors, the matrix $V_k = (v_1, \ldots, v_k) \in \mathbb{R}^{n \times k}$, built from firsts k right singular vectors, and the matrix $D_k = \text{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbb{R}^{k \times k}$, obtained from firsts k rows and columns of the matrix D, allow the construction of the matrix

$$R_k = U_k D_k V_k^t \in \mathbb{R}^{m \times n}. \tag{3.16}$$

According to the Eckart-Young Theorem, the approximation error related to

the approximation of the rating matrix R with the matrix R_k is

$$\min_{rank(B)=k} ||R - B||_2 = ||R - R_k||_2 = \sigma_{k+1} \tag{3.17}$$

and matrix (3.16) represents the best approximation among all matrices of rank k. Moreover, defining matrices

$$P_k = U_k\sqrt{D_k} \in \mathbb{R}^{m\times k}, \qquad Q_k = V_k\sqrt{D_k} \in \mathbb{R}^{n\times k}, \tag{3.18}$$

the matrix $R_k = P_k Q_k^t$ can be seen as the product of fake numerical users' and items' profiles learned through machine learning techniques, whereas the value k represents the number of latent factors.
Therefore, selecting the rows $p_u \in \mathbb{R}^k$ in the matrix P_k and $q_i \in \mathbb{R}^k$ in the matrix Q_k related to the user u and the item i, respectively, the scalar product $p_u q_i^t$ expresses the user-item affinity.
The introduction of biases makes the proposed recommendation engine context-aware, whereas biases b_{uc} and $\tilde{b}_{ic}$ are related to the embedded context c and the user u and the item i, respectively. Considering dataset $D = \{(u, i, c) : \exists r_{uic}\}$, where r_{uic} represents the rating associated to the user u, the POI i and the embedded context c, the applied machine learning technique considers the loss function

$$\mathcal{L} = \sum_{(u,i,c)\in D} \frac{1}{2}\left[\left(r_{uic} - \left(\bar{r}_c + b_{uc} + \tilde{b}_{ic} + p_u \cdot q_i^t\right)\right)^2 + \right. \tag{3.19}$$

$$\left. +\lambda\left(b_{uc}^2 + \tilde{b}_{ic}^2 + \|p_u\|^2 + \|q_i\|^2\right)\right],$$

where $\bar{r}_c$ consists of the mean of available ratings associated with the embedded context c, b_{uc} represent the bias connecting the user u and the embedded context c, $\tilde{b}_{ic}$ is the bias related to the POI i and the embedded context c, and p_u and q_i represents the column vectors of the matrices P_k and Q_k (see (3.18)) learned through the available ratings. The recommender engine also requires a methodology to overcome the cold start problem related to the collaborative filtering approach application. The system can understand user preferences and randomly select suitable items' interactions through chatbot support.
Finally, the application of the CBMF method implies the calculation of the needed parameters. For instance, in the example related to the context information of Table 3.2, the method needs to calculate the parameter shown by Table 3.6.

3.4 PATH ELABORATION MODULE

In this section, we describe the optimisation model used to transform recommended POIs into a feasible visit path. The objective is to select a sequence

Table 3.6. Parameters required to obtain the ratings predictions based on the CBMF method related to the embedded contexts.

Embedded Context	*Mean*	*Bias User*	*Bias Item*
(Home,Alone)	$\bar{r}_{(H,A)}$	$b_{u,(H,A)}$	$b_{i,(H,A)}$
(Home,Family)	$\bar{r}_{(H,F)}$	$b_{u,(H,F)}$	$b_{i,(H,F)}$
(Cinema,Alone)	$\bar{r}_{(C,A)}$	$b_{u,(C,A)}$	$b_{i,(C,A)}$
(Cinema,Family)	$\bar{r}_{(C,F)}$	$b_{u,(C,F)}$	$b_{i,(C,F)}$

of traversed connections that maximises the overall utility of the visited POIs while satisfying time and feasibility constraints. We define the following input parameters:

- V: set of points of interest (POIs);
- $n = |V|$: number of POIs;
- t_{ij}: travel time from POI i to POI j, $i, j = 1, ..., n$;
- T_{max}: maximum time of visit;
- $x_{ij} = \begin{cases} 1, & \text{if the road from } i \text{ to } j \text{ is in the path,} \\ 0, & \text{otherwise,} \end{cases}$ $i, j = 1, ..., n$;
- NI: set of isolated POIs (can be reached from only one POI);
- $g_{ij} = \begin{cases} 1, & \text{if POI } i \text{ is connected to POI } j, \\ 0, & \text{otherwise,} \end{cases}$ $i, j = 1, ..., n$;
- S: starting point;
- A: arrival point.

The objective function aim to maximize the number of roads travelled and, as a consequence, the number of POIs visited.

$$\max \sum_{i,j=1;i\neq j}^{n} g_{ij} x_{ij} \hat{r}_{ujc} \tag{3.20}$$

The preferences of the user u in the embedded context c are expresses by the rating forecast r_{ujc} that weights the POI j in (3.20). The object function is subject to several constraints which:

- assure that the maximum available time is not exceeded (3.21),
- indicate that all the POIs, except starting and arrival ones, are transit points (3.22),
- prevent returning to a not isolated POI ((3.23) and (3.24)),
- force the exit from the starting point (3.25) and the entrance to the arrival point (3.26),
- avoid cycles for not isolated POIs (3.27),
- compute the visit time of the POI (3.28),
- impose that a road can be walk ($x_{ij} = 1$) or not ($x_{ij} = 0$) (3.29).

$$\sum_{i,j=1}^{n} t_{ij} x_{ij} \leq T_{max} \tag{3.21}$$

$$\sum_{j=1;i\neq j}^{n} g_{ij} x_{ij} - \sum_{j=1;i\neq j}^{n} g_{ji} x_{ji} = \begin{cases} 1, & \text{if } i = S \text{ and } S \neq A, \\ -1, & \text{if } i = A \text{ and } S \neq A, \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, ..., n, \tag{3.22}$$

$$\sum_{j=1;j\notin NI}^{n} g_{ij} x_{ij} \leq 1, \qquad i = 1, ..., n, \tag{3.23}$$

$$\sum_{i=1;i\notin NI}^{n} g_{ij} x_{ij} \leq 1, \qquad j = 1, ..., n, \tag{3.24}$$

$$\sum_{j=1;j\notin NI}^{n} g_{ij} x_{ij} = 1, \qquad i = S, \tag{3.25}$$

$$\sum_{i=1;i\notin NI}^{n} g_{ij} x_{ij} = 1, \qquad j = A, \tag{3.26}$$

$$g_{ij} x_{ij} + g_{ji} x_{ji} \leq 1, \qquad i, j = 1, ...n, \quad i, j \notin NI \tag{3.27}$$

$$x_{ii} \geq \frac{x_{ij} + x_{ki}}{2}, \qquad i, j, k = 1, ..., n, \tag{3.28}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j = 1, ...n. \tag{3.29}$$

From an architectural perspective (Chapter 2), the Path Elaboration Module belongs to the Cultural Experience Engine within the Inference Engine Layer. Given the POI relevance scores produced by the Context-Aware Recommendation Module (in the current embedded context), it computes a feasible itinerary that satisfies spatial connectivity and time constraints. In this way, the module acts as the optimisation bridge between predictive preference modelling and real-world visit execution, ensuring that personalisation remains compatible with on-site constraints.

3.5 EXPERIMENTAL PHASE

The validation of the proposed framework related to the Cultural Experience Engine is carried out through a two-step evaluation procedure: the first consists of the recommendation method validation by measuring its accuracy and reliability through collected data, and the second takes advantage of an in situ evaluation based on 1978 users. In particular, the experimental phase globally evaluates several aspects of the developed prototype and, in particular, assesses the recommendation engine. However, the experimental phase focuses on the ability of the recommendation engine to improve the user's cultural experience by personalizing the visit, and the technologies implemented in the prototype represent co-actors of the recommender engine aimed to understand user preferences, improve the elaboration of data, and provide services.

The accuracy and reliability evaluation phase achieves evaluation of the recommender engine through accuracy measures such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) [120] and qualitative measures, such as Normalized Discounted Cumulative Gain (NDCG) [136]. The comparison methods represent recommendation approaches comparable with the proposed one based on Singular Value Decomposition. In particular, the proposed contextual recommender system and comparison methods require fewer parameters with respect to deep learning based approaches, guaranteeing a fast update of the parameters. In addition, the evaluation of the recommendation approach not only exploits accuracy measures, but also focuses on the qualitative analysis through NDCG. Moreover, the choice of a matrix factorization-based recommender system is coherent with the application field, as shown in the literature analysis ([74, 86, 175]). The in situ evaluation, instead, takes advantage of a questionnaire proposed to users, and we collected reviews for six months through visitors to Pompei and Paestum-Velia archaeological sites.

3.5.1 Dataset description

Evaluating the accuracy and reliability of the previously introduced recommendation approach requires a dataset on which to evaluate the metrics of interest. The dataset consists of 972 ratings collected from 162 users who rated some of the 81 POIs within the archaeological park of Pompeii. The dataset acquisition phase lasted three months. In addition, the dataset structure has seven columns representing timestamp, user ID, POI ID, user rating, crowding context, time context, and weather context. The values of the contexts are:

- Crowdedness context: free, almost free, medium, crowded.
- Time context: low, medium, high, very high.
- Weather context: sunny, cloudy, rainy.

The design of the crowding level depends on the specific POI from its location in the archaeological site. Appropriate sensors are exploited to capture the crowding level. In indoor locations with forced entry and exit, sensors capable of evaluating incoming and outgoing people are sufficient. In open places and indoor POIs with multiple entries and exits, sensors that exploit the camera to count the number of people have been installed. The time context depends on the user's available time and has to be set based on the size of the archaeological park. Finally, the weather context summarizes the evaluation of each cultural object based on the weather, implicitly considering whether the item is indoor or outdoor. Moreover, the evaluated contexts are elaborated to identify embedded contexts composed of tuples based on all possible combinations of each value of singular contexts.

3.5.2 *ACCURACY AND RELIABILITY EVALUATION PHASE*

The first evaluation step required the employment of the dataset D described in the Subsection 3.5.1. Considered quantitative evaluation metrics on dataset D consist of the RMSE

$$\text{RMSE} = \sqrt{\frac{1}{|D|} \sum_{r_{uic} \in D} \left(r_{uic} - \hat{r}_{uic}\right)^2}, \tag{3.30}$$

and the MAE

$$\text{MAE} = \frac{1}{|D|} \sum_{r_{uic} \in D} \left|r_{uic} - \hat{r}_{uic}\right|, \tag{3.31}$$

with $\hat{r}_{uic}$ prediction elaborated by the recommender engine about the user u and the item i in the embedded context c, which is the tuple of all evaluated contextual domains.
Before assessing the accuracy related to the proposed recommendation approach, a preliminary investigation permits the identification of the appropriate learning rate α, the proper parameter λ, and the correct number of iterations iter. In particular, Figure 3.3 shows obtained RMSE results related to the variation of the learning rate and the parameter in the interval $[0.01, 0.1]$ and the iteration among values 10, 20, 30, and 40.

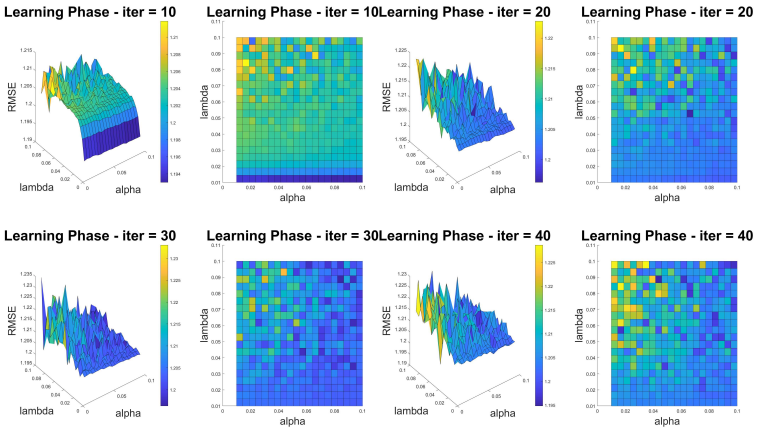


Figure 3.3. Results related to RMSE according to the variation of hyperparameters depending on the number of iteration $\text{iter} = 10, 20, 30, 40$, the learning rate $\alpha \in (0, 0.1]$, and the hyperparameter $\lambda \in (0, 0.1]$ related to the regularization of parameters in (3.19). The best hyperparameter identified are $\alpha = 0.1$, $\lambda = 0.01$, and $\text{iter} = 10$.

According to Figure 3.3, the best values are $\alpha = 0.1$, $\lambda = 0.01$, and iter $= 10$. In addition, Table 3.7 summarizes results related to RMSE and MAE

of the proposed recommendation method, named Contextual Bias Matrix Factorization (CBMF), and the comparison with literature methods that exploit matrix factorization for calculating rating forecasts. In particular, the comparison methods chosen are Item Splitting, User Splitting, User-Item Splitting, CAMF-C, CAMF-CU, and CAMF-CI. The implementation of the comparison methods was carried out using the CarsKit tool [192]. MAE and RMSE results are calculated by dividing the dataset into 5 folds and using the k-fold Cross Validation. Reported values represent the mean of 20 numerical experiments to guarantee the experimental reliability.

Table 3.7. Summary of RMSE and MAE results related to the proposed approach (CBMF) and the comparison methods based on matrix factorization. Results are reported as mean ± standard deviation.

Method	**MAE**	**RMSE**
CAMF-C	1.1068 ± 0.0120	1.5124 ± 0.0324
CAMF-CI	1.0145 ± 0.0130	1.3295 ± 0.0342
CAMF-CU	0.9852 ± 0.0076	1.2204 ± 0.0169
Item Splitting	1.0746 ± 0.0110	1.3967 ± 0.0210
User Splitting	0.9901 ± 0.0085	1.2249 ± 0.0185
User-Item Splitting	0.9837 ± 0.0100	1.2189 ± 0.0200
CBMF	*0.9768 ± 0.0044*	*1.2023 ± 0.0153*

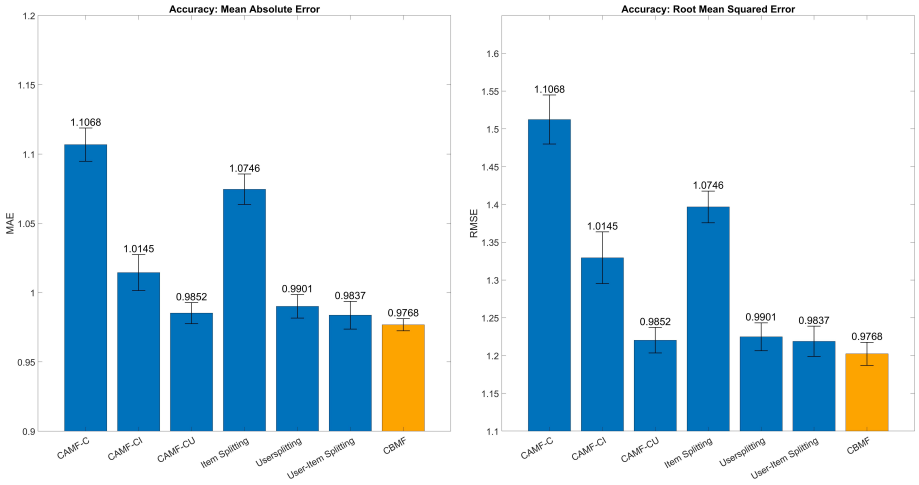


Figure 3.4. Graphical representation of accuracy achieved by the proposed recommendation approach and the comparison methods.

The obtained results confirm the effectiveness of the proposed recommendation approach, which leverages context tuple evaluation to incorporate global con-

textual information. Table 3.7 establishes that CBMF returns lower errors than those of comparison methods, and therefore, the proposed recommendation approach overcomes other methods based on matrix factorization.
Another employed metric consists of Normalized Discounted Cumulative Gain (NDCG), a ranking metric that evaluates how well recommended items are ordered by relevance. In particular, this metric is obtained confronting the Discounted Cumulative Gain (DCG) for the best N recommendations

$$\mathrm{DCG}_N = \sum_{p=1}^{N} \frac{r\,(p)}{\log_2\,(p+1)}, \tag{3.32}$$

where $r\,(p)$ represents the relevance degree of the results in position p, with the Ideal Discounted Cumulative Gain IDCG_N. Specifically, the NDCG is calculated as follows

$$\mathrm{NDCG}_N = \frac{\mathrm{DCG}_N}{\mathrm{IDCG}_N} \tag{3.33}$$

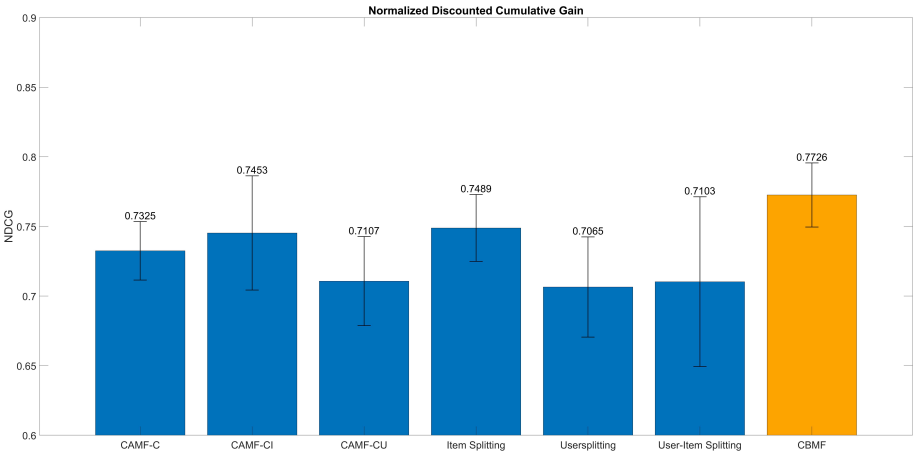


Figure 3.5. Results related to the NDCG metric with $N = 10$.

Figure 3.5 confirms that the proposed approach provides reliable recommendations and obtained results that overcome the ones of the comparison methods. These results allow the validation of CBMF on collected data and represent the basis for the in situ evaluation development.

3.5.3 IN SITU EVALUATION

After the accuracy and reliability evaluation, the in situ evaluation allows for testing the goodness of the ability of the proposed framework to enhance the cultural experiences, focusing mainly on the validation of the recommendation engine. The design of the in situ evaluation required the involvement of

1978 users that visited the Pompei and Paestum-Velia archaeological parks. Moreover, to evaluate the proposed architecture, the evaluation required the development of a prototype to assess all four functional layers.

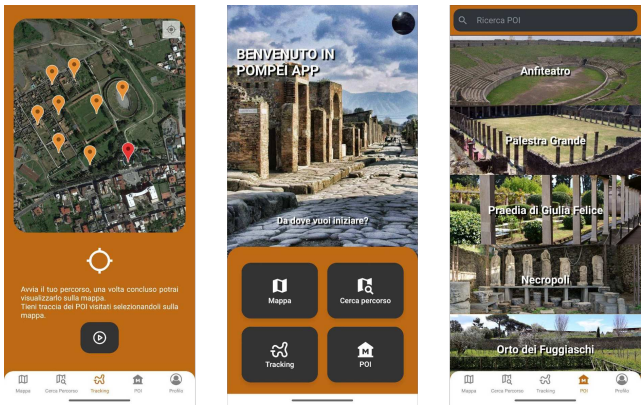


Figure 3.6. Images from the developed prototype.

The prototype necessitates selecting appropriate IoT devices. In particular, crowdedness monitoring takes advantage of counting cameras in outdoor POIs instead of counting sensors at the entrance and exit of indoor places. Specifically, archaeological parks are divided into zonas managed through a Raspberry Pi4 Model B connected to a camera able to count outdoor installation sites. Each Raspberry supervises several Arduino Pro Mini linked to sensors related to indoor spaces. These sensors also allow for the evaluation of the usual people flow in each POI. Therefore, the Raspberry Pi4 preliminary pre-elaborates data before sending it to the Cloud environment ThingsBoard, which allows real-time monitoring through a Dashboard related to each zona. The local oversees permits for collecting local data organized globally into the final database. The IoT components of the prototype also exploit the installation of a weather station based on a Raspberry Pi4 linked to sensors that are able to monitor humidity, temperature, solar radiation, wind velocity and direction, and rainfall level. The development of the Inference Engine takes advantage of the Python programming language. In particular, the recommender engine exploits the package Numpy, and the chatbot requires Tensorflow, Keras, and the Natural Language Toolkit. The prototype provides functionalities related to recommendations, the chatbot exploited by the Interaction Service, non-linear Digital Storytelling, and Augmented Reality. Moreover, it makes available a map to enable users to find their way around archaeological sites and helpful information throughout the experience. In the described prototype, the user experiences a visit that begins with a brief dialogue with the chatbot, designed to gather preferences, available time, and objectives. Combining it with data collected from sen-

sors on weather, crowding, and location, the system generates a personalised itinerary that dynamically adapts to real conditions. During the visit, the user navigates the site with the help of an interactive map, receives context-aware suggestions about POIs, and can explore additional content through Digital Storytelling and Augmented Reality. At the end of the experience, the visitor can provide feedback that improves the system's performance.
After the prototype development, the evaluation takes advantage of a questionnaire based on six sections, where each section aims to estimate different aspects of the prototype. Each question deals with the following topics divided by sections:

1. Usability: users estimate the facility of the prototype usage through the judgement of the following questions:
   1a. Chatbot: evaluation of the Chatbot utility;
   1b. Response Time: measurement of the required times by the prototype for all elaborations that support users;
2. Recommendation: users evaluate the suggested POIs through the assessment of
   2a. Suitability: visited POIs result adapt to users' preferences;
   2b. Path Appropriateness: global evaluation of visited sites proposed by recommender engine;
   2c. Reliability: recommender engine provides appropriate forecasts;
3. Presentation: users estimate the experience enjoyability provided by exploited tools. In particular, they evaluate:
   3a. Digital Storytelling: the sites' history narration comes across as stimulating and engaging;
   3b. Augmented Reality: the employment of AR makes the cultural experience fascinating;
4. Dialogue: users assess the interaction with the chatbot by evaluating the following aspects:
   4a. Smoothness: the chatbot dialogue takes place smoothly and without unexpected jumps;
   4b. Adequacy: the chatbot engine can understand user intentions correctly;
   4c. Understanding: the chatbot does not claim not to have understood what the user is asking for;
5. Support: a qualitative assessment of the prototype's ability to help users through:
   5a. Map utility how much the map assistance results helpful for users;
   5b. Information Utility: the prototype provides functional information;
6. Enjoyment: global evaluation of the cultural experience through the estimation of the following aspects:
   6a. Easiness to access services and POIs;

6b. The experience falls within the available time.

Users can answer each section of the questionnaire with five levels of agreement: Totally Agree (TA), Agree (A), Neutral (N), Disagree (D), and Totally Disagree (TD).

The understanding of the users' satisfaction level takes advantage of dividing answers into three classes:

- the class Positive includes answers TA and A;
- the class Neutral involves the answers N;
- the class Negative considers answers TD and D.

Table 3.8 and Table 3.9 focus on overall results obtained in six months of evaluation.

Table 3.8. Results related to the responses obtained throughout the evaluation period.

Question	TA	A	N	D	TD
1a	974	769	191	34	10
1b	901	814	217	37	9
2a	1093	689	163	26	7
2b	1075	720	149	25	9
2c	1048	740	153	27	10
3a	1088	675	181	27	7
3b	1068	699	188	18	5
4a	1038	716	187	27	10
4b	1023	736	177	31	11
4c	1034	743	181	19	1
5a	752	950	228	37	11
5b	883	848	217	23	7
6a	1043	715	198	19	3
6b	870	863	204	29	12

All evaluation fields overcome the percentage of 86%, verifying the goodness of the developed prototype and the proposed architecture. Due to the evaluation of questions 1a (Chatbot utility), 4a (dialogue smoothness), 4b (dialogue adequacy), and 4c (understanding), Tables 3.8 and 3.9 show the interaction facility between users and the chatbot. Indeed, the satisfaction results related to these questions are 88.12%, 88.68%, 88.93%, and 89.84%, respectively. The presentation techniques based on Digital Storytelling and Augmented Reality also reached 89.13% and 89.33% satisfaction percentages, respectively. In particular, the recommendation section globally reaches almost 90%, ensuring the suggestions' quality. In particular, users evaluate the global suitability of visited POIs, the personalization level of the proposed path, and the reliability of the recommendations. The satisfaction evaluation obtained in all the

Table 3.9. Results related to the percentage of satisfaction throughout the evaluation period.

Question	Positive	Neutral	Negative
1a	88.11931	9.656218	2.224469
1b	86.70374	10.97068	2.325581
2a	90.09100	8.240647	1.668352
2b	90.74823	7.532861	1.718908
2c	90.39434	7.735086	1.870576
3a	89.13043	9.150657	1.718908
3b	89.33266	9.50455	1.162792
4a	88.67543	9.453994	1.870576
4b	88.92821	8.948433	2.123357
4c	89.83822	9.150657	1.011122
5a	86.04651	11.52679	2.426694
5b	87.51264	10.97068	1.516684
6a	88.87765	10.01011	1.112235
6b	87.61375	10.31345	2.072801

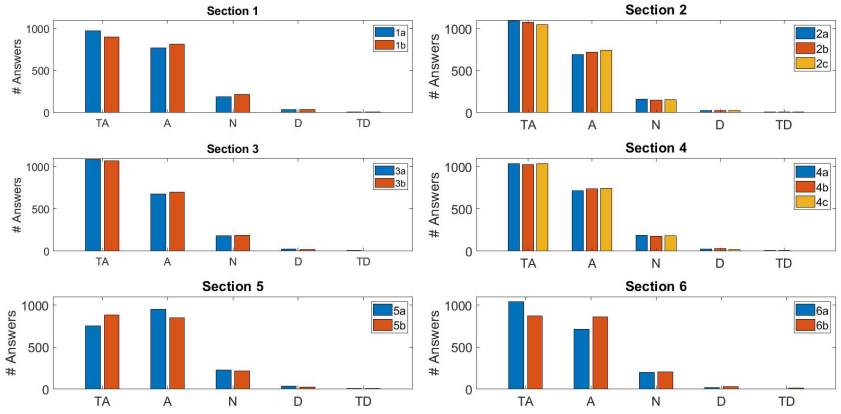


Figure 3.7. Graphical summary of the obtained answers, divided into the six sections of the evaluation questionnaire.

recommendation questions returns the goodness of the design done on the recommendation engine and the overall evaluation consistency of the embedded context through contextual biases by going on to confirm the results obtained in the accuracy and reliability evaluation phase of the contextual recommendation engine.
To further confirm these assessments, Table 3.10 summarizes the percentage of the users' satisfaction level month by month related to the recommendations.

In particular, Table 3.10 shows that the satisfaction level overcomes 88% in all months and reaches a value higher than 91% in the last two months.

Table 3.10. Satisfaction results related to the recommendation section for each evaluation month. Each Table shows Positive (P), Neutral (NT), and Negative (NG) percentages.

Month 1

	P	NT	NG
2a	88.26291	9.389671	2.347418
2b	88.73239	8.450704	2.816901
2c	89.20188	7.042254	3.755869

Month 2

	P	NT	NG
2a	89.48864	8.522727	1.988636
2b	90.05682	8.238636	1.704545
2c	90.34091	7.954545	1.704545

Month 3

	P	NT	NG
2a	90.21407	7.951071	1.834862
2b	91.13150	7.033639	1.834861
2c	89.90826	7.339450	1.741294

Month 4

	P	NT	NG
2a	89.30348	8.706468	1.990050
2b	90.54726	7.960199	1.492537
2c	90.04975	8.208955	1.741294

Month 5

	P	NT	NG
2a	91.18280	7.956989	0.860215
2b	91.39785	7.096774	1.505376
2c	91.18280	7.741935	1.075269

Month 6

	P	NT	NG
2a	91.78082	6.849315	1.369863
2b	92.23744	6.392694	1.369863
2c	91.32420	7.762557	0.913242

Therefore, the obtained results confirm the possibility of improving users' cultural experience by customizing paths through suitable and reliable suggestions.

3.5.4 *DISCUSSION ABOUT EXPERIMENTAL RESULTS*

Section 3.5 introduces the experimental phase, which consists of two steps: the first aims to quantitatively and qualitatively evaluate the developed recommendation approach, while the second tests the ability of the proposed framework to improve users' cultural experience.
The first part of the evaluation, focused on accuracy and reliability, confirms the robustness of the Contextual Bias Matrix Factorization (CBMF) approach based on the dataset, described in Section 3.5.1 and based on the ratings collected in specific contextual domains. As reported in Table 3.7, CBMF results are more accurate and reliable than other matrix factorization-based techniques. Specifically, CBMF achieved the lowest RMSE (1.2023) and MAE (0.9768). These results, reinforced by the NDCG analysis (Figure 3.5), highlight the method's aptitude to generate relevant and well-ranked recommendations, benefiting from the comprehensive use of embedded contexts.
The in situ evaluation, involving 1978 users across two archaeological parks, further validated the practical effectiveness and user-centered impact of the system. As illustrated in Tables 3.8 and 3.9, the system received positive feedback from the proposed questionnaire. Notably, the recommendation component reached a satisfaction rate of over 90%, with specific indicators such as Suitability, Path Appropriateness, and Reliability reflecting the users' recognition of the system's personalization capabilities. Moreover, the dialogue system showed elevated levels of interaction fluency and user satisfaction (88.68%–89.84%). The consistency of high satisfaction scores over time (Table 3.10) suggests that the system is not only effective but also scalable and sustainable in real-world deployments. This longitudinal stability is a strong indication of the system's robustness across seasonal variations in user behaviour, crowding patterns, and environmental conditions. Overall, the proposed framework successfully meets its stated goals, demonstrating how IoT-enabled contextual data acquisition, coupled with a lightweight and effective recommendation strategy, can drive meaningful personalization in cultural heritage settings. The convergence of quantitative accuracy and real-world user satisfaction underscores the effectiveness of a hybrid evaluation methodology, which combines algorithmic rigour with empirical validation.
Thus, from the results obtained in the two experimental phases, not only was a recommendation system based on SVD and machine learning capable of providing accurate and reliable predictions introduced but in the transition to the case study, users benefited from the framework that exploits it. Specifically, we can summarize the goals achieved by the Cultural Experience Engine as follows:

1. The framework captures context and user preference data to provide personalized cultural routes to cultural users.

2. Cultural heritage has been enhanced through improving the user's enjoyment of the cultural user experience. The framework makes the experience more interactive at cultural sites and has received positive user feedback.
3. The proposed recommendation technique, CMBF, obtains better results with respect to the comparison methods based on the metrics applied to the collected data.

Chapter 4

Predictive Maintenance of cultural artefacts

The conservation of cultural heritage is a challenge and a duty of humanity to preserve the memory and evidence of past cultures. As a result, over the years, researchers and experts in the field have pursued the demand for effective and efficient strategies to preserve such historical treasures. Recent advancements and novel strategies are currently playing a crucial role, improving many fundamental aspects of the conservation phase such as monitoring and predictive maintenance [94]. Additionally, the possibility of integrating several technologies and approaches allows a significant support in the conservation of cultural assets. In fact, acquiring data through smart sensors according to the Internet of Things (IoT) paradigm [95], the capacity to elaborate data by employing Artificial Intelligence (AI) [110], and the ability to define a Digital Twin (DT) of the asset for analyzing possible risk scenarios, are all key tools to develop a framework aimed at the predictive maintenance of cultural heritage [98]. In literature, several works exploit these three steps for the maintenance task, aiming at monitoring the asset health and predicting possible situations of risk or damage for the asset [187, 153]. Furthermore, integrating AI and IoT in a Digital Twin framework have proved to be a winning strategy, with many applications in several fields (such as structural health monitoring, healthcare, and manufacturing) [21, 54].
This work not only aims to introduce a novel application of AI and IoT integration in a DT framework, but also enhances its effectiveness and reliability by combining physical knowledge and data to obtain more accurate and trustworthy simulations. In the field of Artificial Intelligence, Machine Learning (ML) focuses on enabling systems to automatically learn patterns and make predictions or decisions by analyzing data. Furthermore, in recent years, a new branch of ML has emerged to investigate the potentiality of this multidisciplinary approach: Scientific Machine Learning (SciML). SciML combines

computational and computer science to develop machine learning methods capable of tackling complex physical problems characterized by multiscale dynamics, sparse data, and high-impact decisions [128, 130]. Its strength lies in integrating at different levels the robustness of physics-based models, providing predictive capability, interpretability, and domain knowledge. This novel paradigm opens new possibilities and challenges, especially in the cultural heritage field, where the expertise about the materials and the phenomena that affect the deterioration of the assets represents invaluable information that restoration experts can share to improve the conservation task. Therefore, a framework for designing the analysis of deterioration phenomena is a first crucial step for significantly improving cultural heritage maintenance. It follows that to provide reliable inferences, both physics and data are necessary to build a way to automatically analyze the digital replicas of cultural assets. Therefore, in a DT scenario, the framework also requires defining standards for their analysis, usually characterized by different shapes and irregularities.
With these motivations, this Chapter describes the components of the proposed framework that can automatically process digital replicas of cultural assets and elaborate on reliable simulations based on data-driven and physical-based approaches. The framework developed requires the definition of four functional and task-specific layers involving: (i) data acquisition (information acquired from sensors and APIs or related to digital replicas), (ii) knowledge-based for storing and pre-processing data, (iii) data elaboration to provide reliable simulations, and (iv) services to expert users for cultural heritage restoration, as shown in Figure 2.2.
The automatic process of digital replicas requires interacting with tools to manage the 3D models related to the acquired digital version, for instance, through laser scanning or photogrammetry [126]. To obtain a broad impact and effective customization, our framework exploits open-source solutions and develops strategies to connect 3D models automatically with the knowledge base and the elaboration phase.
The latter integrates data and physics by taking advantage of Physics-Informed Neural Networks (PINNs), a deep learning approach able to learn directly from the physical laws and easily incorporate data [135]. Moreover, depending on the problem at hand, the proposed framework also enables to exploitation Reduced Order Models for many-query and real-time evaluations, providing data from classical numerical methods such as Finite Element (FE) methods to PINNs.
The main objective of this Chapter consists of describing the components of the framework aimed to improve the conservation and analysis of 3D cultural assets by employing and integrating novel approaches based on scientific machine learning. We highlight the following main contributions:

- We introduce a novel methodology for processing 3D models of cultural artifacts for the elaboration phase, providing rapid and reliable simula-

tions to users via an integrated framework.
- We exploit PINNs to combine physical knowledge with data-driven approaches to preserve cultural assets.
- We integrate ROM and PINNs to efficiently solve problems related to the predictive maintenance of cultural heritage that depends on parameters such as material, weather conditions, or external factors.

4.1 Related works

The conservation of cultural artefacts and buildings represents a challenge for researchers who have found in novel technologies a strategic ally.
Over the years, several approaches have been developed to protect cultural heritage starting from the digital transformation, allowing the use of recent techniques for conservation, documentation, and management. Therefore, Digital Twin (DT), Internet of Things (IoT), Artificial Intelligence (AI), and 3D models such as Building Information Model (BIM) for building [28], or Heritage BIM (HBIM) for cultural structures [182], are key tools to exploit in this field. Specifically, DT has a growing relevance in the cultural heritage field to monitor degradation, schedule restoration interventions, and predict possible future damages. This scenario is evident from several works in literature that exploit DT, taking advantage of 3D models by integrating sensor data and AI. Still, these technologies are not limited to them.
It is possible to classify the developed workflows in three classes, focusing on the specific objectives and by means of numerical modelling and structural simulation, integration of HBIM, AI, and IoT, or visual documentation, and digital conservation. The classification provided below highlights the heterogeneity of applications in the literature, underlying their strengths and limits.
The first class consists of workflows based on numerical simulations (specifically, FE method) integrated with the DT paradigm to simulate the behaviour of structures, bridges and monuments. The objective consists of predicting possible risk scenarios and monitoring the behavior in the presence of environmental stresses. These frameworks take advantage of 3D models and the physical knowledge, and require the integration of Structural Health Monitoring (SHM) [177].
Shabani et al. [151] provides a workflow for developing DTs of historical architectonic structures to analyze vulnerability and support strategies for reducing damage risks. In this case the DT, intended as a numerical model suited with physical properties of the building, allows simulations related to the structural behavior through the FE analysis of the meshed 3D models. The paper's objective consists of documenting, preserving, and managing the architectural heritage, and it requires 3D modelling through CAD or BIM [116]. Instead, Zhang et al. [189] introduce a case study based on the conservation of the Great

Wall, focusing on the site of Beichakou. In this case, the DT is developed as a dynamic and integrated platform for merging data and digital models based on a multilevel decision-making process aimed at monitoring, predicting risks, and planning strategies to improve the conservation. The process has four levels of interest: data collection, model construction, plan simulations, and value inheritance. Therefore, conservation actions are organized around five key functional areas: (i) heritage status assessment through real-time monitoring and IoT systems; (ii) optimization of conservation planning at national, provincial, and local levels, (iii) risk monitoring and intervention strategies using FEM-based simulations, (iv) presentation and public engagement via Augmented and Virtual Reality, and (v) multifaceted system assurance to coordinate stakeholders, data sources, and monitoring tools effectively. Rios et al. [80] focus on the role of DT in managing and monitoring bridges through a systematic review. The DT is introduced as a virtual replica of the real bridge developed through BrIM (Bridge Information Modeling), FE method, and sensors data. The DT aims to evaluate potential hazards through simulation, integrating anomaly detection algorithms based on Machine Learning. Therefore, the work analyses several strategies describing input data employed, typologies of algorithms, and applications. Among the described approaches, Rios et al. described the usage of Convolutional Neural Networks (CNNs) for crack detection, Bayesian methods for updating FE method with real data, and recurrent CNN for semantic segmentation of images. Finally, Dabiri et al. [52] introduce a case study based on the structural monitoring of Vittoriano in Rome, integrating real data acquired from satellites, FE analysis, and a Machine Learning regression method for time series to predict the vertical shifting of the building.

The second class includes workflows that integrate semantic models based on HBIM, environmental sensors, and AI. In such cases, the DT represents a dynamic and adaptive system that exploits the elaboration of a significant amount of data to guarantee the management and conservation of cultural heritage. In addition, these workflows require dealing with complex urbanistic scenarios with a quantitative analysis of risks. Li et al [94] analyze three aspects related to the virtual reconstruction and dynamic simulation, the immersive digitalization exploiting Virtual Reality, Metaverse, and Gamification, towards improving the enjoyment of cultural assets. The work focuses on heritage conservation by analyzing disaster cycles and proposing ML methodologies related to the three phases of disaster: before, during, and after. Sebouti et al. [150] introduce a workflow for conserving African cultural assets with a case study based on the Bab Al-Mansour Gate in Meknes, Morocco. The DT is exploited as an interactive digital replica in which predictive ML models, sensor data, and HBIM cooperate. Specifically, the proposed workflow exploits Neural Networks, Random Forest, Support Vector Machine, and Linear

Table 4.1. Summary of the discussed works, excluding review papers and surveys. The columns Internet of Things (IoT), Machine Learning (ML), and Physics indicate the presence of the respective technologies in each study.

Authors	Objective	IoT	ML	Physics
Shabani et al. [151]	A workflow for Digital Twin of historic buildings, with FEM simulations on 3D models for structural analyses and conservation strategies.	✗	✗	✓
Zhang et al. [189]	A dynamic Digital Twin for the conservation of the Great Wall, integrating data collection, FEM simulations and immersive technologies for monitoring and planning.	✗	✗	✓
Dabiri et al. [52]	A case study on the structural monitoring of the Vittoriano in Rome, combining satellite, FEM and ML data to predict vertical displacements over time.	✗	✓	✓
Sebouti et al. [150]	A workflow for African heritage conservation with an interactive DT that integrates ML/DL, sensors, and HBIM for degradation prediction and environmental analysis.	✓	✓	✗
Angheluță et al. [10]	Angheluță et al. develop a DT to document and protect Romania's wooden churches, integrating environmental data and platforms for interactive mapping and scientific repositories.	✓	✗	✗
Kong and Hucks [84]	Kong and Hucks propose a DT divided into five components to monitor the degradation of historic structures, focusing on high-fidelity 3D documentation.	✗	✗	✗

Regression for the degradation prediction, classifying risk levels, and analyzing environmental factors. In addition, a Bayesian approach aims to regulate the dynamic interaction between physical and virtual components.

Finally, the third class includes papers that analyze the digital documentation of cultural heritage through non-invasive conservation, enhancement, and monitoring. These works introduce DT as a visual model that stores information about rural and isolated sites with limited resources to improve accessibility and enjoyment. Angheluță et al. [10] exploit a digital replica of the heritage of Romanian wooden churches integrating environmental data. The DT aims to document, analyze degradation, and plan restoration actions to protect the churches threatened by rural abandonment and environmental degradation. Two platforms are described: the first represents a visual-scientific inventory and interactive map for prioritizing intervention, the second consists of a scientific repository for 3D images and models, with data overlays. Kong and Hucks [84] propose the employment of DT for monitoring historical structures degradation, dividing the DT into five parts: Physical part, Virtual part, Dataset, Service, and Connections. Their objective consists of creating a high-fidelity 3D documentation and detecting degradation.
All previously discussed references are reported in Table 4.1. Specifically, the table underlines how none of the analzsed works simultaneously integrates sensor data, data-driven approaches, and physical knowledge for cultural heritage conservation. Instead, this work aims to define a framework for combining IoT, Deep Learning, and physics-based approaches to improve the reliability of simulations and guarantee the speed of predictions.

4.2 REDUCED ORDER MODELS AND PHYSICS-INFORMED NEURAL NETWORKS

As evidenced by the literature overview, the lack of frameworks to integrate IoT, data-driven, and physics-based approaches represents a limit for conserving cultural heritage. Therefore, this work aims to fill this gap via SciML, combining data-driven and physical-based approaches and, in this context, this Section shortly introduces two of the fundamental novel tools employed by the proposed framework: Reduced Order Models (ROMs) and Physics-Informed Neural Networks (PINNs).

4.2.1 REDUCED ORDER MODELS

Integrating physical knowledge implies dealing with differential problems and their numerical approximation. In addition, the physical behavior is strictly related to specific conditions associated with the cultural assets at hand, such as their material and environmental conditions. Therefore, considering parametrized PDEs is fundamental to reproduce the general framework of cultural heritage conservation, generalizing the analysis and take advantage of

tools that can improve evaluation efficiency.
The resolution of parametric differential problems requires a significant computational effort, leading to methods that can reduce the computational cost, the Reduced Order Models (ROMs) [24, 25]. Among these methods, we highlight Reduced Basis (RB) approach [71, 131, 140], exploiting the information from a set of high-fidelity solutions, the so-called snapshots, to construct a low-dimensional space onto which performing a Galerkin projection, allowing for efficient approximations for unseen values of the parameters. These methods exploit the offline-online paradigm. The offline phase entails the expensive computations and snapshot data collection, also taking advantage of High Performance Computing facilities. On the contrary, the online phase fully exploits dimensionality reduction strategies built on top of the collected data, enabling efficient evaluation in the many-query and real-time context.
The reduced space is determined through the Proper Orthogonal Decomposition (POD), based on the Singular Value Decomposition [71, 131] of the dataset. This method allows compressing and extracting the most relevant information from the snapshots, providing suitable *principal directions* for expressing the parametrized solutions.
To set up the notation, let us consider a differential problem in the domain $\Omega \subseteq \mathbb{R}^n$ based on the parametrized PDE:

$$\mathcal{A}\left[u\left(\mathbf{x}, t; \mu\right), t; \mu\right] = 0, \qquad \mathbf{x} \in \Omega, \tag{4.1}$$

$$u : \Omega \times [0, T] \times \mathbb{P} \mapsto \mathbb{R}^m, \quad \mu \in \mathbb{P} \subset \mathbb{R}^D,$$

where $\mathcal{A}$ is the operator defining the PDE including temporal and spatial differential terms, and D represents the number of parameters, with the following appropriate boundary and initial conditions

$$\mathcal{B}\left[u\left(\mathbf{x}, t\right), t\right] = 0, \qquad \mathbf{x} \in \Gamma = \partial\Omega, \qquad u : \Omega \times [0, T] \mapsto \mathbb{R}^m, \tag{4.2}$$

$$\mathcal{I}\left[u\left(\mathbf{x}, 0\right), 0\right] = 0, \qquad \mathbf{x} \in \Gamma = \partial\Omega, \qquad u\left(\mathbf{x}, 0\right) : \Omega \mapsto \mathbb{R}^m. \tag{4.3}$$

For the numerical discretization we focus on the classical FE methods [133], deriving the weak formulation of the problem, written in the abstract form as:

$$a\left(u, v; \mu\right) = L\left(v; \mu\right), \qquad \forall v \ \in \mathbb{V} \tag{4.4}$$

where a is a linear/non-linear bilinear form, L a linear form including the forcing terms, and v are test functions belonging to a suitable function space $\mathbb{V}$. A similar derivation can be obtained for time-dependent problems, e.g. by identifying an appropriate temporal discretization of N time points $0 = t_0 < t_1 < \cdots < t_{N-2} < t_{N-1} = T$ and using a suitable numerical method, such as the Runge-Kutta methods.

More specifically, the FE approximation, obtained discretizing the weak formulation (4.4), is exploited during the offline phase to compute the snapshots for a fixed set of M parameter $\{\mu_1, \ldots, \mu_M\} \subset \mathbb{P}$ to obtain a dataset describing the variability of the system's solution w.r.t. the parameter setting. Then, we obtain a sampling $\{u(\mu_1), \ldots, u(\mu_M)\}$ of the discrete version of the solution manifold $\mathcal{M} = \{u(\mu) : \mu \in \mathbb{P}\}$ based on high-fidelity solutions.
A fundamental principle in reduced-order modeling is that the solution set can be well approximated within a low-dimensional subspace. This means that a small number of well-chosen basis functions, called the reduced basis, can represent the full solution space with a small approximation error. Given the reduced basis $\{\xi_i\}_i^M \subset \mathbb{V}$, the reduced space is defined as

$$\mathbb{V}_{\mathrm{rb}} = \operatorname{span}\{\xi_1, \ldots, \xi_k\} \subset \mathbb{V}. \tag{4.5}$$

For any parameter $\mu \in \mathbb{P}$, the reduced solution $u_{\mathrm{rb}} \in \mathbb{V}_{\mathrm{rb}}$, obtained as the linear combination of the reduced basis $\{\xi_i\}_{i=1}^k$ as

$$u_{\mathrm{rb}}(\mu) = \sum_{i=1}^{k} \alpha_i(\mu)\, \xi_i, \tag{4.6}$$

where the coefficients $\alpha_i(\mu)$ are uniquely determined by enforcing the reduced form of (4.4) given by

$$a(u_{\mathrm{rb}}, v_{\mathrm{rb}}) = f(v_{\mathrm{rb}}; \mu), \qquad v_{\mathrm{rb}} \in \mathbb{V}_{\mathrm{rb}}. \tag{4.7}$$

Notably, the reduced solution requires a much lower computational effort while retaining a high level of accuracy, only assuming a low intrinsic dimensionality of the solution manifold.
During this work, the construction of the reduced basis is obtained by performing the POD on the matrix of solution snapshots $S \in \mathbb{R}^{N_h \times M}$, and the most important modes, selected via a thresholding argument based on the retained energy, are used to approximate the solution for new values of the parameter $\mu \in \mathbb{P}$ as a linear combination in terms of the computed modes. Specifically, the POD-space represents the k-dimensional space that minimizes

$$\sqrt{\frac{1}{M} \sum_{i=1}^{M} \inf_{v_{\mathrm{rb}} \in \mathbb{V}_{\mathrm{rb}}} \|u(\mu_i) - v_{\mathrm{rb}}\|_{\mathbb{V}}^2}. \tag{4.8}$$

This formulation concludes the construction of the reduced-order approximation, which provides an efficient and accurate surrogate for the high-fidelity model. It establishes the foundation for its application in parametrized and computationally demanding scenarios.

4.2.2 *Physics-Informed Neural Networks*

For application purposes, the resolution of parametric PDEs requires investigating and identifying parameters that fit the specific cultural asset and environmental condition. Therefore, to enable the applicability of the proposed framework in real-life contexts, a key feature comes from embedding novel and Machine Learning (ML) enhanced strategies for discovering parameter values from data knowing the governing equations [63, 127]. Since physical knowledge represents an added value to the framework, we exploit a recently proposed Neural Network (NN) architecture called Physics-Informed Neural Networks (PINNs), allowing for the simultaneous reconstruction of some field of interest (direct problem), and the identification of characteristic unknown parameters, e.g. the so-called inverse problem setting [77, 26]. This way, the network allows to integrate the physical knowledge in the training phase and can be hybridized with data-driven approaches.
Let us consider the problem without the parameter dependency. For both direct and inverse problems involving the approximated continuous solutions of the PDE in Equation (4.1), A PINN consists of a neural network depending on the weights $\mathbf{w}$ that recovers the approximation of the solution in the input points. Therefore, the training of the networks requires a sampling step to obtain the r_Ω collocation points

$$\left\{\left(\mathbf{x}_1^{(\Omega)}, t_1^{(\Omega)}\right), \ldots, \left(\mathbf{x}_{r_\Omega}^{(\Omega)}, t_{r_\Omega}^{(\Omega)}\right)\right\} \subset \Omega \times [0, T], \tag{4.9}$$

the r_Γ spatial boundary points

$$\left\{\left(\mathbf{x}_1^{(\Gamma)}, t_1^{(\Gamma)}\right), \ldots, \left(\mathbf{x}_{r_\Gamma}^{(\Gamma)}, t_{r_\Gamma}^{(\Gamma)}\right)\right\} \subset \Gamma \times [0, T], \tag{4.10}$$

and r_0 points for the initial condition

$$\left\{\mathbf{x}_1^0, \ldots, \mathbf{x}_{\mathbf{r_0}}^{\mathbf{0}}\right\} \subset \Omega. \tag{4.11}$$

Then, the training of the networks consists of minimizing the loss function given by

$$\begin{aligned}\mathcal{L}(\mathbf{w}) = &\frac{1}{r_\Omega}\sum_{i=1}^{r_\Omega}\left\|\mathcal{A}\left[u\left(\mathbf{x}_i^{(\Omega)}, t_i^{(\Omega)}\right), t_i^{(\Omega)}\right]\right\|^2 + \\ &\frac{1}{r_\Gamma}\sum_{i=1}^{r_\Gamma}\left\|\mathcal{B}\left[u\left(\mathbf{x}_i^{(\Gamma)}, t_i^{(\Gamma)}\right), t_i^{(\Gamma)}\right]\right\|^2 + \\ &\frac{1}{r_0}\sum_{i=1}^{r_0}\left\|\mathcal{I}\left[u\left(\mathbf{x}_i^0, 0\right), 0\right]\right\|^2\end{aligned} \tag{4.12}$$

with respect the weights vector $\mathbf{w}$, including the physical knowledge by directly exploiting the differential problem, where $\|\cdot\|$ represents a suitable norm. In

case of a time independent differential problems, the loss (4.12) does not integrate the component associated with the initial condition. Solving inverse problems requires also the integration of physical or geometrical parameters in the vector of weights $\mathbf{w}$ defining the neural network. These weights are then optimized and updated during the training phase accordingly to the physical knowledge incorporated in the loss function (4.12). Toward this goal, the neural network can easily integrate supervised terms, e.g. when sensor measurements are available or when imposing a known function as a boundary condition. It is worth noting that, by including additional data-informed components to the global loss function, the training procedure benefits from additional information, but in general the optimization task is more difficult due to the complex loss landscape, and weighted sum strategies could be applied.
Indeed, already from their first introduction [89, 135], PINNs have reached a significant interest in the community, with several recent improvements related to two main issues: the loss balancing and the causality. The first affects the learning efficacy since different terms in the loss function are associated with Neural Tangent Kernel (NTK) eigenvalues of different magnitudes [175]. Since the NTK spectrum determines how quickly each error component decreases, significant eigenvalue disparities lead to imbalanced convergence: some terms are minimized rapidly, while others stagnate. This stiffness in the training dynamics undermines the effectiveness of gradient descent, ultimately reducing the accuracy of the learned solution. The second issue is related to the imposition of the principle of causality in the learning of the approximate solution for time-dependent differential problems, meaning that local variations in the initial or boundary conditions of a spatio-temporal dynamical system influence its subsequent states over time [172].
Several approaches dealt with the loss imbalance issue, aimed at improving the learning capability of NNs. Yu et al. [185] propose Gradient PINNs (GPINNs) based on integrating gradient information in the loss function to reduce loss fluctuation in the training phase. McClenny and Braga-Neto [107] introduce Self-Adaptive PINNs (SA-PINNs) employing additional self-adaptive weight in the loss function that are maximized at the training point where the loss is bigger. Zeng et al. [188] proposed Competitive PINNs (CPINNs), which replace the classical squared-residual loss with a game-theoretic minimax formulation. In this approach, a discriminator network learns to detect the PDE and boundary violations made by the PINN, while the PINN is trained to minimize them, achieving higher accuracy and faster convergence across linear and nonlinear PDEs. Zou et al. [194] proposed an ensemble PINN framework to capture multiple solutions of nonlinear differential equations, a challenge where standard PINNs typically converge to a single mode. The approach systematically uncovers diverse stable and unstable solutions by exploiting random initialization and the deep ensemble method. Moreover, realistic PINN outputs

can be used as initial guesses for conventional solvers (Finite Difference Methods, FEMs, Spectral Element Methods), establishing a general and efficient strategy for addressing solution multiplicity. Finally, Anagnostopoulos et al. [8] proposed Residual-Based Attention PINNs (RBA-PINNs) by exploiting an attention scheme coming from Transformer architectures [179] that evaluates the residual of the differential problem at each collocation point.
Trying to respect the temporal causality has also led to different advancements of the original PINN approach. Wang et al. [172] introduced Causal PINNs, a reformulation of the loss function with temporal weights that enforce the causal structure of time-dependent PDEs. By ensuring that residuals at later times are minimized only after earlier ones are resolved, this approach corrects the NTK-induced bias of standard PINNs. Instead, in Valentino et al. [168], the authors introduce a Step-by-Step Time Discrete PINNs based on the integration of an iterative scheme typical of classical numerical methods to obtain a semi-discretization of the time interval and enforce the learning from the initial condition of the problem (see Chapter 5).

4.3 3D Model Module: processing geometries to provide simulations

The architecture described in Chapter 2 and reported in Figure 4.1 only in the Predictive Maintenance task, is characterized by the possibility to work with very complex 3D geometries via the 3D Model Module of the Acquisition Layer.
For this purpose, the architecture exploits the API provided by Blender[1], an open-source software for creating and managing 3D content. The application of Blender regards multiple fields, such as three-dimensional modeling, animation, rendering, and physical simulation. Blender is a cross-platform software that allows access to the modeling phase both by experts and via Python scripts. In addition, as mentioned above, it is possible to exploit APIs via the bpy library [160], enabling a seamless and easy interaction with the geometries.
The architecture exploits Blender's API to acquire the information of interest after an appropriate pre-processing of the model. In fact, after a phase in which the model is triangulated, information about points, faces, scales, location, and the list of normals to the faces is acquired. The sub-modules of the architecture then use this information to integrate it into the PINA library [47], a Python package developed for PINNs and more broadly SciML, via the developed tool bl2pina and by automatically generating the mesh of the domain for the differential problem at hand through bl2msh.

[1] https://www.blender.org/about/

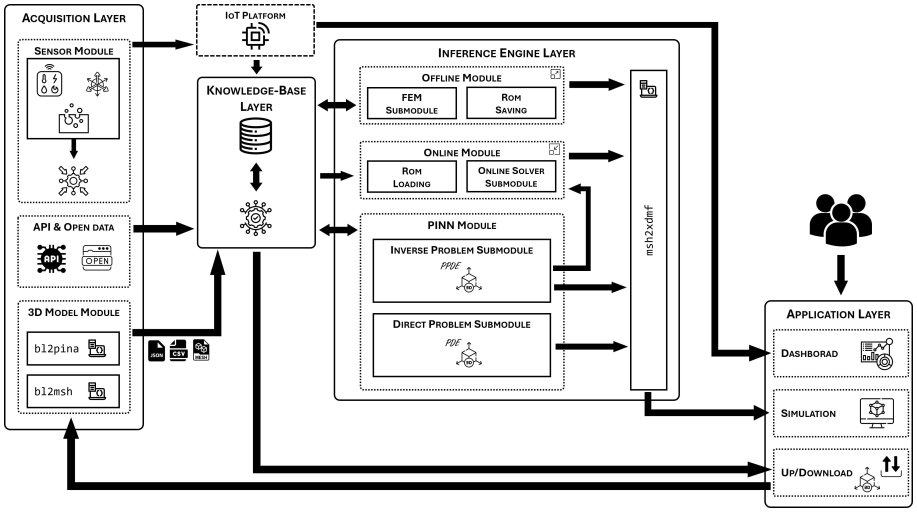
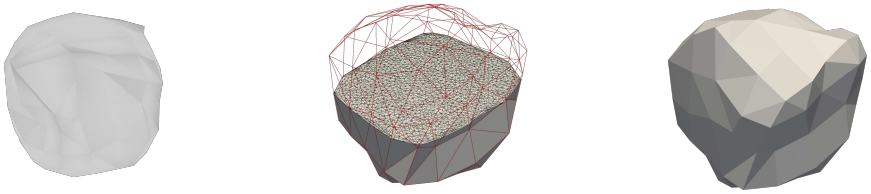


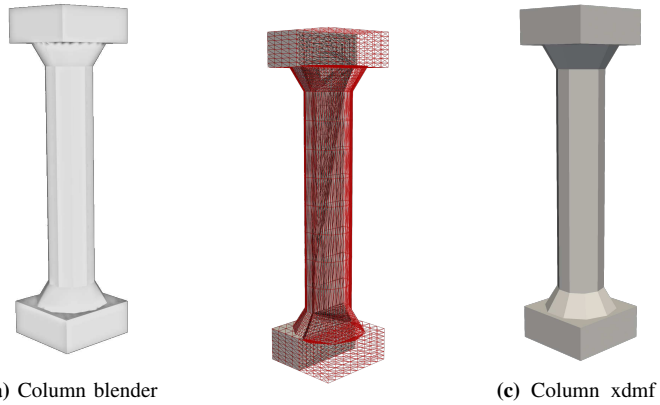
Figure 4.1. The architecture associated with the proposed framework for the predictive maintenance task consisting of four functional layers: the Acquisition Layer, the Knowledge-Base Layer, the Inference Engine Layer, and the Application Layer.



(a) Rock blender model **(b)** Rock mesh **(c)** Rock xdmf file

Figure 4.2. Acquisition of the 3D model related to the rock represented in Figure. The 3D model (a) is acquired and elaborated to provide to the Knowledge-Base Layer the list of collocation and boundary points for PINNs. In addition, the framework elaborates the mesh (b) and prepares the visualization through an XDMF file (c).

In particular, the acquisition of the blender model exploits an organization of the model information according to semi-structured data in a key-value format that allows the model summary to be exported in JSON files. The key-value structure allows, on the one hand, integrating the model information both within the PINN and the ROM workflows. For the PINN workflow, the model includes a description of the domain's boundary, so it is easier to describe all processes requiring the acquisition of data both on the boundary and within the domain. The triangulation of the boundary is obtained via Blender's API, while PINA library allows the integration of the model for the definition of the domain for the PINN problem. However, the architecture integrates the services of PINA by defining integration and sampling strategies, also managing the cases in



(a) Column blender model

(b) Column mesh

(c) Column xdmf file

Figure 4.3. Acquisition of the 3D model related to the column.

(a) Temple blender model

(b) Temple mesh

(c) Temple xdmf file

Figure 4.4. Acquisition of the 3D model related to the temple.

which the 3D model is composed by multiple 3D objects, by creating a list of key-values structures.
Thus, the proposed architecture automatically generates meshes through the 3D models starting from such list of key-values structures and exploiting the information acquired from the Blender APIs. The blender API also allows the proposed framework to elaborate the model to sample the digital replica of the asset. The sampled points, exploited to elaborate a simulation through PINNs, are an alternative to mesh points and are stored in tabular format in the Knowledge-Base layer.
To test of the efficacy of the developed strategies in integrating the Blender model for PINNs and ROMs purposes, we show in Figures the 3D models described in Figures 4.2, 4.3, and 4.4: a rock, a column, and a temple[2]. These structures show the performance of 3D Model Module for increasing level of difficulty in their acquisition. In fact, the rock solely consists of a unique 3D object, the column consists of three 3D objects (the basis, the central part, and

[2]The 3D models for the rock and the column have been acquired from: https://free3d.com/3d-model/low-poly-rock-4631.html and https://free3d.com/3d-model/white-column-44873.html, respectively. The 3D model of the temple is obtained as a re-elaboration of the model available at the following link: https://free3d.com/3d-model/temple-57751.html

the top component), while the temple consists of nine 3D objects (eight columns and the top part). Figures 4.2b, 4.3b, and 4.4b demonstrate the robustness of the architecture to automatically acquire the object from the Blender file and produce an accurate mesh for all benchmarks.
Finally, managing 3D models also allows the architecture to generate XDMF (eXtensible Data Model and Format) and HDF5 (Hierarchical Data Format version 5)files via the msh2xdmf module. XDMF and HDF5 files are often used together to represent and manage complex scientific data, usually coming from numerical simulation, to store large datasets, and within applications where a structured data representation is required. In particular, an HDF5 file is a high-performance binary format that allows large amounts of structured data to be stored in a hierarchical structure-based manner, with multi-language compatibility, high storage capacity, and the possibility of compression. On the other side, an XDMF file is an XML-based metadata format designed to describe complex scientific data, and it is often used as an index pointing to an HDF5 file for the actual data. The idea is to separate metadata (description) from numerical data (content). Specifically, the msh2xdmf module enable the access to the visualization of the simulation elaborated by employing such architecture.

4.4 NUMERICAL RESULTS

To validate the efficacy of the proposed workflow, we set up here an experimental phase deploying numerical simulations for simulated benchmarking scenarios involving physical problems defined on realistic domains of potential interest for the control and monitoring of cultural heritage assets. In particular, we test the performances of the integrated environment, from the data-acquisition to the numerical prediction, in both settings in which:

- physics is known and enforced via PINN, and the goal is to identify the physical parameters by exploiting the Offline-Online Modules in combination with the Inverse Problem one;
- we use the PDE Solver to obtain an efficient evaluation of the PDE for known parameter values.

In both cases, the objective consists of evaluating the accuracy of the simulations obtained to testify the effectiveness of the proposed architecture in supporting expert users for preventive maintenance of cultural assets.

4.4.1 OFFLINE-ONLINE MODULES WITH INVERSE PROBLEM SUBMODULE

The experimental evaluation of the architecture exploiting ROMs for the cultural heritage maintenance requires testing specific modules of all architecture's layers, as highlighted in Figure 4.5.

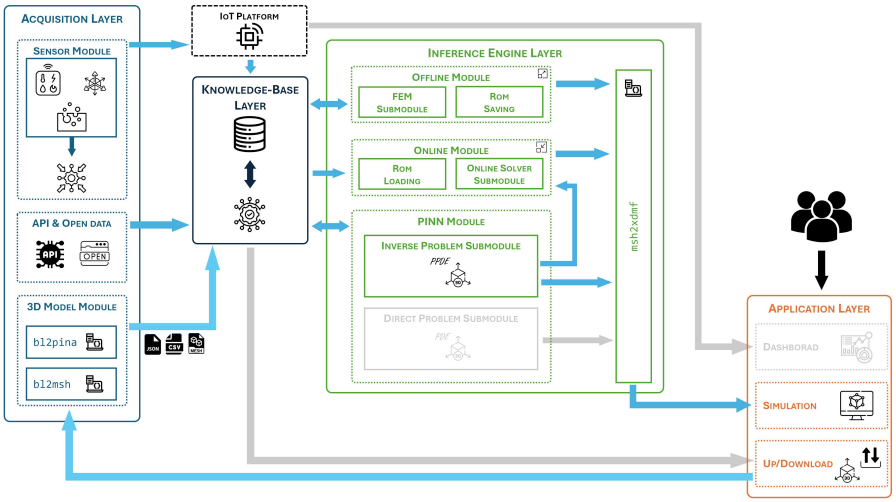


Figure 4.5. Highlighted are the active modules of the architecture to perform the numerical approximation of parametrized differential problems.

The Blender model, loaded through the Up/Download Module, allows the acquisition of the digital replica of the cultural asset. Therefore, the 3D Model Module elaborates the Blender file to obtain a JSON file in which the main features of the domain are collected. This JSON file permits the automatic elaboration of the mesh of the cultural asset, the random sampling of the domain to acquire the boundary and collocation points, and the preparation of the XDMF file that will be employed to visualize the elaborations (see Figure 4.6).
In addition, the Sensors module and the API & Open Data Module, communicating with the IoT platform, allow the acquisition of data related to the phenomenon, providing collected data to the database in the Knowledge-Based Layer. This data allows for solving inverse problems and identifying parameters related to the parametric field configurations as solution to PDEs.
Therefore, the Inference Engine Layer can analyze the parametric PDE by activating the Offline Module, obtaining reliable reference solutions to apply the Proper Orthogonal Decomposition as a Reduced Order Model. Specifically, the user can select the POD energy tolerance through the Simulation Module to control its reliability, otherwise, a pre-defined tolerance of $1e-6$ is applied. Similarly, the Simulation Service allows choosing the level of exploration of the high fidelity model, i.e. the number of sampled snapshots, whose default value is equal to 100. We remark that this number has to be carefully chosen according to the complexity of the parametric space and the computational budget available during the offline phase.
The Offline Module stores the information needed for the ROM Module in the

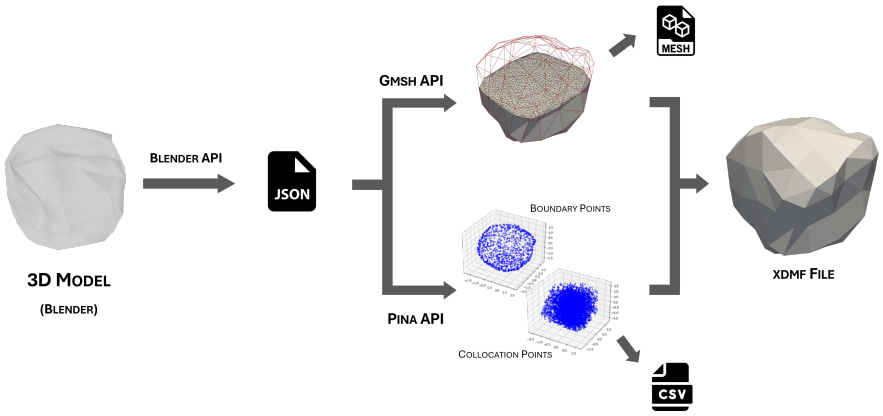


Figure 4.6. Steps of the model elaboration performed by the 3D Model Module. The main features of the model in the Blender file are collected through a JSON file. From the JSON file, the framework exploits the GMSH API [61] to automatically elaborate the mesh of the model exported in an msh file, and the PINA API [47] to acquire boundary and collocation points. In addition, the framework elaborates the xdmf file in which the numerical solutions elaborated through the Inference Engine Module are integrated to visualize the simulation.

database through the ROM Saving submodule. This step takes advantage of the RBniCS package for Python, allowing the integration of the FEniCS package for the PDE resolution via the Finite Element Method by exploiting the FEM Submodule and the application of the reduced strategies.
Then, the Inverse Problem Submodule in the PINN Module take advantage of the physics underlying the analyzed phenomenon, integrating this with the collected data, to identify the parameters of the simulated PDE. In this phase, the high-level of customization of the framework allows the user to choose if apply the PINN approach on the mesh points' coordinates or if randomly select points in the domain defined by the 3D model. Therefore, the user have access to both the solution provided by PINN and the solution obtained via the POD executed by the Online Solver Submodule of the Online Module.
Finally, the msh2xdmf Module elaborates on the simulation and stores it in an XDMF file built on the available mesh.
To perform the experimental phase related to this architecture, we elaborated simulated data related to benchmark test problems. Moreover, we evaluated the Acquisition Layer's ability to analyze three different 3D models related to different structures. The test problems are described as follows. After introducing the parametric PDE under investigation, we describe the acquisition of the 3D model, the process to obtain simulated data, and the accuracy performance, by comparing the ROM procedure with full-order and analytical solutions. As concerns the reliability of the PINN in solving the problem, we postpone the discussion to subsection 4.4.2

Test Problem 1: Poisson problem on a rock

Let us consider a Poisson problem defined on a domain $\Omega \subseteq \mathbb{R}^3$ represented by the rock with shape shown in Figure 4.2a, and the following parametric elliptic PDE

$$\begin{cases} \Delta u\left(x,y,z\right) = -\left(\alpha^2+\beta^2\right)\pi^2\lambda x\cos\left(\alpha\pi y\right)\sin\left(\beta\pi z\right) & (x,y,z)\in\Omega, \\ u\left(x,y,z\right) = \lambda x\cos\left(\alpha\pi y\right)\sin\left(\beta\pi z\right) & (x,y,z)\in\partial\Omega, \end{cases} \tag{4.13}$$

where the function $u : \Omega \cup \partial\Omega \mapsto \mathbb{R}$ represents the solution of the problem (4.13), λ, α, and β are the physical parameters defining the forcing term, and the analytical solution, constructed by the method of manufactured solutions, is given by

$$u\left(x,y,z\right) = \lambda x\cos\left(\alpha\pi y\right)\sin\left(\beta\pi z\right), \qquad (x,y,z)\in\Omega. \tag{4.14}$$

The application of the FEM Submodule of the Offline Module requires the derivation of the weak form

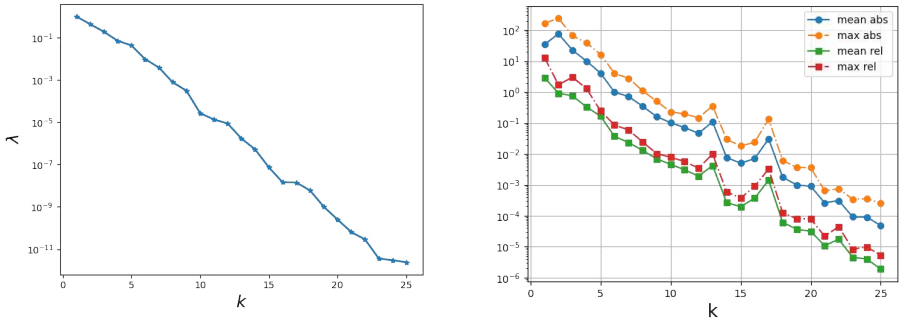
$$a\left(u,v\right) = L\left(v\right) \tag{4.15}$$

of the problem (4.13), where the components are defined as follows:

$$a\left(u,v\right) = -\int_\Omega \nabla u\cdot\nabla v\,\mathrm{d}x, \tag{4.16}$$

$$L\left(v\right) = -\int_\Omega \Big(\left(\alpha^2+\beta^2\right)\pi^2\lambda x\cos\left(\alpha\pi y\right)\sin\left(\beta\pi z\right)\Big)v\,\mathrm{d}x. \tag{4.17}$$

The application of the FEM to the problem (4.13) exploits a mesh of $N_h = 42343$ nodes, and the polynomial basis functions to obtain the form (4.15) is the first-order Lagrange finite element.
The POD procedure then requires the definition of a set of snapshots $\{(\lambda_i,\alpha_i,\beta_i) : i = 1,\dots,M\}$, with $M = 100$, and the FEM Submodule, that allows to numerically solve the problem for each sample $(\lambda_i,\alpha_i,\beta_i)$, defining the matrix $S \in \mathbb{R}^{N_h\times M}$, where N_h represents the number of mesh points (see Figure 4.2b). From the POD we extract the reduced basis of order k that allow us to project the system in a lower dimensional space and obtain a reduced solution in a more efficient way. In Figure 4.7a, we show the singular value decay denoting the "reducibility" of the problem for $k \leq 25$, meaning the dimensionality of the most important modes to be retained to obtain an accurate reconstruction of the reduced approximation.
Selecting $k = 25$, and running the ROM Saving Submodule, we save the reduced basis and assemble the relevant reduced operators, including their



(a) Singular Values decay

(b) Error Analysis

Figure 4.7. Decay of the singular values and error analysis, respectively left and right, obtained from the ROM Submodule for the Poisson equation in Test Problem 1.

storage in the database through the connection between the Offline Module in the Inference Engine Layer and the database in the Knowledge-Base Layer. Moreover, the Submodule also can perform the error analysis to investigate and a-posteriori estimation of the accuracy of the model when a different amount of basis function is employed for the dimensionality reduction.
Figure 4.7b introduces the analysis of the singular values and the errors on the selected modes. Specifically, Subfigure 4.7a describes the behaviour of the singular values normalized according to the infinite norm and related to the matrix $S \in \mathbb{R}^{N_h \times M}$. Instead, Subfigure 4.7b describes the behaviour of the mean and maximum absolute and mean and maximum relative errors by adding modes in the procedure and comparing the full-order with the reduced-order solution. This analysis exploits 10 testing snapshots randomly sampled. The error has an exponential trend, indicating excellent performance even for a low-dimensional basis. However, the figure shows some noise for the modes 13, 16, and 17, where the errors increase instead of decrease, underlying some instability related to the mesh and the analysed problem.
When sensor data are available, one could be interested in the identification of a specific tuple of parameter (λ, α, β) that are related to the phenomenon under investigation. Specifically, we exploit simulated data obtained by choosing the potentially unknown parameter sample $(\lambda, \alpha, \beta) = (0.1, 0.2, 0.5)$ to discover the properties of the physical model starting from randomly chosen points in the domain Ω.
Indeed, the parameter identification task requires solving an inverse problem through the employment of Physics-Informed Neural Networks. The PINN solver selected is the Residual-based Attention PINN [8], based on the Residual Feed Forward neural network [174]. Specifically, the details of the exploited networks are in Table 4.2.
The Table introduces the number of sampling points exploited to solve the

Table 4.2. PINN hyperparameters for Test Problem 1.

Detail	Value
Collocation Points	200
Boundary Points	50
Data Points	500
Epochs	3000
Batch size	50
Learning rate	$1e-3$
Decay rate	$1e-8$
Optimizer	Adam
Network structure	$[3, 400, 400, 1]$

inverse problem through the PINN. The collocation points are 200, sampling the domain's interior; the boundary points exploited to force the known physics on the domain's boundary are 50; finally, the Data Points represent the data collected through simulated sensors on the boundary of the rock.
During the training, the Inverse Problem Submodule of the PINN Module learns an approximation of the physical parameters, obtaining a relative accuracy reported in Table 4.3.

Table 4.3. Results obtained resolving the inverse problem through the PINN. The Table reports the approximated and the expected parameters, and the relative errors.

Parameter	Approximated Value	Expected Value	Relative Error
λ	0.1006	0.1000	5.8391e-03
α	0.2024	0.2000	1.2022e-02
β	0.5032	0.5000	6.3606e-03

In addition, Figure 4.8 shows the value of parameters identified during the training process.
Finally, the Online Module load the reduction method through the ROM Loading Submodule for the identified parameter value and solve the analyzed problem by exploiting the Online Solver Submodule.
The msh2xdmf Module allows the construction of the XDMF file supported by a h5 file, enabling the visualization and inspection of the results. Indeed, the module provide the files to the Simulation Module of the Application Layer to visualize the simulation of the phenomenon. Moreover, the msh2xdmf can provide the simulation of the error comparing the reduced solution obtained through the Online Solver Submodule with the full order solution obtained through the FEM Module.

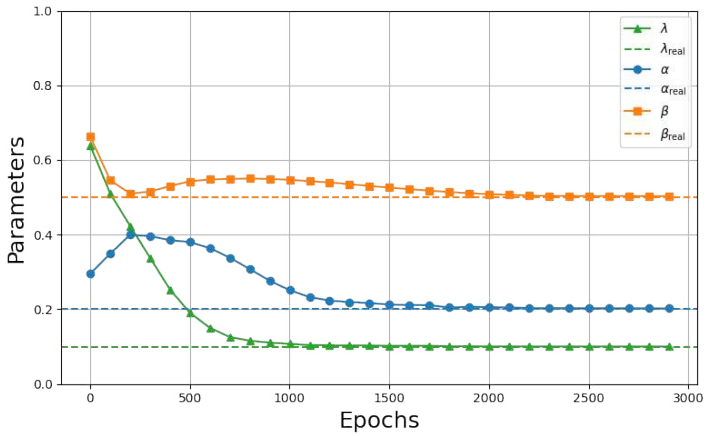
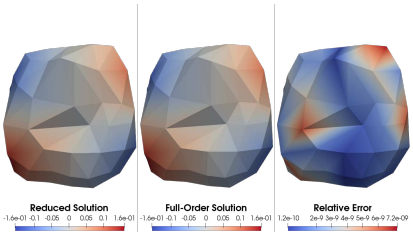
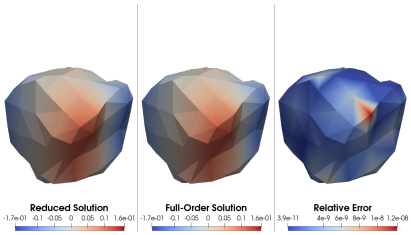


Figure 4.8. Convergence for parameter values λ, α, β during the training phase compared with the expected values $\lambda_{\text{real}}, \alpha_{\text{real}}, \beta_{\text{real}}$.

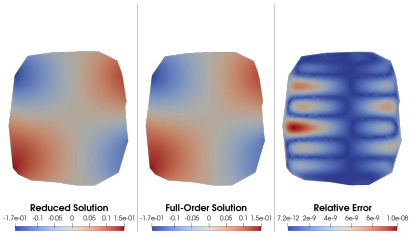
We show in Figure 4.9 the reduced solution, the full order solution, and the relative error on boundary and on a slice of the domain.



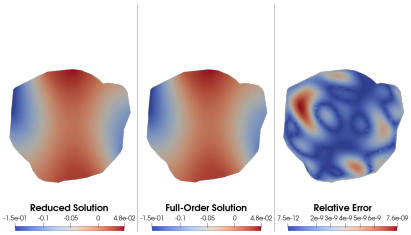
(a) Boundary - Front view



(b) Boundary - Isometric View



(c) Slice - Front view



(d) Slice - Isometric View

Figure 4.9. Comparison between the reduced and the full order solutions for Test Problem 1 on the boundary and on a slice with different views. In addition, the Figures show the relative errors obtained from the comparison.

In particular, the reduced solution is computed using the approximated parameters obtained from the Inverse Problem Submodule, rather than the exact ones($(\lambda, \alpha, \beta) = (0.1, 0.2, 0.5)$). This choice is aimed at evaluating the ro-

bustness of the entire framework, not only the ability of the physics-informed inverse model to infer the governing parameters reliably, and of the reduced-order solver to reconstruct the physical state through the online phase, but also the overall capability of the methodology to represent real-world phenomena accurately. Finally, the total relative error introduced with the FE approximation and the subsequent reduction approach with respect to the analytical solution in (4.13) is only $7.77e-3$, and it is mostly coming from the first discretization itself relative error with respect the full-order solution is $4.5594e-07$. We remark that more accurate high-fidelity solutions can be obtained by refining the original mesh obtained from the 3D model, or choosing higher order spaces for the polynomials. This comes at the cost of a more costly offline phase, but the online phase remains unaffected, still providing reliable and real-time approximations.

Test Problem 2: parabolic PDE on a column

For the second problem, we study the solution of a parametric parabolic PDEs defined on the temple domain $\Omega \subseteq \mathbb{R}^3$ showed in Figure 4.4a. The system of equations governing the phenomenon are given by:

$$\mathbf{u}_t\left(x,y,z,t\right) = \Delta\mathbf{u}\left(x,y,z,t\right) + F\left(x,y,z,t,\mathbf{u}\left(x,y,z,t\right)\right), \qquad (4.18)$$

$$\left(x,y,z,t\right) \in \Omega \times [0,1]$$

where the function $\mathbf{u} : \Omega \cup \partial\Omega \mapsto \mathbb{R}^2$ represent the solution of the system of PDEs (4.18), and

$$F\left(x,y,z,t,\mathbf{u}\left(x,y,z,t\right)\right) = \begin{pmatrix} \lambda e^{\lambda t} \\ \lambda e^{\lambda t} - 2\left(\alpha+\beta+1\right) \end{pmatrix}, \qquad (4.19)$$

$$\left(x,y,z,t\right) \in \Omega \times [0,1]$$

with boundary condition

$$\mathbf{u}_b\left(x,y,z,t\right) = \begin{pmatrix} e^{\lambda t} + \alpha x + \beta y + z \\ e^{\lambda t} + \alpha x^2 + \beta y^2 + z^2 \end{pmatrix}, \qquad \left(x,y,z,t\right) \in \partial\Omega \times [0,1] \qquad (4.20)$$

initial condition

$$\mathbf{u}\left(x,y,z,0\right) = \mathbf{u}_0\left(x,y,z\right) = \begin{pmatrix} 1 + \alpha x + \beta y + z \\ 1 + \alpha x^2 + \beta y^2 + z^2 \end{pmatrix}, \qquad \left(x,y,z\right) \in \Omega \qquad (4.21)$$

and analytical solution

$$\mathbf{u}\left(x,y,z,t\right) = \begin{pmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \end{pmatrix} = \begin{pmatrix} e^{\lambda t} + \alpha x + \beta y + z \\ e^{\lambda t} + \alpha x^2 + \beta y^2 + z^2 \end{pmatrix} \in \mathbb{R}^2, \qquad (4.22)$$

$$(x, y, z, t) \in \Omega \times [0, 1] .$$

Test Problem 2 represents a further experiment aimed at testing the proposed framework in the context of a system of time-dependent PDEs. In this case, the experiment, similarly to the previous test problem, aims to apply the POD procedure, identify the parameters that fit the simulated data, and compare the reduced solution with the full-order solution. The presence of the time-dependency requires the application of a Finite Difference scheme, specifically, the implicit Euler method. Therefore, we define the temporal grid

$$0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = 1, \tag{4.23}$$

$$t_i = hi, \quad i = 0, \ldots, N, \qquad h = \frac{1}{N}.$$

In this way, we can perform the iteration in time via the Euler method as

$$u_{n+1}(x, y, z) = u_n(x, y, z) + h f_{n+1}(x, y, z, t_{n+1}, u_{n+1}), \tag{4.24}$$

$$n = 0, \ldots, N - 1,$$

where we have defined $u_n(x, y, z) \approx u(x, y, z, t_n) \in \mathbb{R}^2$ for $n = 1, \ldots, N$, and

$$f_{n+1}(x, y, z, t_{n+1}, u_{n+1}) = \Delta u_{n+1}(x, y, z) + F(x, y, z, t_{n+1}, u_{n+1}(x, y, z)). \tag{4.25}$$

Applying (4.24) to the PDE defined in (4.18) and avoiding to report dependent variables to simplify the notation, we obtain

$$u_{n+1} = u_n + h\Delta u_{n+1} + hF(x, y, z, t_{n+1}, u_{n+1}(x, y, z)), \qquad n = 0, \ldots, N-1. \tag{4.26}$$

After the temporal discretization, we can define the variation form

$$a(u, v) = L_{n+1}(v), \qquad n = 0, \ldots, N - 1, \tag{4.27}$$

where

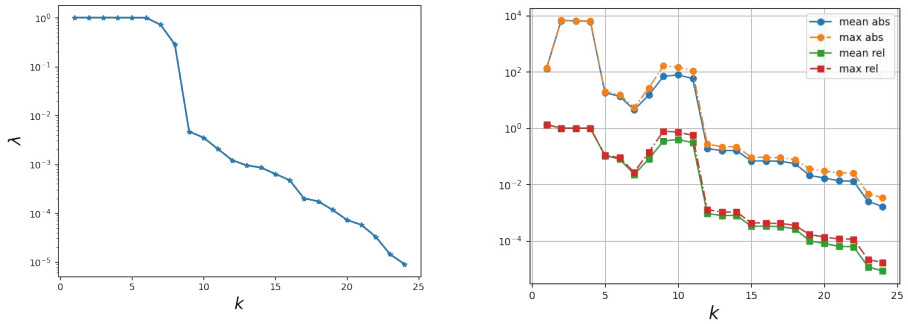
$$a(u, v) = \int_\Omega (uv + h\nabla u \cdot \nabla v)\, \mathrm{d}x, \tag{4.28}$$

$$L_{n+1} = \int_\Omega (u_n + hF(x, y, z, t_{n+1}, u_{n+1}(x, y, z)))\, v \mathrm{d}x, \tag{4.29}$$

$$n = 0, \ldots, N - 1$$

In the time-dependent context, applying the POD procedure requires managing the temporal dependence of the problem. Having discretized the temporal variable according to (4.23), time represents an additional parameter that can be dealt by a nested application of the POD.

Applying the same procedure seen before on the novel matrix, the architecture employs POD to the time-dependent problem (4.18). Therefore, storing the reduction exploits the ROM Saving Submodule as before by selecting $k = 24$. Figure 4.10 shows the singular values of S selected by imposing a tollerance of $1e - 6$ (Figure 4.10a) and the error analysis (Figure 4.10b).



(a) Singular values analysis

(b) Error Analysis

Figure 4.10. Decay of the singular values and error analysis, respectively left and right, obtained from the ROM Submodule for the Poisson equation in Test Problem 2.

Figure 4.10 introduces the analysis of the singular values and the errors on the selected modes. Specifically, Subfigure 4.10a describes the behaviour of the singular values normalized by the infinite norm and related to the matrix that synthesizes the analysis of time and the parameters. Instead, Subfigure 4.10b describes the behavior of the mean and maximum absolute and mean and maximum relative errors by adding modes in the procedure and comparing the full-order with the reduced-order solution. This analysis exploits five randomly sampled testing snapshots. The error has an exponential trend, indicating excellent performance even for a low-dimensional basis. However, the figure shows some noise in the modes from 8 to 11, where the errors increase rather than decrease. This behaviour is related to a decrease of respective singular values, as shown by Figure 4.10a, where the first seven eigenvalues have similar magnitude. Then the quantity of information related to the singular values decreases significantly until the singular value number 9. Therefore, the increase in the error can be interpreted as noise related to the decrease in the quantity of information and the applied mesh. When the decrease in singular values is more uniform, the errors related to it decrease exponentially.
The procedure in the architecture respects the one described for Test Problem 1, in 4.4.1. Table 4.4 describes the hyperparameters of the networks, where the necessity of facing a system of time-dependent PDEs requires the employment of a greater number of collocation and boundary points. In addition, the PINN Module also requires points that reduce the component of the loss function related to the initial conditions. Other hyperparameters increased are the

simulated data and the number of epochs. Instead, the network's structure is similar, except for the input and output layers.

Table 4.4. Details concerning the hyperparameters of the PINN to solve the Test Problem 2.

Detail	Value
Collocation Points	1000
Boundary Points	400
Initial Points	400
Data Points	1000
Epochs	10000
Batch size	None
Learning rate	$5e-4$
Decay rate	$1e-8$
Optimizer	Adam
Network structure	$[4, 400, 400, 2]$

Table 4.5 introduces how the Inverse Problem Submodule approximates the parameters. Specifically, the submodule achieves a relative error of order 10^{-2} for parameter λ and 10^{-3} for parameters α and β.

Table 4.5. Results obtained resolving the inverse problem through the PINN. The Table reports the approximated and the expected parameters, and the relative errors.

Parameter	Approximated Value	Expected Value	Relative Error
λ	0.0960	0.1000	4.0492e-02
α	0.2014	0.2000	7.1611e-03
β	0.5026	0.5000	5.1683e-03

In addition, Figure 4.11 describes how the PINN identifies the parameters through the epochs, showing that 5000 epochs are sufficient to estimate the parameter.
Finally, Figure 4.12 shows results related to the comparison between the Reduced solution obtained with the approximated parameters and the Full-Order solution elaborated on the exacts parameters. The error on the magnitude shows as the reduced simulation achieves at the most an error of order $1e-08$ in the point to point analysis of the relative error. In addition, the simulated solution, compared with the full-order, obtains a norm-2 relative errors of $4.3861e-06$ and $2.4059e-06$ on the first and second components, respectively. Instead, the comparison with the exact solution returns relative errors in norm-2 equal to $2.1636e-02$ and $9.9044e-03$, respectively.

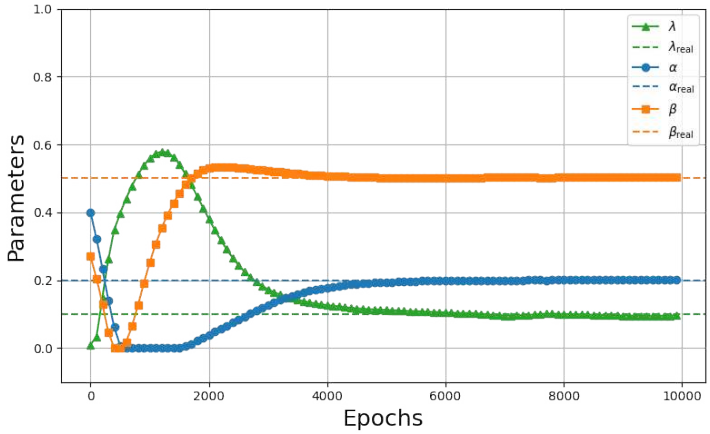
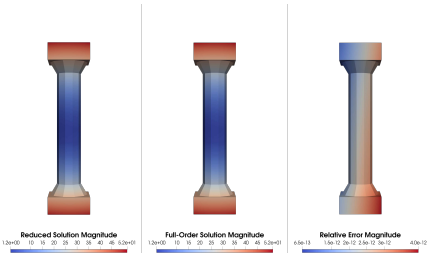


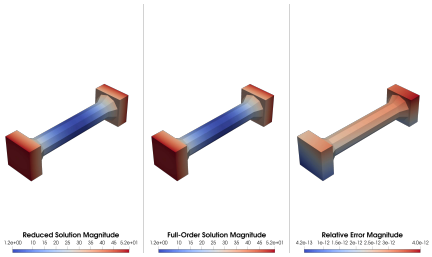
Figure 4.11. Convergence for parameter values λ, α, β during the training phase compared with the expected values $\lambda_{\text{real}}, \alpha_{\text{real}}, \beta_{\text{real}}$.

4.4.2 *Direct Problem Submodule*

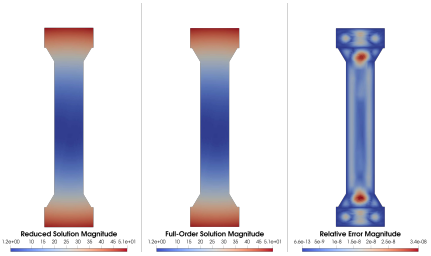
Section 4.4.1 focuses on the analysis of the proposed framework's ability to handle parametrized PDEs by managing the parameters, identifying them via the Inverse Problem Submodule of the PINN Module, and then applying the online phase of the POD. However, not all problems in the cultural heritage field require the analysis of parametric PDEs. Sometimes the parameters are known, and we should solve PDEs. In this case, the proposed framework evaluates the phenomena by applying PINNs for direct problems. In the case of experimentation for the resolution of direct problems in which parameter of PDEs are known, the architecture reacts differently to provide the services related to the simulation to the user. Specifically, Figure 4.13 shows which modules and submodules are activated in this case.
The workflow for acquiring 3D models is similar to the previous case: the Blender model is uploaded by the user via the upload/download module in the Application Layer, activating the 3D Model Module in the Acquisition Layer. As in the previous experiment, the data relating to the model are stored in the Knowledge-Base Layer which also includes the data acquired via the sensors, filtered through the IoT platform, or via API services and Open Data.
The task of the Inference Engine is to solve monitoring problems that use both a physics-based and a data-driven approach. Consequently, on the one hand, there are differential problems in which, unlike the case discussed in Section 4.4.1, the parameters are known. Through the use of PINNs, it is possible to integrate the data acquired either through the API or through the Sensor Module. It follows that the Module of Interest in this experimental step



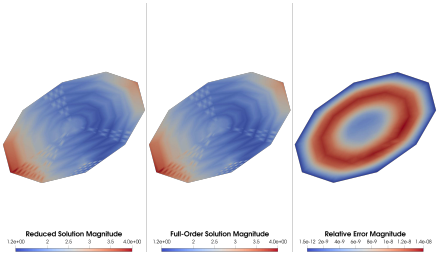
(a) Boundary - $T = 0.5$ - Front view



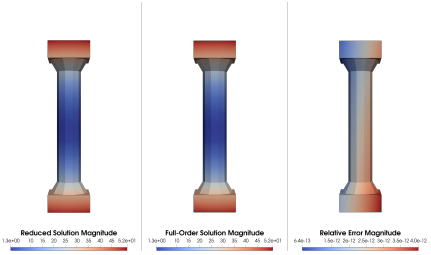
(b) Boundary - $T = 0.5$ - Isometric View



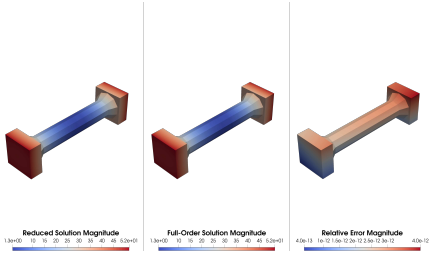
(c) Slice - $T = 0.5$ - Front view



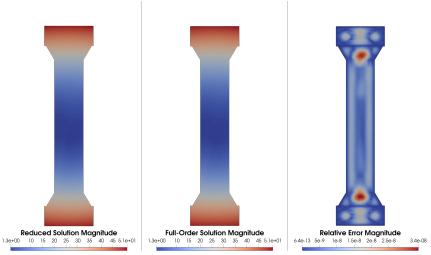
(d) Slice - $T = 0.5$ - Isometric View



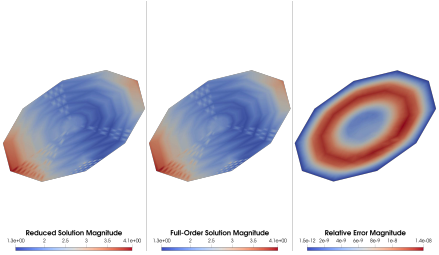
(e) Boundary - $T = 1.0$ - Front view



(f) Boundary - $T = 1.0$ - Isometric View



(g) Slice - $T = 1.0$ - Front view



(h) Slice - $T = 1.0$ - Isometric View

Figure 4.12. Comparison between the reduced and the full order solutions for Test Problem 2 on the boundary and on a slice with different views at time points $T = 0.5$ and $T = 1$. In addition, the Figures show the relative errors obtained from the comparison.

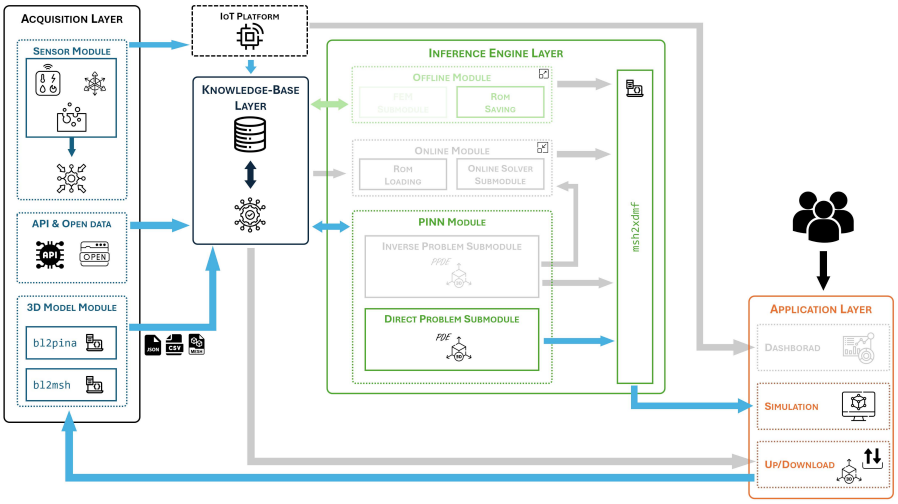


Figure 4.13. Active modules of the architecture related to the proposed framework to perform the resolution of differential problems.

is still the PINN Module, particularly the Direct Problem Submodule, where the processing takes place, allowing the integration of data with PDE-based problems in order to provide simulations to users. For the visualization part, the msh2xdmf Module is used again to process the simulation and store it in an XDMF file.
This experimental step uses benchmark problems in which the data simulate the boundary conditions of the analyzed PDE-based problems. In addition, the employment of the FEM Submodule in the Offline Module allows to obtain the benchmark solution in cases of problems with unknown analytical solution.

Test Problem 3: temperature monitoring through data with PINN on a rock

Let us consider the problem related to the monitoring of temperature of an outdoor cultural assets. This test problem combines the physical-base approach related to the heat equation with a data-driven one consisting of a simulated scenario showed in Figure 4.14 that represents the data simulating the boundary conditions of the heat equation

$$u_t(x,y,z,t) - k\Delta u(x,y,z,t) = 0, \qquad (x,y,z,t) \in \Omega \times [0,1], \quad (4.30)$$

where the domain Ω consists of the rock of Figure 4.2 introduced in Sub Section 4.3, $u : \Omega \cup \partial\Omega \mapsto \mathbb{R}$ represents the solution, and $k = 1$. In addition, the first simulated data represents the initial condition of the problem seen as constant temperature.

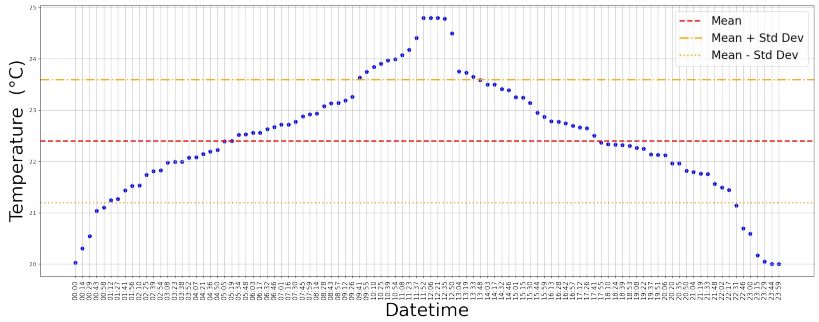


Figure 4.14. Simulated data concerning the temperature on the boundary of the domain during 24 hours for Test Problem 3.

Elaborating the benchmark problem requires the semi-discretization in time the definition of a temporal grid to apply the Euler method, as seen previously for the Test Problem 2. Then, the variation form

$$a\left(u,v\right)=L_{n+1}\left(v\right) \tag{4.31}$$

with the following right and left hand sides

$$a\left(u,v\right)=\int_{\Omega}\left(uv+hk\nabla u\cdot\nabla v\right)\mathrm{d}x,\qquad L_{n+1}\left(v\right)=\int_{\Omega}u_{n}v\,\mathrm{d}x\,. \tag{4.32}$$

is obtained similarly from the semi-discretized problem.
The resolution via the PINN requires the definition of the networks with the hyper-parameters described in Table 4.6.

Table 4.6. Details concerning the hyperparameters of the PINN to solve the Test Problem 3.

Detail	**Value**
Collocation Points	400
Data Points	9900
Epochs	30000
Batch size	None
Learning rate	$1e-3$
Decay rate	$1e-8$
Optimizer	Adam
Network structure	$[3,200,200,1]$

We define a network with 2 hidden layers of 200 neurons and the output represent the temperature $u\left(x,y,z,t\right)$ in the input point at time t. The training phase exploits 9900 simulated data on the boundary, and 400 collocation points.

The boundary consists of fixed 100 spatial points that simulate the sensors installed near the cultural assets analyzed. For each spatial point, we have the temperature at each time of sampling

$$u\left(x_j, y_j, z_j, t_i\right) = T_i, \qquad j = 1, \ldots, 100, \quad i = 1, \ldots, N, \tag{4.33}$$

where $\{(x_j, y_j, z_j) \,:\, j = 1, \ldots, 100\} \subseteq \partial\Omega$ represent the spatial location of sensors on the boundary $\partial\Omega$, and $N = 99$ represents the number of time acquisition of the data, excluding the first one (T_0) that will be employed as initial constant condition.
In this application case, we employ the classical structure of PINNs, as defined in [135], without the employment of the Residual-Basis Attention approach [8]. Therefore, the definition of the loss $\mathcal{L}$ composed by the one related to the residual (4.30) and the one related to the data (4.33)

$$\mathcal{L} = w_{\text{residual}}\mathcal{L}_{\text{residual}} + w_{\text{data}}\mathcal{L}_{\text{data}} \tag{4.34}$$

permits the integration of the weights w_{residual} and w_{data} to manage the information unbalancing. Specifically, in this test problem we fixed

$$w_{\text{residual}} = 0.1 \qquad w_{\text{data}} = 0.9. \tag{4.35}$$

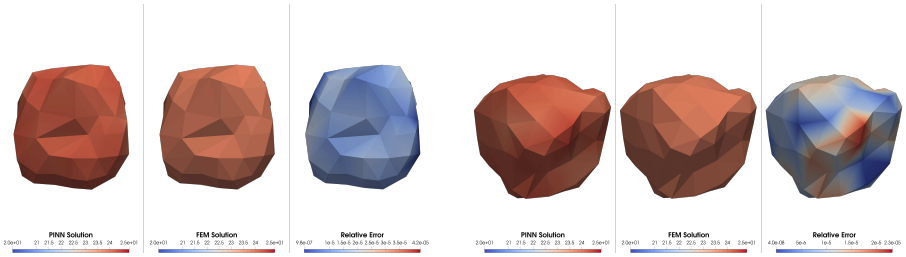
In addition, this test problem requires the application of the hard-constraints [89], indeed, the procedure that allows to satisfy automatically the initial condition of the problem (4.30)

$$u\left(x, y, z, 0\right) = T_0, \qquad (x, y, z) \in \Omega \tag{4.36}$$

where, from the simulated data, $T_0 = 20.02672606321051 = u_0$. A suggested by (4.34), the integration of hard constraints permits to avoid the training of the initial condition by requiring to the network an output designed on T_0. Specifically, the output $\text{out}\left(x, y, z, t\right)$ of the network is adapted as follows

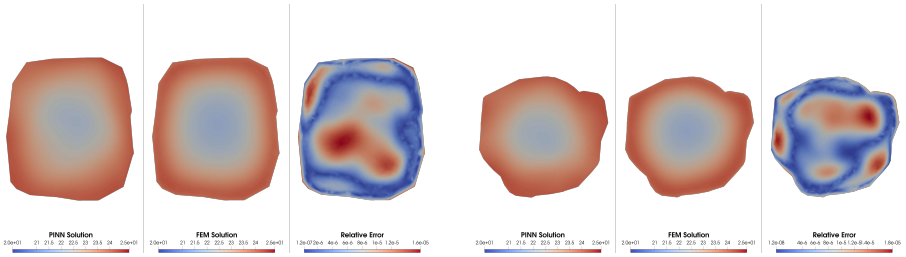
$$\hat{u}\left(x, y, z, t\right) = T_0 + t\ \text{out}\left(x, y, z, t\right). \tag{4.37}$$

Finally, after the training of the network, Figure 4.15 shows the comparison between the solution approximated with the PINN and a benchmark solution elaborated through FEM and the relative error at times $T = 0.5, 1$ on the boundary and inside the domain.
The results shows as the solution approximated with the PINN is reliable compared with the benchmark one, testifying the ability of the PINN to simulated the temperature monitoring imposing a simulated scenario. In addition, the advantage of integrating data with a physics-based approach represent a significant step for exploiting the Internet of Things paradigm and the physical knowledge in an unified workflow. Finally, the norm-2 relative error obtained with respect the benchmark solution is equal to $8.6973e-03$.



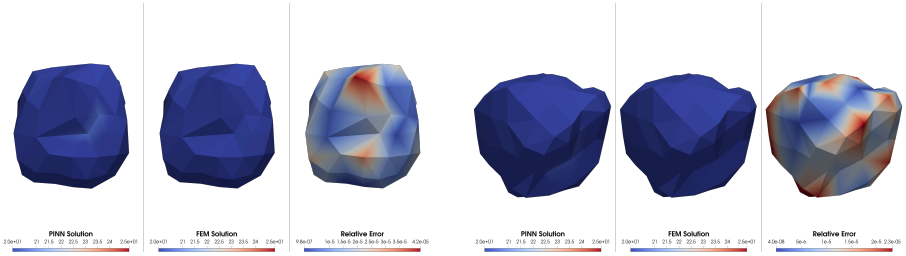
(a) Boundary - $T = 0.5$ - Front view

(b) Boundary - $T = 0.5$ - Isometric View



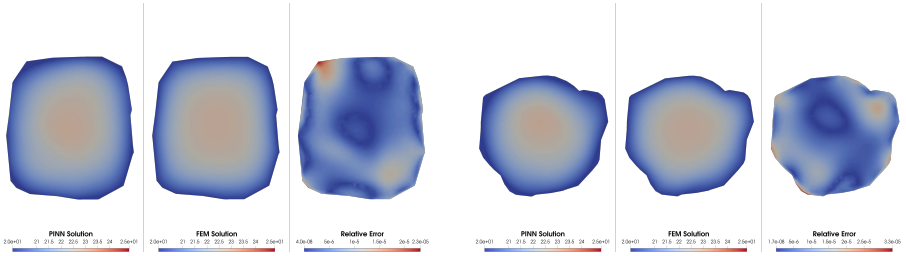
(c) Slice - $T = 0.5$ - Front view

(d) Slice - $T = 0.5$ - Isometric View



(e) Boundary - $T = 1.0$ - Front view

(f) Boundary - $T = 1.0$ - Isometric View



(g) Slice - $T = 1.0$ - Front view

(h) Slice - $T = 1.0$ - Isometric View

Figure 4.15. Comparison between the solution obtained with PINN and a benchmark solution obtained with FEM concerning the Test Problem 3 on the boundary and on a slice with different views at time points $T = 0.5$ and $T = 1$. In addition, the Figures show the relative errors obtained from the comparison.

Test Problem 4: system of PDEs on a column

The test problem 4 evaluates on the domain $\Omega \subseteq \mathbb{R}^3$ showed in Figure 4.3b the following system of Poisson PDEs

$$\Delta \mathbf{u}(x,y,z) = F(x,y,z,\mathbf{u}(x,y,z)), \qquad (x,y,z) \in \Omega \tag{4.38}$$

where $\mathbf{u} : \Omega \cup \partial\Omega \mapsto \left(u^{(1)}, u^{(2)}\right)^t \in \mathbb{R}^2$ represents the solution of the problem (4.38) and

$$F(x,y,z,u(x,y,z)) = \begin{pmatrix} 2u^{(1)} \\ 2 \end{pmatrix} \in \mathbb{R}^2, \qquad (x,y,z) \in \Omega. \tag{4.39}$$

The introduced PDEs have the following boundary conditions

$$u(x,y,z) = \begin{pmatrix} e^{x+y} \\ x^2 - z \end{pmatrix} \in \mathbb{R}^2, \qquad (x,y,z) \in \Omega \tag{4.40}$$

that also represent the analytical solution of the problem. The peculiarity of this test problem consists of the comparison between two different analysis: in the first case we consider a fully-physic approach by defining the residual and boundary losses based on (4.38) and (4.40), instead, the second case evaluated the boundary condition as simulated acquired data, as shown in Figure 4.16.

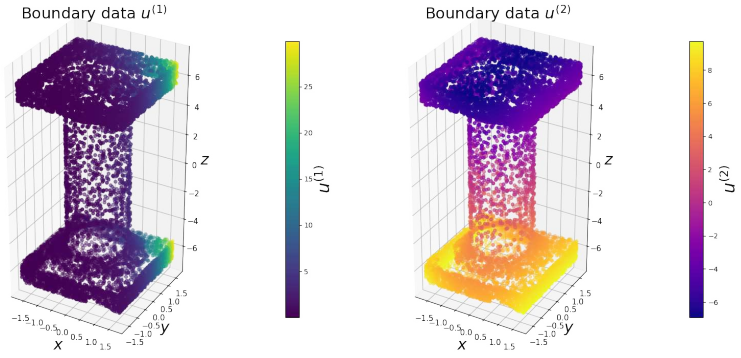
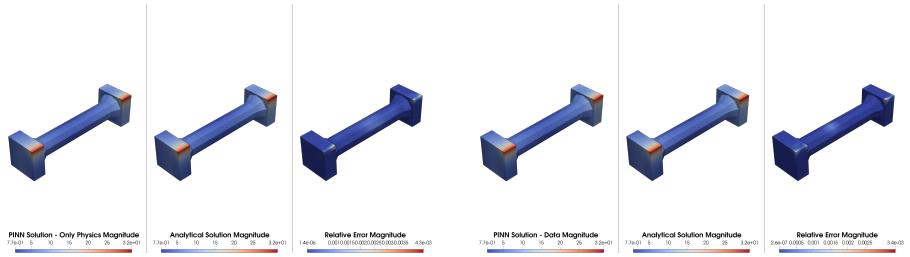


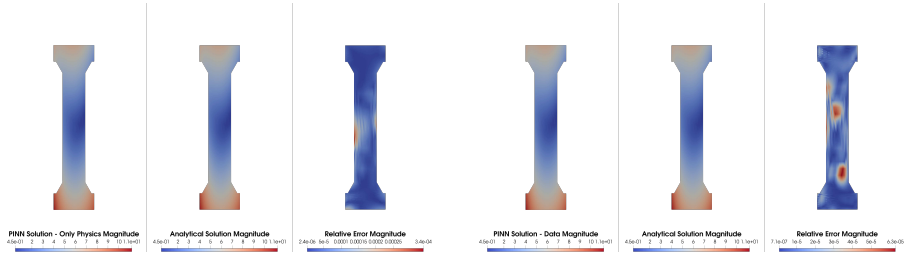
Figure 4.16. Values of boundary data with respect the two components $u^{(1)}$ and $u^{(2)}$.

Table 4.7 resumes the hyper parameters of the network employed that exploits the Residual-Basis Attetion approach [8] integrated with the Stochastic Weight Avareging strategy to improve the optimizer employed [78].
The results obtained are compared with the analytical solution and, as in the previuos cases, the relative error represents the comparison metric.
Figure 4.17 shows the results obtained on boundary and inside the domain with the approach based only on the physics of the problem and by applying a data approach on the boundary.

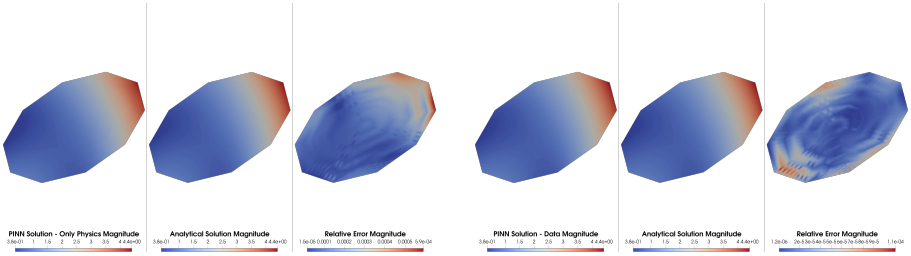
(a) Boundary - Only Physics - Front view **(b)** Boundary - Data - Front View



(c) Boundary - Only Physics - Isometric view **(d)** Boundary - Data - Isometric View



(e) Slice - Only Physics - Front view **(f)** Slice - Data - Front View



(g) Slice - Only Physics - Isometric view **(h)** Slice - Data - Isometric View

Figure 4.17. Comparison between the solution obtained with PINN and the analytical solution concerning the Test Problem 4 on the boundary and on a slice based only on physics and with data integration with several views. In addition, the Figures show the relative errors obtained from the comparison.

Table 4.7. Details concerning the hyperparameters of the PINN to solve the Test Problem 4 based only with physics and with integrating data.

Only Physichs		**Data Integration**	
Detail	**Value**	**Detail**	**Value**
Collocation Points	1000	Collocation Points	1000
Boundary Points	500	Boundary Points	500
-	-	Data Points	500
Epochs	5000	Epochs	5000
Batch size	None	Batch size	None
Learning rate	$1e-3$	Learning rate	$5e-4$
Decay rate	$1e-8$	Decay rate	$1e-8$
Optimizer	Adam	Optimizer	Adam
Network structure	$[3, 200, 200, 1]$	Network structure	$[3, 200, 200, 1]$

In addition, Table 4.8 introduced the errors obtained on components $u^{(1)}$ and $u^{(2)}$ of the solution and on the magnitude of the components.

Table 4.8. Relative errors obtained by the two approaches (only physics and with data integration) on the components $u^{(1)}$, $u^{(2)}$, and the magnitude for the Test Problem 4.

Component	**Relative Error Only Physics**	**Relative Error Data Integration**
$u^{(1)}$	1.1824e-02	7.4181e-03
$u^{(2)}$	4.8771e-03	1.0097e-03
Magnitude	7.5959e-03	4.0997e-03

Results obtained return a comparable results among the only-physics approach and the data-integration, underlying how the employment of data acquired from sensors guarantees the wellness of the employment of PINNs for applications. Specifically, the possibility of easily integrate data and physics accordingly a scientific machine learning paradigm allows a full exploiting of novel technologies to provide reliable simulation aimed to predictive maintenance of cultural heritage. Moreover, the relative errors showed in Table 4.8 returns lower relative errors in the case of the Data Integration approach even if this singular cases represents a simulated scenario.

4.4.3 Discussion

This work's experimental results and architectural design demonstrate the feasibility and potential of a comprehensive framework for the predictive maintenance of cultural heritage assets based on integrating IoT, 3D model processing,

and Scientific Machine Learning based on combining data-driven and physics-based approaches.
One of the key contributions of the work is the development of a modular and automated 3D Model Module, capable of extracting geometric and semantic information from digital replicas of cultural assets. The use of Blender APIs and the conversion into structured key-value formats facilitates interoperability with downstream simulation tools, enabling automatic meshing and domain sampling. This step proves crucial in bridging geometric data and simulation-ready models, and is scalable to increasingly complex structures, as shown in the processing of models with different topological complexities.
The framework exploits both PINNs and ROMs, combining the interpretability and domain-awareness of physical models with the data-adaptive capabilities of neural networks. This dual approach was validated on inverse and direct problems, using benchmark differential problems with varying complexity and structure. The Inverse Problem Submodule successfully identified parametric values with reliable precision, with relative errors on parameters below 10^{-2}. Additionally, ROMs based on POD improve the computational speed of approximations with a satisfying level of accuracy.
In the case of direct problems, the architecture allowed the simulation of dynamic scenarios such as temperature monitoring, where data-driven boundary conditions were integrated with physical constraints. Using hard constraints and weighted losses within PINNs confirmed the ability to balance empirical data and physical knowledge through the differential problem. Particularly notable is the result showing that a data-integrated PINN approach produced comparable or better relative errors than a purely physics-based one in a multi-equation system.
In conclusion, it follows that the objectives set out in the introductory section have been achieved:

- the proposed framework, through the Predictive Maintenance Engine, introduces a methodology for analysing 3D models via the 3D Model Module of the Acquisition Layer. In this module, 3D models acquired via Blender's API are processed, providing the necessary data for training PINNs, the mesh for performing FEM, and consequently applying ROMs, and finally the visualisation of the results via XDMF files.
- Through the PINN module, the proposed framework performs direct and inverse problems by exploiting physical knowledge and data for tasks such as parameter identification and developing simulations on differential problems.
- The proposed framework exploits the combination of PINN and ROM first to identify the parameters inherent to the cultural asset being analysed in the context of the phenomenon being studied and then, through the online phase of ROM, quickly develops a reliable simulation that can

contribute to the conservation of artistic and cultural heritage.
However, some limitations emerged. The architecture depends heavily on the quality and consistency of 3D models and sensor data. While automated mesh generation from Blender was successful, the fidelity of simulations depends on the granularity and accuracy of input data. Similarly, the performance of PINNs depends on the choice of network architecture and training dynamics, especially in time-dependent problems.

4.5 The case study: Furia Selvaggia

This Section introduces a real-world case study designed to validate the proposed predictive maintenance pipeline for a cultural heritage asset subject to environmental degradation. The analysed artefact is *Furia Selvaggia*, a large outdoor steel sculpture by Umberto Mastroianni (1910–1998), installed at the southern entrance of the University of Salerno campus and donated by the artist's heirs in 2001. Created in 1975, the work exemplifies Mastroianni's mature language, where linear, dynamic structures generate fragmented spatial configurations and a strong sense of internal motion, making the sculpture not only a significant artistic landmark but also a complex object from an engineering and conservation standpoint[3].
From a conservation perspective, the sculpture is made of CORTEN steel, a material that develops a protective patina when exposed to suitable conditions [31]. Nevertheless, in outdoor environments, this passivation mechanism may be compromised, leaving the structure vulnerable to localized corrosion. In particular, pitting corrosion may initiate when the protective film partially breaks down, producing highly localized attacks that can accelerate mechanical weakening due to perforation and loss of effective cross-section [73]. For long-term preservation and safety, it is therefore essential to support monitoring activities with models capable of predicting the evolution of corrosion pits and, more generally, the progression of damage over time.
To describe this phenomenon in a physically grounded and differential form, the case study adopts a phase-field formulation for corrosion pitting. The phase-field approach implicitly represents the metal–electrolyte interface via an auxiliary field variable, enabling the evolution of complex pit morphologies without explicitly tracking moving boundaries [100]..
Building on the PINN background introduced in the previous sections, this Section employs the Direct Problem Submodule to solve the governing time-dependent PDE system underlying the phase-field corrosion model. In this setting, PINNs serve as the computational core to approximate solution fields while enforcing differential constraints via a physics-informed loss. The ob-

[3] https://web.unisa.it/vivere-il-campus/campus/arte-e-architettura/furia-selvaggia

jective is twofold:

- to demonstrate that the direct, physics-constrained solution can serve as a reliable engine for forecasting corrosion evolution under observed or assumed environmental conditions;
- to establish the methodological bridge toward predictive maintenance tasks, where model-based simulation, monitoring data, and decision-oriented indicators must be combined in a coherent workflow.

In the remainder of the chapter, the case study is formalized by detailing the geometry and modelling assumptions, the adopted phase-field equations and parameters, and the training strategy used by the Direct Problem Submodule. The results are then discussed in terms of the predicted spatio-temporal evolution of pits, the interpretability of the learned fields, and the practical implications for monitoring and maintenance planning of *Furia Selvaggia* as an outdoor cultural heritage asset.

4.5.1 THE PDE MODEL

In the adopted phase-field formulation for pitting corrosion, the metal-electrolyte interface is represented implicitly through an order parameter $\phi(x,t)$, where $\phi = 1$ denotes the metal phase and $\phi = 0$ the electrolyte. The normalized concentration field $c(x,t) \in [0,1]$ describes the dissolved metal-ions concentration (normalized by c_{solid}). The governing equations are obtained from a free-energy functional and lead to a coupled Allen-Cahn / Cahn-Hilliard-type system. Using the KKS construction and neglecting the concentration gradient energy (i.e., $\alpha_c = 0$) [100], the final PDE system can be written in compact form as:

$$\begin{aligned} \partial_t \phi &= D_\phi \, \Delta\phi + F_1(\phi, c), \\ \partial_t c &= D_c \, \Delta\big(c + F_2(\phi)\big), \end{aligned} \tag{4.41}$$

with

$$\begin{aligned} F_1(\phi, c) &= 2AL(1-c_L)\big(c - (1-c_L)h(\phi) - c_L\big)\, h'(\phi) \;-\; \omega L \, g'(\phi), \\ F_2(\phi) &= (c_L - 1)\, h(\phi), \\ D_\phi &= L\,\alpha_\phi, \qquad D_c = D, \end{aligned} \tag{4.42}$$

where

$$h(\phi) = -2\phi^3 + 3\phi^2, \qquad g(\phi) = \phi^2(1-\phi)^2. \tag{4.43}$$

INITIAL AND BOUNDARY CONDITIONS.

The phase-field system is completed by prescribing initial conditions for both state variables,

$$\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}), \qquad c(\mathbf{x}, 0) = c_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \tag{4.44}$$

Table 4.9. Numerical parameters of the phase-field model.

Parameter	Physical meaning	Value
σ	Interfacial energy	$10\ \mathrm{J/m^2}$
l	Interface thickness	$5\ \mu\mathrm{m}$
D	Diffusion coefficient	$8.5 \times 10^{-10}\ \mathrm{m^2/s}$
L	Interfacial kinetic coefficient	$2\ \mathrm{m^3/(J \cdot s)}$
A	Free-energy curvature	$5.35 \times 10^{7}\ \mathrm{J/mol}$
c_L	Normalized equilibrium concentration	3.57×10^{-2}
α_ϕ	Gradient-energy coefficient (for ϕ)	$3.01 \times 10^{-6}\ \mathrm{J\,m}$
ω	Double-well potential height	$2.08 \times 10^{6}\ \mathrm{J/m^3}$

and mixed boundary conditions on $\partial\Omega$,

$$\phi = c, \qquad \frac{\partial c}{\partial n} = -\kappa\, f_{\mathrm{TOW}} f_{\mathrm{SO}} f_{\mathrm{T}}, \qquad \mathbf{x} \in \partial\Omega, \tag{4.45}$$

where n denotes the outward unit normal and κ is a proportionality factor mapping the corrosion loss into an equivalent concentration flux.

Table 4.10. Coefficients used in (4.46)-(4.47)-(4.48).

Symbol	Value
k_0	0.0018
B	0.98
C	3800
D	0.46
E	25
F	0.62
J	0.016
T_0	20

The terms $f_{\mathrm{TOW}}, f_{\mathrm{SO}}, f_{\mathrm{T}}$ are modelled through an empirical dose-response function derived from the ISOCORRAG[4] [83]. To avoid symbol clashes with the phase-field parameters introduced in the previous section, we write

$$f_{\mathrm{TOW}} = A_{\mathrm{T}} t^{B} \left(\frac{\mathrm{TOW}_{\mathrm{mean}}}{} \right) \tag{4.46}$$

$$f_{\mathrm{SO}} = \left(1 + \frac{\mathrm{SO}_{2\mathrm{mean}}}{E} \right) \tag{4.47}$$

$$f_{\mathrm{T}} = e^{J(T_{\mathrm{mean}} - T_0)} \tag{4.48}$$

[4] https://www.iso.org/standard/66233.html

Here, t is the exposure time (years), TOW is the time of wetness (annual fraction of hours/year), SO_2 is the sulphur dioxide pollution level (as defined in the adopted ISOCORRAG standard), and T is the mean air temperature (°C). The coefficients used in (4.46)-(4.47)-(4.48) are reported in Table 4.10.
In addition, the values on the boundary are obtained on data acquired through REST API from external sources of the years 2019-2025, as shown in Figure 4.18.

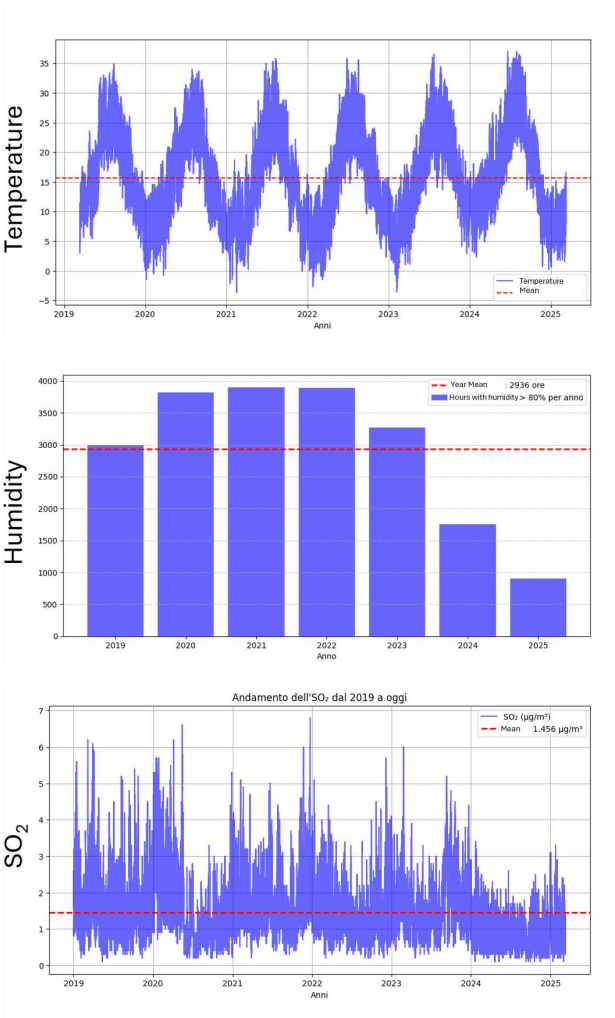


Figure 4.18. Data acquired from external resources to analyse the corrosion phenomenon on *Furia Selvaggia*

4.5.2 *Experimental Phase*

This subsection presents the experimental phase conducted to validate the proposed predictive maintenance workflow on a real cultural heritage asset. The experiments are organised into three stages:

- digital model analysis;
- network identification;
- one-year simulation results and comparison with a real reference image showing qualitative correspondence between predicted corrosion patterns and the current state of the artefact.

Digital Model Analysis

The predictive maintenance architecture employed in this chapter enables the integration of complex 3D geometries through the 3D Model Module of the Acquisition Layer. In the present case study, this capability is essential because *Furia Selvaggia* is sculptural artefact characterised by highly irregular geometry, sharp edges, and cavities. These features make corrosion analysis more challenging than in simplified geometries, as local exposure, runoff, and stagnation zones can induce heterogeneous degradation across the surface.

Figure 4.19. *Furia Selvaggia*

To manage such complexity, the workflow leverages Blender as a geometry processing environment, exploiting its Python API (via bpy) to access and preprocess the digital replica of the artefact. Specifically, Figure 4.19[5] introduce the real artefact, instead, Figure 4.20 shows the digital model obtained.
After model preparation (including triangulation), geometric information relevant for simulation is extracted. This information is then automatically trans-

[5] https://censimentoarchitetturecontemporanee.cultura.gov.it/scheda-opera?id=3078
https://it.wikipedia.org/wiki/File:Furia_selvaggia.jpg

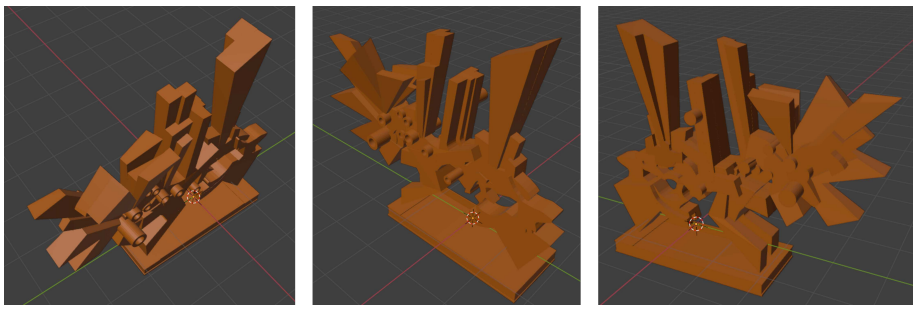


Figure 4.20. Views of the Blender model of *Furia Selvaggia*.

ferred into the PINN workflow through dedicated conversion tools: the geometry metadata is structured in a semi-structured key–value representation, enabling consistent handling of boundaries and multi-object models; moreover, the domain discretisation is supported through automatic generation of mesh and/or sampled point clouds for collocation and boundary constraints. In addition, the workflow supports exporting simulation-ready formats (e.g., XDMF/HDF5) to facilitate post-processing and visualisation of the predicted corrosion fields. Figures 4.21 and 4.22 show this part of the analysis of the digital model that enables the elaboration phase. Specifically, Figure 4.21 presents the obtained mesh and Figure 4.22 represents a view of the `xdmf` file.

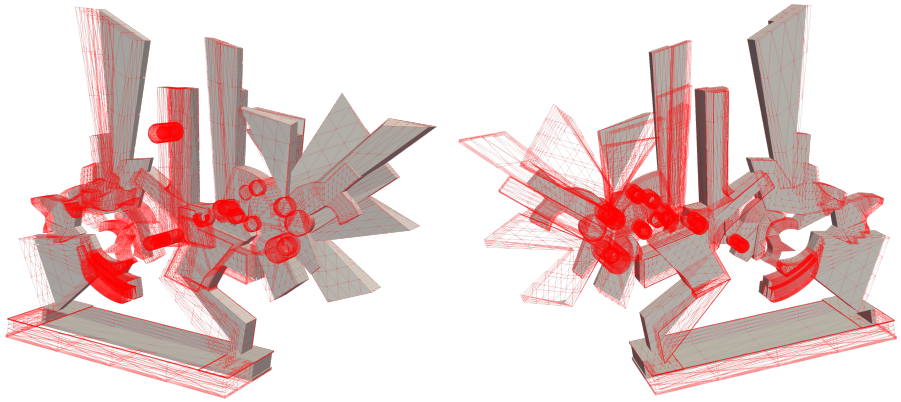


Figure 4.21. Elaborated mesh of *Furia Selvaggia* through the 3D Model Module of the proposed framework.

NETWORK IDENTIFICATION

After defining the domain and the governing PDE model, the next step is to identify a suitable neural architecture for the PINN solver. In particular, a set of candidate networks was evaluated by varying the number of hidden layers and

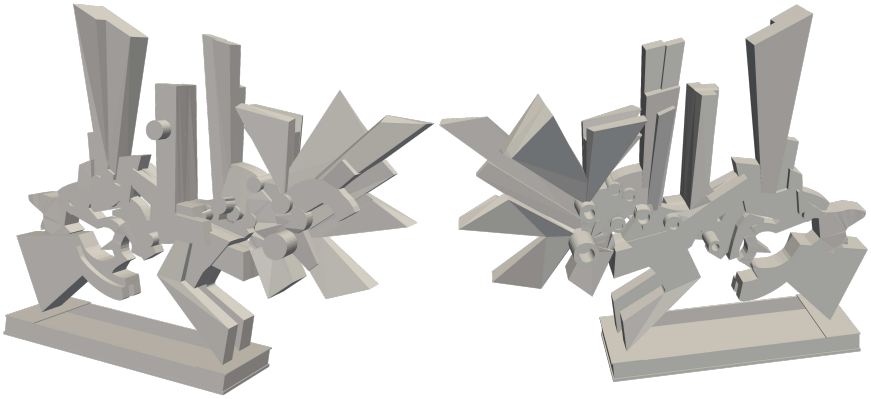
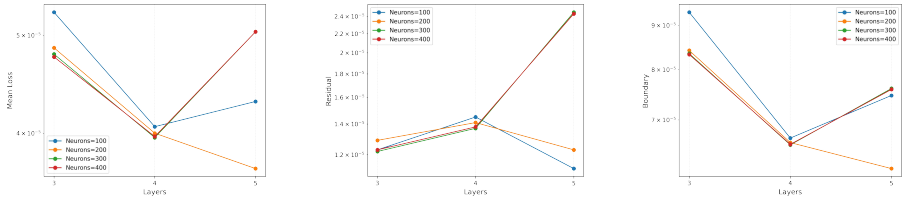


Figure 4.22. Elaborated XDMF file of *Furia Selvaggia* through the 3D Model Module of the proposed framework.

the number of neurons per layer, while keeping the training setup consistent. The comparison was conducted by analysing the resulting error indicators (mean loss, residual loss, and boundary loss) and selecting the configuration that provides the best compromise between accuracy and stability.
Importantly, the selection was not based only on training performance: once the model was trained on the designated training points, the quality of the learned solution was assessed on a separate set of testing points, sampled in the domain after training. This procedure, synthesized in Figure 4.23, ensures that the chosen architecture can generalise the physical constraints beyond the sampled points used for optimisation, which is particularly relevant for irregular 3D geometries such as Furia Selvaggia.



(a) Mean Loss **(b)** Residual Loss **(c)** Boundary Loss

Figure 4.23. Results of training procedure concerning the structure of the PINN to analyse the corrosion model. Subfigure (a) represents the mean loss, (b) the residual, and (c) the boundary loss with respect to testing points.

Simulation

Using the selected network configuration, the model is then employed to perform a one-year simulation horizon of corrosion evolution. This time window

is meaningful for predictive maintenance because it allows

- capturing seasonal and cumulative effects in exposure-driven boundary conditions;
- producing middle term forecasts that can support inspection scheduling and intervention planning.

The simulation outputs are analysed to examine the spatio-temporal evolution of corrosion-related fields, highlighting where corrosion is predicted to initiate or intensify, and how degradation patterns evolve across the artefact geometry. Specifically, the obtained simulation are shown in Figures 4.24 and 4.25 where the several time steps of the simulation are shown.

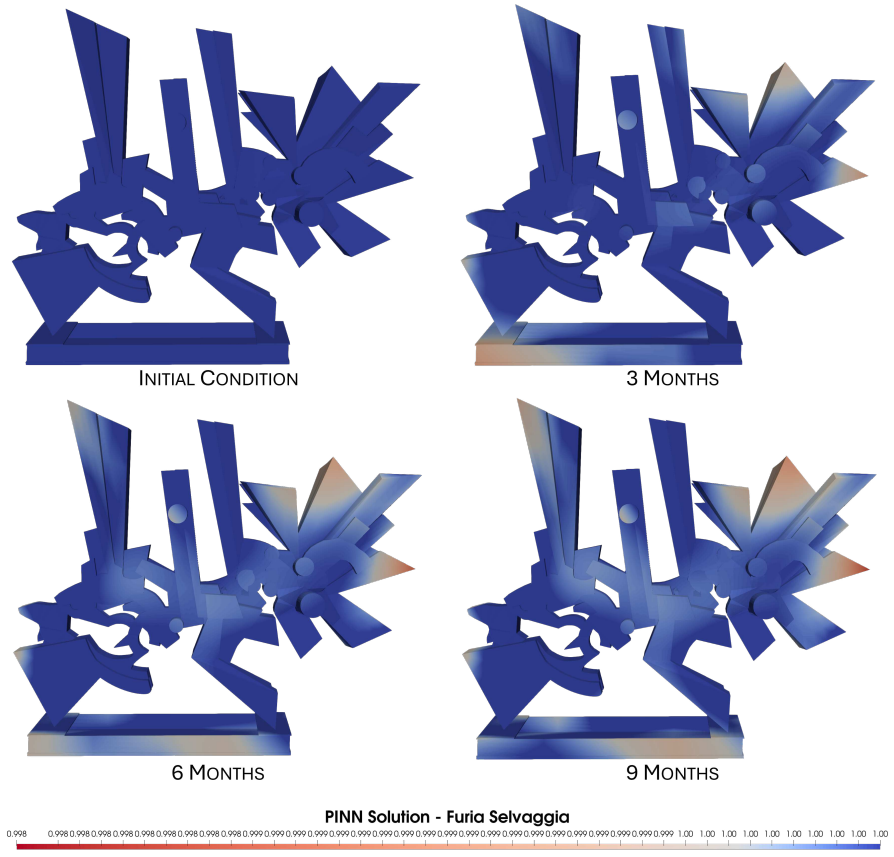


Figure 4.24. Results obtained in the one-year simulation. The Figure shows the obtained results on the value of c for initial conditions and 3, 6, 9 months of simulations

The simulations highlight the progressive development of the corrosion process. Specifically, Figure 4.24 illustrates the evolution from the pristine condition to the state reached under the atmospheric conditions described in Figure 4.18. Notably, corrosion tends to concentrate on the frontal region and on the

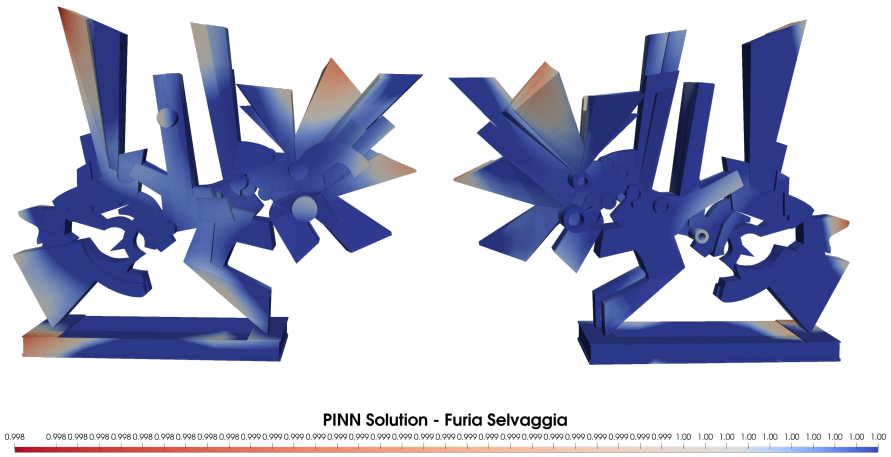


Figure 4.25. Results obtained in the one-year simulation.

upper areas of the artefact, particularly around the head. Moreover, since the phenomenon is characterised by small-scale variations, as indicated by the value bars in Figures 4.24 and 4.25, some inconsistencies emerge in the numerical solution. In particular, the lower portion appears more corroded at three and six months than at nine months. The process becomes more pronounced at the final time instant, as shown in Figure 4.25. This behaviour arises because small oscillations in the solution can produce large colour variations in the visualisation. These results further confirm that the causality issue significantly affects the simulated outcomes; for this reason, Chapter 5 introduces a method to mitigate this limitation. Finally, Figure 4.25 reports the simulation at the end of the analysed year. The previously identified areas remain the most affected, while the artefact's lateral cylindrical elements begin to exhibit early signs of corrosion.

Finally, the predictive value of the approach is discussed by comparing the simulated corrosion distribution with a real image of *Furia Selvaggia*. The objective is not to claim pixel-wise equivalence, but rather to verify whether the simulation reproduces qualitatively consistent corrosion localisation, i.e., whether areas predicted as more prone to corrosion correspond to regions where visible corrosion/patina alteration is observed on the real artefact. In particular, the reference comparison images were acquired after the 2015 restoration and date to 2024. By contrast, the simulation starts from an undamaged material state and reproduces one year of evolution under the environmental conditions described above. As shown in Figures 4.26, 4.27, 4.28, and 4.29, the simulation identifies several areas affected by the corrosive process, which become more apparent to the naked eye after years of exposure. Specifically, the analysis considers the head from both sides and the upper portion of the structure.

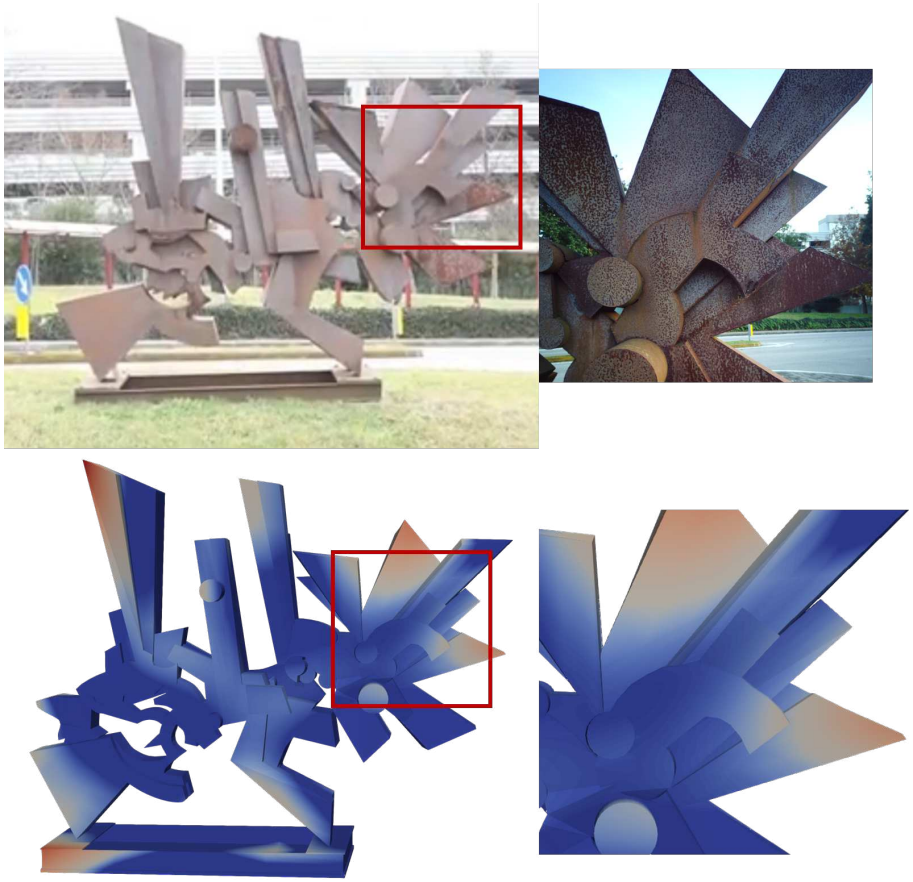


Figure 4.26. Comparison between simulation and real-scenario - Case 1.

Overall, the simulation indicates a higher intensity of the phenomenon in the same regions where evident corrosion is observed.
From a broader perspective, these results also support the effectiveness of the proposed digital workflow. The pipeline enables the automatic acquisition of the digital model, reducing manual intervention and ensuring geometric consistency across analyses. In addition, the framework can ingest contextual information from external resources, which is then integrated into the modeling stage. The corrosion dynamics are simulated using Physics-Informed Neural Networks, which combine the governing physical formulation with the available data to constrain the solution space and improve robustness under limited or heterogeneous observations. By coupling the PINN-based corrosion model with the acquired geometry and externally sourced boundary conditions, the proposed approach yields simulations that are consistent with observed dete-

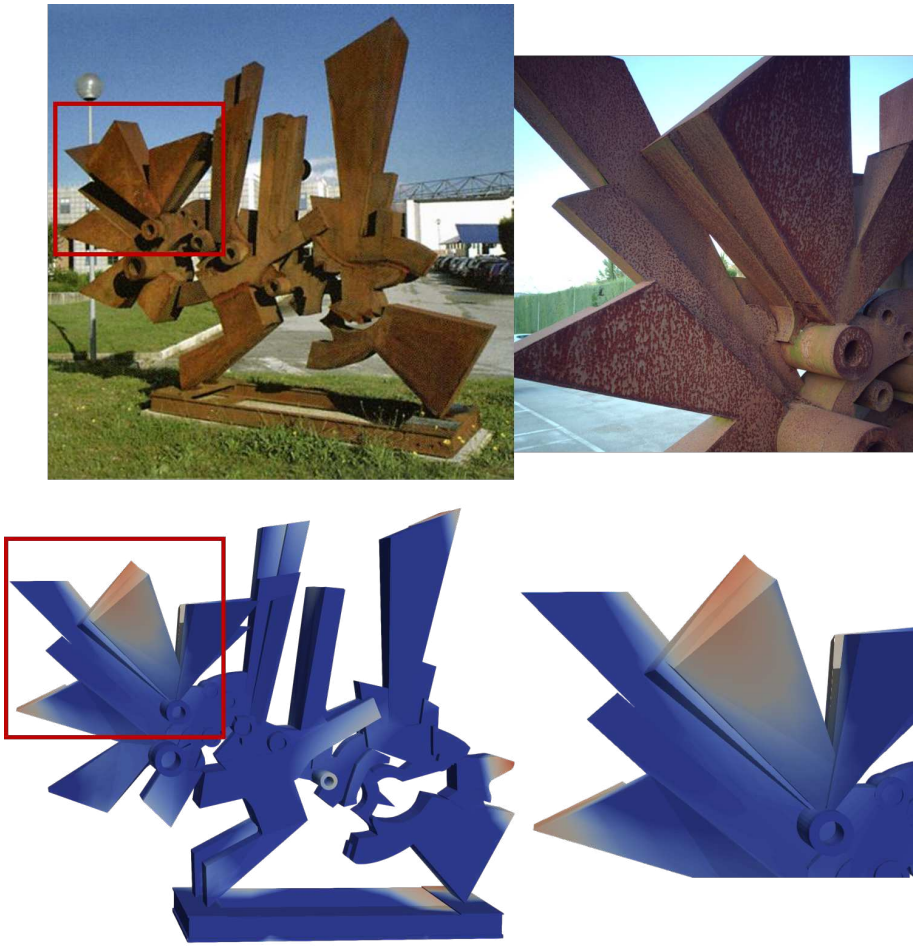


Figure 4.27. Comparison between simulation and real-scenario - Case 2.

rioration patterns. Although the simulation spans only one year, starting from an intact condition, the correspondence with the areas that later exhibit visible corrosion suggests that the model captures meaningful drivers of the process and provides a reliable basis for predictive assessment.

Technical Implications and Limitations of PINN-Based Predictions

The Furia Selvaggia case study demonstrates how corrosion-related data reconstructed using PINNs can inform predictive maintenance plans for cultural heritage. The network's smooth maps help identify areas where damage changes rapidly or are at higher risk. These mapped results can be displayed on a 3D digital model of the asset, enabling targeted monitoring and repair of specific

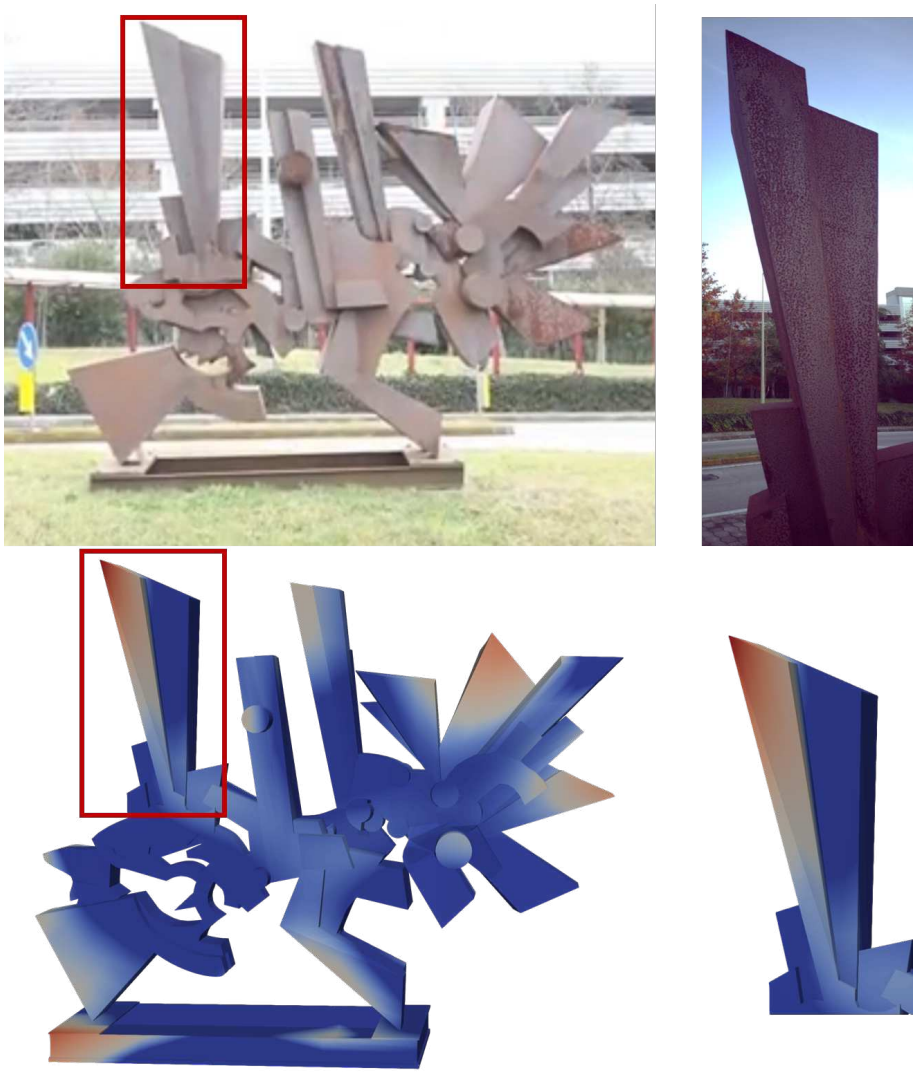


Figure 4.28. Comparison between simulation and real-scenario - Case 3.

features rather than the entire structure.
In the digital twin setup, the PINN serves as a physics-based tool that combines limited sensor data with physical equations to infer missing information and predict how systems will evolve over time. This lets the system assess what might happen if conditions at the boundary or in the environment change and how these changes could affect long-term wear. The same approach can be applied to other types of assets by replacing the physical rules and specifying new boundary details, keeping the process the same while adjusting the model.

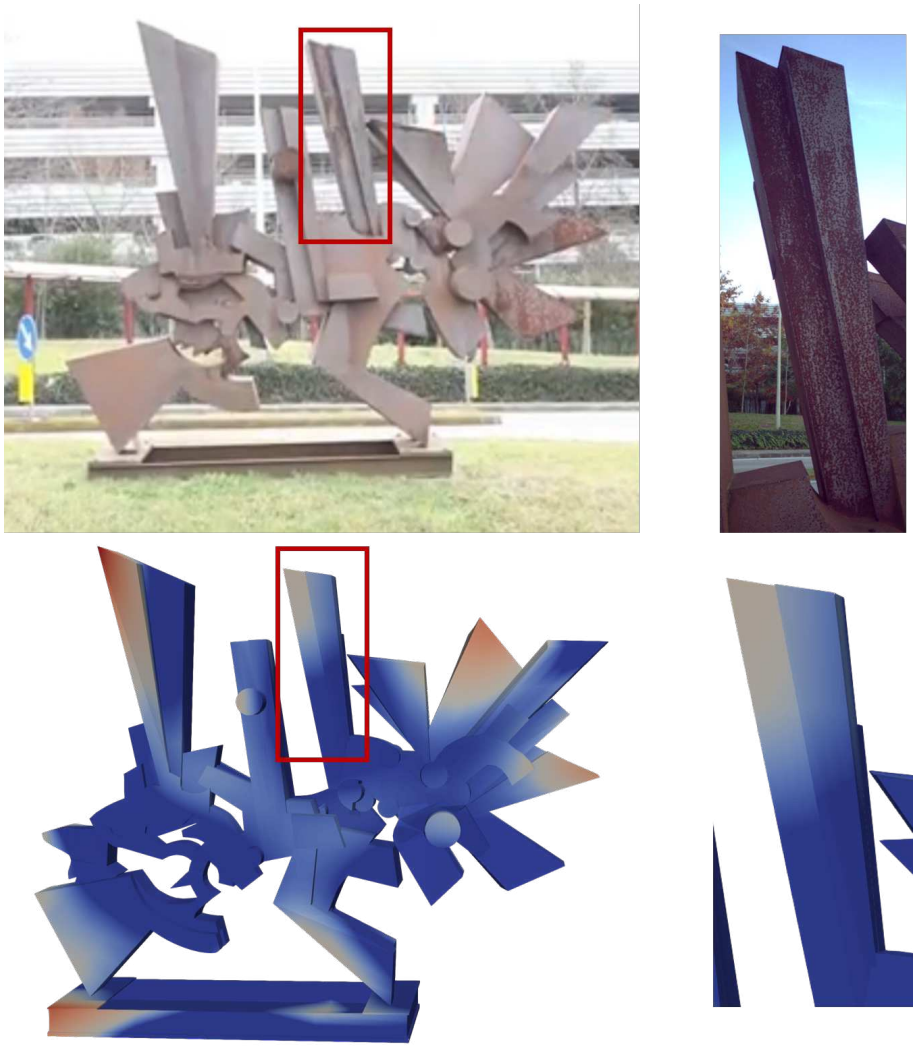


Figure 4.29. Comparison between simulation and real-scenario - Case 4.

However, the reliability of the predictions is strongly conditioned by the fidelity of the embedded physical model and the identifiability of its parameters. Inaccurate or oversimplified governing equations, poorly constrained boundary conditions, or limited excitation in the available data may lead to solutions that satisfy the imposed residual constraints while deviating from the true system dynamics. In addition, PINN performance remains sensitive to loss-term weighting, network architecture, and optimisation dynamics, which can lead to convergence to physically plausible but quantitatively biased solutions, particularly in long-horizon extrapolation or out-of-distribution environments.

Chapter 5

Step-by-Step Time Discrete PINNs: a novel approach to face the causality issue

As shown in the previous Chapter, Physics-Informed Neural Networks (PINNs) have emerged as a powerful tool for solving differential equations by embedding physical laws directly into the training process [135]. Despite their success across a wide range of applications, several intrinsic limitations have been identified in the literature, particularly in time-dependent problems. Among these, two critical issues stand out: the unbalancing of the loss term and the lack of explicit enforcement of causality. The former arises from the heterogeneous nature of the loss components, which typically combine residuals from the governing equations with data and initial and boundary conditions, often leading to stiff optimization dynamics and uneven convergence [107, 173, 174]. The latter concerns the difficulty of ensuring that the learned solution respects the forward propagation of information in time, a fundamental property of evolutionary differential equations.

In standard continuous-in-time PINNs, the solution is approximated over the entire space–time domain simultaneously, and the optimization process does not explicitly encode the temporal directionality inherent to initial-value problems. As a consequence, errors occurring at later time instants may influence the training in a non-physical manner, weakening the causal dependency between past and future states.

Motivated by these considerations, this Chapter focuses on the causality issue in physics-informed models for time-dependent partial differential equations and introduces a methodology to strengthen the temporal structure of the learned solution. The proposed approach integrates classical numerical time-integration schemes into the PINN framework, yielding a continuous-in-space, discrete-in-time formulation. By embedding step-by-step strategies, such as the Implicit

Euler and Crank–Nicolson schemes the network is guided to advance the solution sequentially in time, in accordance with the underlying physical evolution [168].
This hybrid strategy enforces causality by construction. Each time step depends explicitly on the solution from the previous one. Computationally, the method aims to reduce overall cost compared to traditional solvers, which usually require solving a non-linear system at every step. The neural network implicitly approximates time advancement, preserving flexibility in time discretization while limiting computational overhead.
To assess the effectiveness of the proposed methodology, this Chapter exploits a non-linear test problem chosen to show how the proposed approach improve the causality issue. The Chapter focuses on a non-linear diffusion–reaction model from dye-sensitized solar cells, with non-linear reaction terms and time-dependent dynamics. The non-linearity make the problem a good benchmark for testing the step-by-step PINN's ability to capture complex temporal behaviour without loosing the advantages of PINNs that take advantages of automatic differentiation. Numerical experiments compare the proposed approach with time-discrete and continuous-in-time PINN formulations, assessing accuracy, stability, and computational efficiency.

5.1 TIME DISCRETE PINNS

Deep neural networks constitute a recent approach for solving models of PDEs. They have been massively used in recent years to produce a continuous approximation of the exact solution of differential equations, appropriately defining the loss function by taking into account the physics of the problem. Denoting by s the vector containing all the parameters of the network (i.e. weights and biases), the loss is therefore defined as follows:

$$\text{LOSS}\,(s) = \text{LOSS}_{\text{Res}}\,(s) + \text{LOSS}_{\text{IC}}\,(s) + \text{LOSS}_{\text{BCs}}\,(s)\,. \qquad (5.1)$$

Here, $\text{LOSS}_{\text{Res}}(s)$, $\text{LOSS}_{\text{IC}}\,(s)$, $\text{LOSS}_{\text{BCs}}\,(s)$ denote, respectively, the loss functions associated to the residual, initial and boundary conditions. For simplicity of notation, from now in the Chapter the dependence of the loss function on network parameters is omitted. Continuous PINNs take as input a time instant t and a point x in the space of the problem definition domain, and return as output $f(x,\, t)$, which constitutes the approximation of the continuous solution of the PDE to solve. Clearly, this approach involves defining a loss function that takes into account the equations that define the problem; LOSS_{Res} in Eq. (5.1) represents precisely the residual of the PDE. As already mentioned, the loss also takes into account the initial and boundary conditions.
In this Chapter, the focus consists of proposing a new approach that allows the use of neural networks to compute a continuous in space and discrete in time

approximation of the solution of a PDE. To this aim, the proposed approach appropriately *incorporate* known numerical methods for the solution of IVPs within deep neural networks. The idea of using numerical schemes for IVPs together with neural networks to solve PDEs was investigated by Raissi et al. in 2019 [135, Subsec. 3.2]. The current subsection is dedicated to the description of such approach, which is based on RK methods. Then, the next section introduces a new approach based on merging the one-stage implicit Euler and Crank-Nicolson methods into NNs. Specifically, these methods are simple to handle, and allow to easily set up the network in order to get the solution at each point of a discrete time grid. As it will see, this is not possible with existing time discrete PINNs.
Let us consider a time evolutionary PDE, equipped with known initial and boundary conditions, expressed in the following way:

$$\frac{\partial u}{\partial t} + \mathcal{N}(u(x,\ t)) = 0, \qquad t \in [t_0,\ t_e] \subset \mathbb{R}. \tag{5.2}$$

Here, x is a vector denoting the space variables, $\mathcal{N}(u(x,\ t)) = F(D^k u,\ D^{k-1}u,\ \ldots,\ Du,\ u,\ x,\ t)$, with F functional related to the problem. In F, D represents a differentiation operator with respect to x, and k is an integer indicating the order of derivation. For simplicity, a one-dimensional space is considered; therefore, $x \in [x_0,\ x_e] \subset \mathbb{R}$. Let us define the time uniform grid $\{t_n = t_0 + n\,\delta t;\ n = 0,\ \ldots,\ n_{\text{end}};\ t_{n_{\text{end}}} = t_e\}$, where δt denotes the step-size related to the application of the numerical method. RK methods to solve problem (5.2) can be expressed as follows:

$$\begin{aligned} u_{n-1,i}(x) &= u_{n-1}(x) - \delta t \sum_{j=1}^{q} a_{ij}\mathcal{N}(u_{n-1,j}(x)), \\ u_n(x) &= u_{n-1}(x) - \delta t \sum_{j=1}^{q} b_j\mathcal{N}(u_{n-1,j}(x)), \end{aligned} \tag{5.3}$$

$$i = 1,\ \ldots,\ q, \quad n = 1,\ \ldots,\ n_{\text{end}}.$$

Let denote $u_n(x) \approx u(x,\ t_n)$, $n = 0,\ \ldots,\ n_{\text{end}}$; observe that $u_0(x)$ is known, since it corresponds to the initial condition of the problem. Furthermore, q indicates the number of stages $u_{n,i}(x) \approx u(x, t_n + c_i\ \delta t)$, with c_i nodes belonging to the interval $[0,\ 1]$. Recall that the coefficients of RK methods can be expressed through the matrix $A = (a_{ij})_{i,j=1}^{q} \in \mathbb{R}^{q\times q}$ and the vectors $b = (b_i)_{i=1}^{q} \in \mathbb{R}^q$, $c = (c_i)_{i=1}^{q} \in \mathbb{R}^q$. They constitute the so-called Butcher tableau of the RK method.
Let us fix the current time step t_n. Thus, assuming that $u_{n-1}(x)$ is known from

the previous time-step, the RK method in Eq. (5.3) can be written as

$$\begin{aligned} u_{n-1}(x) &= u^i_{n-1}(x), \\ u_{n-1}(x) &= u^{q+1}_{n-1}(x), \end{aligned} \qquad i = 1,\ \ldots,\ q,$$

with

$$\begin{aligned} u^i_{n-1}(x) &= u_{n-1,i}(x) + \delta t \sum_{j=1}^{q} a_{ij}\mathcal{N}(u_{n-1,j}(x)), \\ u^{q+1}_{n-1}(x) &= u_n(x) + \delta t \sum_{j=1}^{q} b_j\mathcal{N}(u_{n-1,j}(x)), \end{aligned} \qquad i = 1,\ \ldots,\ q. \quad (5.4)$$

Raissi et al. consider a feed-forward deep network with one neuron in the input layer and $q+1$ neurons in the output layer. Their goal is to define the network in such a way that, taking as input a point $x \in (x_0,\ x_e)$, it returns as outputs approximations of the RK stages and advancing solution, see also [135, Eq. (10)], i.e. the vector:

$$(u_{n-1,1}(x),\ \ldots,\ u_{n-1,q}(x),\ u_n(x)).$$

To this aim, the authors choose the loss

$$\text{LOSS}^n_{\text{RK}} = \frac{1}{M}\sum_{j=1}^{q+1}\sum_{i=1}^{M}(u^j_{n-1}(x_i) - u_{n-1}(x_i))^2, \quad (5.5)$$

where $u^j_{n-1}(x_i)$ and $u^{q+1}_{n-1}(x_i)$ are derivable from Eq. (5.4), and the quantities $u_{n-1}(x_i),\ \forall i$, are known from the previous time-step. Furthermore, $x_i,\ i = 1,\ \ldots,\ M$ denote the training points, belonging to spatial interval $(x_0,\ x_e)$, to be chosen at the time step t_n (for simplicity the dependence of these points is omitted, and of M, on the index n).

Let us represent the neural network described in Figure 5.1; let denote the activation functions associated to each layer by $g^1,\ \ldots,\ g^N$, being N integer (in the figure, $N = 4$), and the output by the vector $f(x) := (f_1(x),\ \ldots,\ f_L(x)) \in \mathbb{R}^L$. The sizes of the domain and codomain of $g^\ell,\ \ell = 1,\ \ldots,\ N$, correspond to the number of neurons in the layer ℓ and $\ell+1$, respectively. Clearly, $f(x) = (g^N \circ g^{N-1} \circ \cdots \circ g^2 \circ g^1)(x)$. For the time discrete model by Raissi et al. presented in this subsection, $L = q+1$, and the network outputs are constructed in order to satisfy $f_1(x) \approx u_{n-1,1}(x),\ \ldots,\ f_{L-1}(x) \approx u_{n-1,q}(x),\ f_L(x) \approx u_n(x)$.

Note that in defining the loss function as in Eq. (5.5), it is crucial that the value of $u_{n-1}(x)$ is known. Finally, let underline that when a neural network is used to solve a PDE, it is necessary to define the loss function so that it takes into

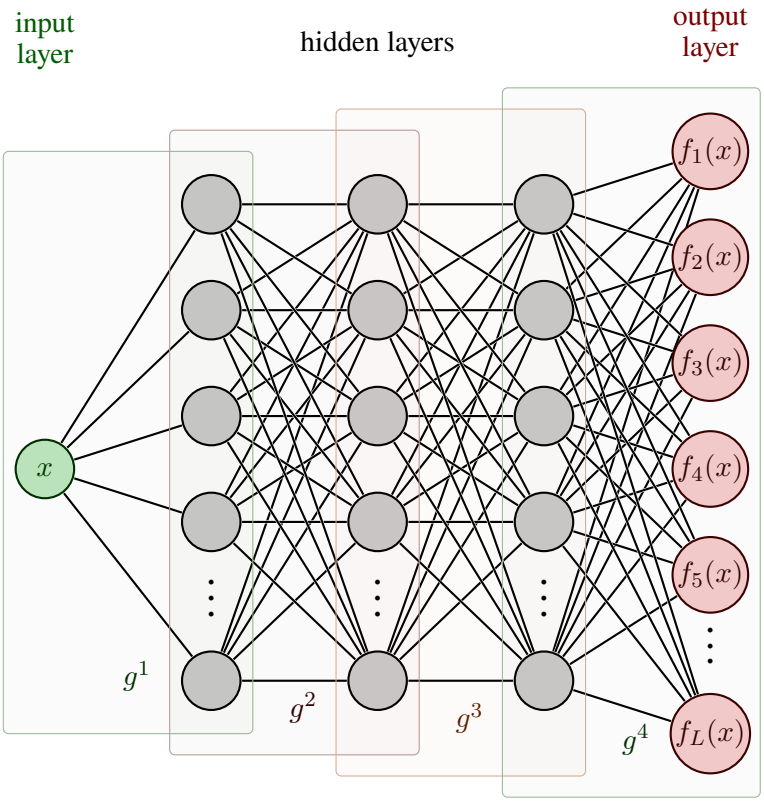


Figure 5.1. Representation of the feed-forward deep neural network described in Subsection 5.1 and in Section 5.2.

account the physics of the problem, e.g. the boundary conditions. Therefore, the loss in Eq. (5.5) must be slightly modified, by taking into account the problem to solve. An example of this in Section 5.4 will be shown, where the discrete neural networks described here is employed, compared with the new ones proposed in this Chapter, to solve a sustainability model.

Remark 1. *To use the approach described in this section (i.e. the time discrete models proposed in the Chapter [135]), it is necessary to consider a different neural network for each step of the RK method. That is, every time a step of the RK method is taken, it is necessary to carry out a new training to redefine the network parameters. To avoid this, Raissi et al. propose to use a RK method with a decidedly high number of stages (e.g.* $q = 100$ *or* $q = 500$*), carrying out a single time integration step, see [135, Par. 3.2.1 and Subsec. A.2]. In particular, they compute a reference solution at a time grid point* $t^* > t_0$ *(with* t^* *close to* t_0*), then applying the method* (5.3) *using step-size* $\delta t = t_e - t^*$*. Clearly, this generally involves the employment of a non-negligible*

discretization step, corresponding to the size of almost the entire time interval. Furthermore, note that using a RK method with a high number of stages is not trivial; in fact, it is necessary to compute its Butcher tableau, which in this case would have large dimensions. The next section defines new types of time discrete PINNs, which aim to overcome the drawbacks listed above.

5.2 NEW STEP-BY-STEP TIME DISCRETE PHYSICS-INFORMED NEURAL NETWORKS

This section is dedicated to the introduction of new time discrete neural networks for the solution of PDEs. In particular, the next subsection describes the new approach, and the last one highlights connections and differences with the neural networks proposed by Raissi et al. in 2019 [135] (which have been just described).

5.2.1 *TIME DISCRETE PINNS BASED ON IMPLICIT EULER AND CRANK-NICOLSON METHODS*

The new proposed neural networks are based on the one-step single-stage implicit Euler and Crank-Nicolson methods. Let express them below, respectively:

$$\begin{aligned} &u_n(x) = u_{n-1}(x) - \delta t\, \mathcal{N}(u_n(x)), \\ &u_n(x) = u_{n-1}(x) - \frac{\delta t}{2}\left(\mathcal{N}(u_{n-1}(x)) + \mathcal{N}(u_n(x))\right), \end{aligned} \tag{5.6}$$

$$n = 1,\ \ldots,\ n_{\text{end}}.$$

The notation used here corresponds to that introduced in Subsection 5.1. The goal lies in appropriately defining a neural network that solves problem (5.2) based on the methods (5.6). In particular, let consider the neural network in Figure 5.1, with $L = n_{\text{end}}$; the loss function is defined in such a way that the outputs satisfy $f_n(x) \approx u_n(x),\ n = 1,\ \ldots,\ n_{\text{end}},\ \forall x \in (x_0,\ x_e)$.
In this regard, let us prove the following result.

Proposition 5.1. *The numerical solutions at the time grid point t_n given by the implicit Euler and Crank-Nicolson methods can be expressed as follows,*

respectively:

$$u_n(x) = u_0(x) - \delta t \sum_{i=1}^{n} \mathcal{N}(u_i(x)), \qquad n \geq 1, \tag{5.7}$$

$$u_n(x) = u_0(x) - \frac{\delta t}{2}\left(\mathcal{N}(u_0(x)) + 2\sum_{i=1}^{n-1} \mathcal{N}(u_i(x)) + \mathcal{N}(u_n(x))\right), \quad n \geq 1. \tag{5.8}$$

Proof. The proof of this result is done by induction, both for implicit Euler and Crank-Nicolson.
Regarding the implicit Euler method, the induction basis is certainly verified, given that

$$u_1(x) = u_0(x) - \delta t\, \mathcal{N}(u_1(x)).$$

Therefore, let as assume that Eq. (5.7) holds true up to the index $n-1$, i.e.

$$u_{n-1}(x) = u_0(x) - \delta t \sum_{i=1}^{n-1} \mathcal{N}(u_i(x)).$$

Since by implicit Euler $u_n = u_{n-1}(x) - \delta t\, \mathcal{N}(u_n(x))$, from the above formula, the result follows.
Regarding the Crank-Nicolson method, the induction basis is verified since

$$u_1(x) = u_0(x) - \frac{\delta t}{2}\left(\mathcal{N}(u_0(x)) + \mathcal{N}(u_1(x))\right).$$

Therefore, assuming that

$$u_{n-1}(x) = u_0(x) - \frac{\delta t}{2}\left(\mathcal{N}(u_0(x)) + 2\sum_{i=1}^{n-2} \mathcal{N}(u_i(x)) + \mathcal{N}(u_{n-1}(x))\right),$$

the result follows by substitution, given $u_n(x) = u_{n-1}(x) - \frac{\delta t}{2}\left(\mathcal{N}(u_{n-1}(x)) + \mathcal{N}(u_n(x))\right)$. □

Note that Proposition 5.1 allows each step of the implicit Euler and Crank-Nicolson methods to be expressed as a function of the initial condition u_0. Although very simple, Proposition 5.1 and the related proof are very important, both to understand how to define the loss function of the neural network, and to highlight the connections and differences between the methods proposed in this paper and those proposed by Raissi et al. in 2019 [135].

Let first consider the implicit Euler method. Using the assumption $f_n(x) \approx u_n(x)$, and deriving $u_0(x)$ from Eq. (5.7), is derived

$$u_0^n(x) = f_n(x) + \delta t \sum_{i=1}^{n} \mathcal{N}(f_i(x)), \text{ with } u_0^n(x) \approx u_0(x), \quad n = 1, \ldots, n_{\text{end}}.$$

Remembering that $u_0(x)$ is known from the problem (since it represents the initial condition), by introducing the residuals

$$\text{Res}_n^{\text{Euler}}(x) = u_0^n(x) - u_0(x) = f_n(x) + \delta t \sum_{i=1}^{n} \mathcal{N}(f_i(x)) - u_0(x), \quad (5.9)$$

$$n = 1, \ldots, n_{\text{end}},$$

a convenient way to define the loss function of the neural network with 'embedded' implicit Euler can therefore be

$$\text{LOSS}_{\text{Euler}} = \frac{1}{M} \sum_{n=1}^{n_{\text{end}}} \sum_{i=1}^{M} (\text{Res}_n^{\text{Euler}}(x_i))^2. \quad (5.10)$$

According to the notation introduced in the previous section, x_i, $i = 1, \ldots, M$ denote the spatial training points chosen in the spatial interval (x_0, x_e). Clearly, the lower the residuals, the closer the solution given by the neural network is to that obtained through the classical implicit Euler method at each point of the time grid (at the selected training points of the spatial grid).
Analogously for the Crank-Nicolson method, using Eq. (5.8), the following relation is obtained

$$\text{Res}_n^{\text{CN}}(x) = f_n(x) + \frac{\delta t}{2}\big(\mathcal{N}(u_0(x)) + 2\sum_{i=1}^{n-1} \mathcal{N}(u_i(x)) + \mathcal{N}(u_n(x))\big) - u_0(x), \quad (5.11)$$

$$n = 1, \ldots, n_{\text{end}},$$

and consequently the loss function

$$\text{LOSS}_{\text{CN}} = \frac{1}{M} \sum_{n=1}^{n_{\text{end}}} \sum_{i=1}^{M} (\text{Res}_n^{\text{CN}}(x_i))^2. \quad (5.12)$$

Section 5.4 will show which terms to add to these loss functions in order to also incorporate into the neural network the information given by the boundary conditions of the PDE to solve. Note that, with the proposed approach, the initial condition is automatically incorporated into the loss function. This does not happen for continuous PINNs, for which the initial condition must be integrated together with the boundary conditions in the loss function.

The proposal of this subsection allows the implicit Euler and Crank-Nicolson methods to be merged into neural networks. The following remark explains why this can be advantageous, compared to using these numerical methods in the standard way.

Remark 2. *The classical implicit Euler and Crank-Nicolson methods require the solution of n_{end} nonlinear systems, of size depending on that of the problem. Merging these methods in the neural network, as shown in this subsection, avoids the solution of such systems. Therefore, in the proposed setting it is more advantageous to use implicit methods than explicit ones, since they have good stability properties and are able to solve stiff problems by using discretization steps of reasonable size. It is underlined that to change the choice of time step δt, it is sufficient to set the last layer of the network so that it is composed of $n_{end} = \frac{t_e - t_0}{\delta t}$ neurons. Finally, observe that once the network is trained, it is possible to make predictions of the solution at any point of the spatial domain where the PDE is defined.*

Remark 3. *Note that, due to the way the PINN is constructed with the proposed approach, the neurons of the output layer return approximations of the numerical solutions given by the classical implicit Euler and Crank-Nicolson methods at each time step t_n, $n \geq 1$; let us denote these solutions by $u_n(x)$, $n \geq 1$. That is, if the training phase is done correctly and the neural network 'is able to well minimize' the terms of the residuals in Eqs.* (5.10)-(5.12)*, then the outputs $f_n(x)$, $n \geq 1$, see Figure 5.1, are good approximations of $u_n(x)$, $n \geq 1$, i.e. the implicit Euler and Crank-Nicolson solutions, respectively. For such numerical methods, convergence (they have time order 1 and 2, respectively) and good stability (A-stability) are known properties. Therefore, due to the way the new PINNs are built, the convergence and stability properties of numerical methods are inherited by the neural network. In fact, this is exactly what numerical tests show (see next Figure 5.4).*
Clearly, for the new PINNs to inherit these properties, the method used for minimizing the loss function must converge. In this case, the Adam method is applied. This method is widely used to optimize the neural network hyperparameters, as it is very effective in practice. Theoretical results on the convergence of Adam method can be found e.g. in [20, 42]. To conclude, due to the way in which the new PINNs are built, assuming the convergence of Adam method and that the training phase is carried out successfully, it can be expected that the consistency and stability properties of the numerical methods which are merged in the neural network are preserved.

The following subsection highlights the links and differences between the time discrete PINNs proposed by us and the ones by Raissi et al., explaining the advantages of the former.

5.2.2 CONNECTIONS WITH EXISTING TIME DISCRETE MODELS

To highlight the connections and differences between the proposed approach and the one by Raissi et al. [135] presented in the previous section, the following two results are proved, one linked to the implicit Euler method and the other to the Crank-Nicolson method.

Proposition 5.2. *Let us consider the time interval $[t_0,\ t_n]$. Let us perform one step of a RK method with n stages. Furthermore, let us apply the implicit Euler method, considering the uniform grid $\{t_i = t_0 + i\ \delta t,\ i = 1,\ \dots, n\}$, $n \geq 1$. The steps of implicit Euler correspond, respectively, to the stages of the RK method when the arrays of the Butcher tableau are as follows:*

$$A = \frac{1}{n}\begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & \dots & \dots & 1 \end{pmatrix} \in \mathbb{R}^{n\times n}, \tag{5.13}$$

$$c = \frac{1}{n}\,(1,\ 2,\ \dots,\ n) \in \mathbb{R}^n.$$

Also, taking b equal to the last row of A, the time-marching solution (at t_n) provided by the RK method corresponds to the last step of implicit Euler.

Proof. To carry out the proof, let start from a generic RK with n stages, demonstrating that the result holds true when the arrays of its Butcher tableau are as in Eq. (5.13). Let us consider the interval $[t_0,\ t_n]$, defining $\Delta t = t_n - t_0$. With the same notation used so far, the n stages of a RK method performing one step read

$$u_{0,i}(x) = u_0(x) - \Delta t \sum_{j=1}^{n} a_{ij}\mathcal{N}(u_{0,j}(x)), \quad i = 1,\ \dots,\ n, \tag{5.14}$$

$u_0(x) \approx u(x,\ t_0)$, $u_{0,i}(x) \approx u(x,\ t_0 + c_i\ \Delta t)$. Instead, the n steps of implicit Euler can be written as

$$u_i(x) = \ u_0(x) - \delta t \sum_{j=1}^{i} \mathcal{N}(u_j(x)), \quad i = 1,\ \dots,\ n, \tag{5.15}$$

with $u_i(x) \approx u(x,\ t_0 + i\ \delta t)$, $i = 1,\ \dots,\ n$, being $\delta t = \frac{\Delta t}{n}$, see Eq. (5.7). Imposing that the RK stages in Eq. (5.14) correspond to the Euler steps in Eq. (5.15) means $t_0 + c_i\ \Delta t = t_0 + i\ \delta t$,

$$\Delta t \sum_{j=1}^{n} a_{ij}\mathcal{N}(u_{0,j}(x)) = \delta t \sum_{j=1}^{i} \mathcal{N}(u_j(x)) \quad (u_{0,i}(x) = u_i(x)), \quad i = 1,\ \dots,\ n.$$

Using that $\Delta t = n\ \delta t$, this leads to

$$a_{ij} = \begin{cases} \frac{1}{n}, & \text{if } j \leq i, \\ 0, & \text{if } j > i, \end{cases} \qquad c_i = \frac{i}{n}, \qquad \text{for } i = 1,\ \ldots,\ n.$$

By defining b as the last row of A, the advancing solution of the RK method corresponds to the stage $u_{0,n}(x)$, and therefore, for the above computations, to the solution of implicit Euler at t_n. Since this proof holds for every value of n, the result follows. □

Proposition 5.3. *Let us consider the time interval $[t_0,\ t_n]$. Let us perform one step of a RK method with $n+1$ stages. Furthermore, let us apply the Crank-Nicolson method, considering the uniform grid $\{t_i = t_0 + i\ \delta t,\ i = 1,\ \ldots, n\}$, $n \geq 1$. The steps of Crank-Nicolson correspond, respectively, to the last n stages of the RK method when the arrays of the Butcher tableau are as follows:*

$$A = \frac{1}{n} \begin{pmatrix} 0 & 0 & & \ldots & & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \ldots & \ldots & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 & \ldots & 0 \\ \vdots & & & \ddots & & \vdots \\ \frac{1}{2} & 1 & \ldots & 1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & \ldots & \ldots & 1 & \frac{1}{2} \end{pmatrix} \in \mathbb{R}^{n+1 \times n+1}, \tag{5.16}$$

$$c = \frac{1}{n}\ (0,\ 1,\ \ldots,\ n) \in \mathbb{R}^{n+1}.$$

The first stage corresponds to the initial condition $u_0(x)$. Also, taking b equal to the last row of A, the time-marching solution (at t_n) provided by the RK method corresponds to the last step of Crank-Nicolson.

Proof. The proof is carried out with the same strategy used for the previous result. Therefore, the interval $[t_0,\ t_n]$, $\Delta t = t_n - t_0$ is considered and a RK method with $n+1$ stages

$$u_{0,i}(x) = u_0(x) - \Delta t \sum_{j=1}^{n+1} a_{ij} \mathcal{N}(u_{0,j}(x)), \quad i = 1,\ \ldots,\ n+1,$$

$u_0(x) \approx u(x,\ t_0)$, $u_{0,i}(x) \approx u(x,\ t_0 + c_i\ \Delta t)$. Let us fix $a_{1j} = 0,\ j = 1,\ \ldots,\ n+1,\ c_1 = 0$, i.e. the first stage corresponds to the initial condition. The n steps of Crank-Nicolson can be written as

$$u_i(x) = u_0(x) - \frac{\delta t}{2}\ \big(\mathcal{N}(u_0(x)) + 2 \sum_{j=1}^{i-1} \mathcal{N}(u_j(x)) + \mathcal{N}(u_i(x))\big),$$

$$i = 1, \ \ldots, \ n,$$

with $u_i(x) \approx u(x, \ t_0 + i \ \delta t), \ i = 1, \ \ldots, \ n$, being $\delta t = \frac{\Delta t}{n}$, see Eq. (5.8). Imposing that the last n RK stages correspond to the Crank-Nicolson steps means $t_0 + c_{i+1} \ \Delta t = t_0 + i \ \delta t$,

$$\Delta t \sum_{j=1}^{n+1} a_{i+1j} \mathcal{N}(u_{0,j}(x)) = \frac{\delta t}{2} \left(\mathcal{N}(u_0(x)) + 2 \sum_{j=1}^{i-1} \mathcal{N}(u_j(x)) + \mathcal{N}(u_i(x))\right) \quad (u_{0,i+1}(x) = u_i(x)),$$

$i = 1, \ \ldots, \ n$. Using that $\Delta t = n \ \delta t$, this leads to

$$a_{i+1j} = \begin{cases} \frac{1}{2n}, & \text{if } j = 1, \ i, \\ \frac{1}{n}, & \text{if } 1 < j < i, \\ 0, & \text{if } j > i, \end{cases} \qquad c_{i+1} = \frac{i}{n}, \qquad \text{for } i = 1, \ \ldots, \ n.$$

Defining b equal to the last row of A, as for the previous result, the proof follows. □

Let us observe the loss functions defined for the RK, Euler and Crank-Nicolson methods, in Eq. (5.5), Eq. (5.10), Eq. (5.12), respectively. Note that by defining a RK method as in Proposition 5.2 or Proposition 5.3, performing a single time step in $[t_0, \ t_e]$, the approach of Raissi et al. leads to the time discrete PINNs proposed in this paper. Clearly, this observation serves to highlight the link between the discrete models derived here and those of the paper [135], but obviously there are no RK methods of such type (i.e. whose Butcher tableau is defined as in Proposition 5.2 or Proposition 5.3). Indeed, the approach of Raissi et al. is not intended to merge multiple steps of a single-stage method into a neural network, as proposed by us in this paper. As Section 5.4 will show, the connection between the classical discrete time models, and the new step-by-step models, established in this subsection, will be useful also from an implementation point of view.
To explain the possible advantages of the proposed time dicrete PINNs, let us consider the following remark.

Remark 4. *With the new PINNs proposed in this paper, the drawbacks described in Remark 1 of the time discrete models of the manuscript [135] can be overcome. Indeed, with the new methodology presented in this section, by training a single neural network it is possible to determine the solution of the method at each time grid point. Therefore, it is not necessary to retrain the network for each step. It is also emphasized that, minimizing the loss function exploiting for the learning of the approximated solution the initial condition $u_0(x)$, which is known from the problem, is better than doing so with respect to the solution in the previous step $u_{n-1}(x)$.*

Finally, it can be observed that the proposed approach can also be convenient compared to making a single step of a RK method, as proposed by Raissi et al. to avoid the above issues. In fact, for the latter case it is necessary to use a large time discretization step-size, and generally to have a RK method with a high number of stages; changing the number of stages also means recomputing the coefficients of the RK method. The computation of such coefficients, especially for a very high number of stages q, is not a trivial task; Raissi et al., on github[1]*, have reported the RK coefficients only for some values of q, with $q \leq 500$. Instead, the proposed approach is more flexible, as we can choose the step-length we want without changing the structure of the method. To see this clearly, look for example at the structure of the proposed methods, revisited as RK in Proposition 5.2 and Proposition 5.3; note that the structure of the tableau remains the same for each value of n, and therefore of the time step δt. It is also highlight that having the freedom to use a small time step-size, compared to being forced to use one as large as the time grid, can be advantageous for accuracy reasons, i.e. to get small errors.*

5.3 The testing model: a non-linear diffusion-reaction model for dye-sensitized solar cells

DSSCs were introduced in a famous paper by O'Regan and Grätzel in 1991 [122], with the aim of representing an approach to tackling the issue related to the production of renewable energy. Indeed, they allow the conversion of sunlight into electricity at relatively low production costs. DSSCs and traditional junction solar cells differ in several aspects, see e.g. [101]: from the cost of the materials, which is lower for the former, to the overall structure and the approach to their modeling. There are vast areas of research on them, concerning e.g. determining new materials, optimal structures and device configurations to maximize the performance and efficiency of DSSCs. The mathematical community is also becoming interested in DSSCs, both from a modeling and numerical point of view (i.e. to determine analytical and numerical methods capable of providing accurate solutions of the models), see e.g. [9, 15, 91, 101, 183] and references therein contained.
Below, it is briefly described the main steps related to the functioning of DSSCs, subsequently introducing the PDE model of interest in this manuscript. DSSCs are composed of four main components, see e.g. [101, Sec. 1]: a nano-porous semiconductor, a light-sensitive dye, an electrolyte couple and a counter electrode. The functioning of the device can be summarized as follows: sunlight excites the dye molecules bringing them to a higher energy state than the current one; exploiting this energy, excited molecules then carry electrons to the nano-

[1] https://github.com/maziarraissi/PINNs/tree/master

porous semiconductor, leaving the DSSC in order to power the load (this step is known as electrons injection); the injected electrons are reintroduced into the DSSC through a counter electrode; finally, the electrolytic couple returns the electrons from the counter electrode to the photosensitive dye by means of a redox reaction. DSSCs are also characterized by loss mechanisms, called recombination steps, such as regeneration of the photosensitive dye or electrolyte pair by the nano-porous semiconductor. The recombination steps damage the process of energy generation, since they prevent electrons from leaving the DSSC to power a load, as expected. Therefore, optimizing the performance of a DSSC means enhancing the reactions that allow energy to be generated, while at the same time inhibiting the recombination processes.
The DSSCs model of interest in this manuscript is given by the following non-linear diffusion-reaction PDE, see e.g. [11, 102]:

$$\frac{\partial u}{\partial t} - D_0 \frac{\partial}{\partial x}\left((\frac{u}{u_{\text{eq}}})^{\beta} \frac{\partial u}{\partial x} \right) - \varphi \alpha e^{-\alpha x} + k_R (\frac{u}{u_{\text{eq}}})^{\beta} (u - u_{\text{eq}}) = 0, \quad (5.17)$$

$$(x, t) \in [0,\ d] \times [0,\ t_e].$$

Here, $u(x, t)$ is the electron density of the conduction band at depth x and time t. The point $x = 0$ denotes the position of the conductive oxide, and the point $x = d$ denotes the location of the counter electrode. The PDE (5.17) is also known as electron diffusion equation. The coefficients $D_0,\ \beta,\ u_{\text{eq}},\ \varphi,\ \alpha,\ k_R$ denote, respectively: the diffusion, the diffusion order, the dark equilibrium electron density, the incident photon flux, the absorption of the dye, the recombination rate. The initial and boundary conditions are set in the following way (see e.g. [102]):

$$u(x,\ 0) = u_{\text{eq}}^{\frac{qV}{m_I k_B T}}, \quad u(0,\ t) = u_{\text{eq}}^{\frac{qV}{m_I k_B T}}, \quad \left.\frac{\partial u}{\partial x}\right|_{x=d} = 0.$$

Here, $q,\ V,\ m_I,\ k_B,\ T$ denote, respectively: the electron charge, the applied bias voltage, the diode ideality factor, the constant of Boltzmann, the temperature of the DSSC. That is, it has been used the Boltzmann approximation for the density of the conductive oxide substrate, and it has been hypothesized that the current density vanishes at the counter electrode. Through parameters scaling, see [32, 102], it is possible to move from model (5.17) to the following non-dimensional PDE:

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(u^{\beta} \frac{\partial u}{\partial x} \right) - \mu e^{-\nu x} + \xi u^{\beta} (u - 1) = 0, \quad (5.18)$$

$$(x, t) \in [0,\ d] \times [0,\ t_e].$$

The initial and boundary conditions become

$$u(x,\,0) = e^{\omega}, \quad u(0,\,t) = e^{\omega}, \quad \left.\frac{\partial u}{\partial x}\right|_{x=d} = 0. \tag{5.19}$$

For the numerical solution of the non-linear reaction-diffusion PDE (5.18), it will be considered in the experiments the following parameters values (see [11, 102]): $\beta = 1,\ \mu = 25,\ \nu = 5,\ \xi = 10^{-5},\ \omega = 0$. Furthermore, it will be taken $d = 1,\ t_e = 1$. The next section shows how a reference solution is derived for this model, in order to use it in numerical tests for computing errors.

5.3.1 Computation of a reference solution

To compute a sufficiently accurate reference solution of the model (5.18), the first step consists of discretizing it in space, then using the Matlab function `ode15s`, requiring absolute and relative tolerance equal to `eps`. Below, we describe the spatial semi-discretization used.
Let us consider the uniform grid in space $\{x_m = m\ \delta x;\ m = 0,\ \ldots,\ m_{\text{end}};\ x_{m_{\text{end}}} = d\}$ related to the interval $[0,\ d]$. Denoting by $v(t)$ the column vector $(v^1(t),\ \ldots,\ v^{m_{\text{end}}}(t))^T$, with $v^m(t) \approx u(x_m,\ t)$, for $m = 1,\ \ldots,\ m_{\text{end}}$, by e_{vec} the column vector $(e^{-\nu x_1},\ \ldots,\ e^{-\nu d})^T$, and by '$\cdot$' the Hadamard product, a spatial discretization is employed, leading to the system of ODEs

$$v'(t) - \Big(\mathcal{M}_2\big(v(t)^{\beta}\mathcal{S}(v(t))\big) + \text{vec}_2(t)\Big) - \mu\, e_{\text{vec}} + \xi\, v(t) \cdot (v(t) - 1) = 0, \tag{5.20}$$

where $\mathcal{S}(v(t)) = \mathcal{M}_1 v(t) + \text{vec}_1(t)$,

$$\mathcal{M}_1 = \frac{1}{2\ \delta x}\begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & 0 & 0 \end{pmatrix} \in \mathbb{R}^{m_{\text{end}} \times m_{\text{end}}},$$

$$\text{vec}_1(t) = \begin{pmatrix} -\frac{e^{\omega}}{2\ \delta x} \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^{m_{\text{end}}},$$

$$\mathcal{M}_2 = \frac{1}{2\,\delta x}\begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & 1 & -4 & 3 \end{pmatrix} \in \mathbb{R}^{m_{\text{end}}\times m_{\text{end}}},$$

$$\text{vec}_2(t) = \begin{pmatrix} \frac{e^{\beta\omega}(-3e^{\omega}+4\upsilon^1(t)-\upsilon^2(t))}{4\,\delta x^2} \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^{m_{\text{end}}}.$$

Let us denote by $\mathcal{F}$ the operator

$$\mathcal{F}(\upsilon(t)) := \upsilon'(t) - \Big(\mathcal{M}_2\big(\upsilon(t)^{\beta}\mathcal{S}(\upsilon(t))\big) + \text{vec}_2(t)\Big) - \mu\, e_{\text{vec}} + \xi\, \upsilon(t)\cdot(\upsilon(t)-1). \tag{5.21}$$

Therefore, note that the system of ODEs (5.20) can be equivalently expressed as $\mathcal{F}(\upsilon(t)) = 0$. The local truncation error $\mathcal{E} = (\mathcal{E}_1, \ldots, \mathcal{E}_{m_{\text{end}}})$ that is committed by approximating the PDE (5.18) with the semi-discretized problem (5.20) can thus be defined as

$$\mathcal{E} := \mathcal{F}((u(x_1,t), \ldots, u(x_{m_{\text{end}}},t))^T). \tag{5.22}$$

That is, $\mathcal{E}$ has been obtained by substituting the exact solution of the PDE (5.18) in Eq. (5.21).

Proposition 5.4. *The local error defined in Eq.* (5.22) *satisfies:* $\mathcal{E}_m = O(\delta x^2),\ \forall\, m = 1, \ldots, m_{end}$.

Proof. First, let us define

$$\mathcal{P}(u) := \frac{\partial u}{\partial t} - \big(\mathcal{D}(u) + \mathcal{R}(u)\big), \tag{5.23}$$

with

$$\mathcal{D}(u) := \frac{\partial}{\partial x}\left(u^{\beta}\frac{\partial u}{\partial x}\right),\ \mathcal{R}(u) := \mu e^{-\nu x} - \xi u^{\beta}(u-1).$$

Note that $\mathcal{P}(u(x,\,t)) = 0,\ \forall\, x \in [0,\,d],\ \forall\, t \in [0,\,t_e]$, by the PDE (5.18). Let us denote by $\mathcal{S}_m$ the components of the vector $\mathcal{S}((u(x_1,t), \ldots, u(x_{m_{\text{end}}},t))^T)$. Due to the shape of $\mathcal{M}_1,\ \text{vec}_1(t)$, from Eq. (5.20) we get for the internal points

$$\begin{aligned}\mathcal{S}_m &= \frac{-u(x_{m-1},\,t) + u(x_{m+1},\,t)}{2\,\delta x} \\ &= \frac{\partial u}{\partial x}\bigg|_{x=x_m} + \frac{\delta x^2}{6}\,\frac{\partial^3 u}{\partial x^3}\bigg|_{x=x_m} + O(\delta x^4), \qquad m = 2,\ \ldots,\ m_{\text{end}} - 1.\end{aligned} \tag{5.24}$$

Furthermore, observe that, according to the Dirichlet and Neumann boundary conditions at $x = 0,\ d$, respectively, see Eq. (5.19), from Eq. (5.20) we have

$$\mathcal{S}_1 = \frac{-e^{\omega} + u(x_2,\ t)}{2\ \delta x} = \frac{\partial u}{\partial x}\bigg|_{x=x_1} + \frac{\delta x^2}{6}\ \frac{\partial^3 u}{\partial x^3}\bigg|_{x=x_1} + O(\delta x^4),$$

$$\mathcal{S}_{m_{\text{end}}} = 0 = \frac{\partial u}{\partial x}\bigg|_{x=d}.$$

Then, due to the shape of $\mathcal{M}_2,\ \text{vec}_2(t)$, from Eq. (5.20) we can write for the internal points

$$\mathcal{E}_m = \frac{-u(x_{m-1},\ t)^{\beta}\ \mathcal{S}_{m-1} + u(x_{m+1},\ t)^{\beta}\ \mathcal{S}_{m+1}}{2\ \delta x} + \mathcal{R}(u(x_m,\ t)), \quad (5.25)$$

$$m = 2,\ \ldots,\ m_{\text{end}} - 1.$$

That is, from Eq. (5.25), using the expressions of $\mathcal{S}_m,\ m = 2,\ \ldots,\ m_{\text{end}} - 1$ in Eq. (5.24) we have

$$\begin{aligned}\mathcal{E}_m &= \frac{\partial}{\partial x}\Big(u(x,t)^{\beta}\big(\frac{\partial u}{\partial x} + \frac{\delta x^2}{6}\ \frac{\partial^3 u}{\partial x^3} + O(\delta x^4)\big)\Big)\bigg|_{x=x_m} + O(\delta x^2) + \mathcal{R}(u(x_m,\ t)) \\ &= \mathcal{D}(u(x_m,\ t)) + O(\delta x^2) + \mathcal{R}(u(x_m,\ t)), \qquad m = 2,\ \ldots,\ m_{\text{end}} - 1.\end{aligned}$$

Exploiting that $\mathcal{P}(u(x_m,\ t)) = 0$, see Eq. (5.18) and Eq. (5.23), this proves the proposition for the interior points. Finally, we need to verify that the result also applies to the boundary points. From Eq. (5.20), Eq. (5.21), we have

$$\mathcal{E}_1 = \frac{-e^{\beta\omega}\ \mathcal{S}_0 + u(x_2,\ t)^{\beta}\ \mathcal{S}_2}{2\delta x} + \mathcal{R}(u(x_1,\ t)),$$

with

$$\mathcal{S}_0 = \frac{3e^{\omega} - 4u(x_1,\ t) + u(x_2,\ t)}{2\ \delta x},$$

$$\mathcal{E}_{m_{\text{end}}} = \frac{u(x_{m_{\text{end}}-2},t)^{\beta}\ \mathcal{S}_{m_{\text{end}}-2} - 4u(x_{m_{\text{end}}-1},t)^{\beta}\ \mathcal{S}_{m_{\text{end}}-1} + 3u(x_{m_{\text{end}}},t)^{\beta}\ \mathcal{S}_{m_{\text{end}}}}{2\ \delta x} + \mathcal{R}(u(x_{m_{\text{end}}},t)).$$

Observe that S_0 has been obtained via forward finite differences of order two for the approximation of $\frac{\partial u}{\partial x}\big|_{x=0}$. Then, proceeding in the same way as the above calculations, we get $\mathcal{E}_1 = \mathcal{D}(u(x_1,\ t)) + O(\delta x^2) + \mathcal{R}(u(x_1,\ t))$, and, using $\mathcal{P}(u(x_1,\ t)) = 0$, $\mathcal{E}_1 = O(\delta x^2)$. Finally, noting that, in $\mathcal{E}_{m_{\text{end}}}$, $\mathcal{S}_{m_{\text{end}}}$ has been approximated through backward finite differences of order two, the proof follows in the same way as in the previous cases. □

Using the above result, it is considered $\delta x = 5 \times 10^{-5}$ for the computation of the reference solution, thus getting a space error of approximately 10^{-9}.

5.4 IMPLEMENTATION

This section focuses on the description of the implementation of the proposed step-by-step time discrete PINNs to solve the PDE (5.18). For convenience, from now on we denote by SBS-PINNs (Step-By-Step PINNs) the new networks proposed in this paper, and by RK-PINNs (RK-based PINNs) the networks derived in the manuscript [135]. First of all, it is reported below Algorithm 1, which summarizes the implementation of both SBS-PINNs and RK-PINNs based on the programming language Python employing the package TensorFlow. Therefore, let us describe in detail the inputs and outputs of the algorithm, as well as all the main steps relating to it, denoted by the numbers $1,\ 2,\ \ldots,\ 10$ (in Algorithm 1).

Algorithm 1 Scheme of the implementation phase.

Input: x_0, d, M, $layers$, σ, $\mathrm{N_{epoch}}$, $\mathrm{lr_{initial}}$.
Output: the trained output of the network $f\,(x)$, see Figure 5.1.

1: Definition of the data $x^0 \in \mathbb{R}^M$, $x^1 = (0,\ d)^T \in \mathbb{R}^2$, $u^0 \in \mathbb{R}^M$.
2: Definition of the network model thorugh the input variables $layers$ and σ.
3: Introduction of the learning rate schedule and choose of the optimizer.
4: **for** $k = 1, \ldots, \mathrm{N_{epoch}}$ **do**
5: Compute the loss related to internal points, which corresponds to Eq. (5.5) for RK-PINNs, and to Eq. (5.10) or Eq. (5.12) for SBS-PINNs using Euler or Crank-Nicolson, respectively.
6: Compute the loss related to the boundary conditions, see Eq. (5.19).
7: Compute the total loss by adding the one related to the internal points (step 5) and the one related to the boundary conditions (step 6).
8: Compute the gradient with respect the network parameters.
9: Update the parameters of the network.
10: **end for**

The inputs are the following: the spatial domain limits, i.e. $x_0 = 0$ and d, see Eq. (5.18); the number of training spatial points M; the features of the neural networks, i.e. the number of hidden layers and neurons of each layer, stored in $layers$, and the activation function σ (consisting of a string variable); the number of training epochs $\mathrm{N_{epoch}}$; the initial learning rate $\mathrm{lr_{initial}}$. Recall that in the input layer there is just one neuron, while the output layer has n_{end} neurons for SBS-PINNs (i.e. the number of steps of Euler and Crank-Nicolson methods), and $q+1$ neurons for RK-PINNs (with q denoting the number of stages of the RK method). Therefore, the Python list $layers$ contains as first element 1, and as last element n_{end} or $q+1$; the internal elements correspond to the number of neurons of each hidden layer. Actually, it will be later shown

that, for the used implementation, the number of elements in the output layer for Euler and Crank-Nicolson based SBS-PINNs are $n_{\text{end}} + 1$ and $n_{\text{end}} + 2$, respectively. The outputs of the network are approximations of the u_n solution at each time step for SBS-PINNs, and approximations of the intermediate RK stages and advancing solution at a certain time step for RK-PINNs. For simplicity, and given that this is how we will proceed in the numerical tests, let assume to carry out only one step of the RK-PINN method, see Remark 1, starting from t_0; therefore, the last element of the RK-PINN output layer corresponds to the solution at the point t_e.
The step 1 of the implementation consists of identifying the training points, stored in the vector $x^0 \in \mathbb{R}^M$, defining also $u^0 \in \mathbb{R}^{\text{M}}$ having elements $u(x_i^0,\, 0)$, with x_i^0 i-th component of x^0; according to Eq. (5.19), in this case u^0 is the unit vector (since it is considered $\omega = 0$). In addition, let define the vector $x^1 = (0,\, d)^T \in \mathbb{R}^2$, containing the edge points of the spatial interval.
The step 2 consists of the definition of the network structure through the input variables $layers$ and σ. In numerical experiments (next section), a network based on 3 hidden layers with 10 neurons each (see the next Figure 5.2) is exploited, setting the hyperbolic tangent as activation function.
Step 3 aims to schedule the learning rate starting from the initial one, and to select the optimizer for the training phase of the network. In particular, the mentioned schedule imposes the use of different learning rates based on the training epochs. In numerical tests (next section), it is exploited four learning rates, setting the schedule in order to obtain the best performances for each experiment. The selected optimizer consists of Adam.
Step 4 consists of the beginning of the training phase. In step 5, the residuals are computed according to the applied method. For convenience, here it is exploited the formulation of the implicit Euler and Crank-Nicolson methods revisited as RK, see Proposition 5.2 and Proposition 5.3. In this way, by defining the variable p, corresponding to q for the RK-PINNs, to n_{end} for the SBS-PINNs based on Euler, and to $n_{\text{end}} + 1$ for the SBS-PINNs based on Crank-Nicolson, it can be expressed the loss function of all these three methods, in Eq. (5.5), Eq. (5.10), Eq. (5.12), respectively, as

$$\text{LOSS}_{\text{Method}} = \frac{1}{M} \sum_{i=1}^{M} (\text{LOSS}_i)^2, \tag{5.26}$$

where

$$\text{LOSS}_i = f(x_i^0) + \Delta t\, \mathcal{A}^T\, \mathcal{N}(f(x_i^0)) - u^0(x),$$

with

$$\mathcal{A} = \begin{pmatrix} A \\ b \end{pmatrix} \in \mathbb{R}^{p+1 \times p}, \tag{5.27}$$

being $A = (a_{ij})_{i,j=1}^{p} \in \mathbb{R}^{p\times p}$, $b = (b_i)_{i=1}^{p} \in \mathbb{R}^p$, suitable arrays. In particular: for RK-PINNs, these arrays correspond to those of the Butcher tableau of a classic RK method; for SBS-PINNs, these arrays correspond to those defined in Proposition 5.2, see Eq. (5.13), or Proposition 5.3, see Eq. (5.16). Using this approach, i.e. considering the Euler and Crank-Nicolson methods revisited as RK, it was possible to write codes related to the proposed approach inspired by those reported on github by Raissi et al.[2] [135], modifying them appropriately. Indeed, it was necessary to update the codes to the latest stable release of TensorFlow. Furthermore, a non-object-oriented approach was preferred in order to get greater understandability (for the developed code). Note that, through this 'unifying framework', it is possible to apply the desired method (RK-PINN or SBS-PINN) simply by properly choosing the matrix $\mathcal{A}$ in Eq. (5.27). With the mentioned methodology, the number of neurons in the output layer for SBS-PINNs changes slightly, see Figure 5.2 and related caption.
At step 6, the residuals related to the Dirichlet and Neumann boundary conditions in Eq. (5.19) are computed. In particular, they take the following expression, respectively:

$$\mathrm{Res}_0 = \frac{1}{p}\sum_{i=1}^{p}\left(f_i\left(0\right) - e^{\omega}\right)^2, \qquad \mathrm{Res}_d = \frac{1}{p}\sum_{i=1}^{p}\left(\frac{\partial f}{\partial x}\left(d\right)\right)^2. \tag{5.28}$$

The boundary loss is thus given by the sum of the two residuals in Eq. (5.28). Step 7 computes the global loss function of the network, given by

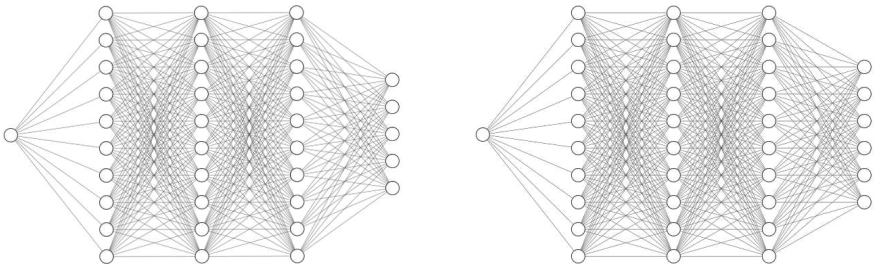
$$\mathrm{LOSS} = \mathrm{LOSS}_{\text{Method}} + \mathrm{Res}_0 + \mathrm{Res}_d,$$

see Eq. (5.26) and Eq. (5.28). Step 8 and step 9 are usual for a network training phase. At step 10 the current epoch ends.

5.5 NUMERICAL RESULTS

This section is dedicated to numerical experiments, which are conducted on the sustainability PDE model described in Section 5.3. In particular, the non-dimensional equation (5.18) equipped with the initial and boundary conditions (5.19) is solved. First, the tests compare the RK-PINNs and SBS-PINNs, also showing that the latter are performing on integrations over long time intervals (Subsection 5.5.1). Then, it is shown how the SBS-PINNs introduced in the manuscript can be used to solve inverse problems (Subsection 5.5.2). Finally (Subsection 5.5.3), some experiments highlight that SBS-PINNs compete with and/or are advantageous to classical PINNs (which are continuous in time and space).

[2] https://github.com/maziarraissi/PINNs/tree/master



(a) **(b)**

Figure 5.2. Subfigure (a) shows the structure of the SBS-PINN based on Euler method described in Section 5, and subfigure (b) does the same regarding the SBS-PINN based on Crank-Nicolson method, for $n_{\text{end}} = 4$; note that the output layer now contains $n_{\text{end}} + 1$ neurons for the first, and $n_{\text{end}} + 2$ neurons for the second, since SBS-PINNs have been revisited as RK schemes in the implementation phase, see Proposition 5.2 and Proposition 5.3.

5.5.1 Comparison with existing time discrete PINNs and solution over long time intervals

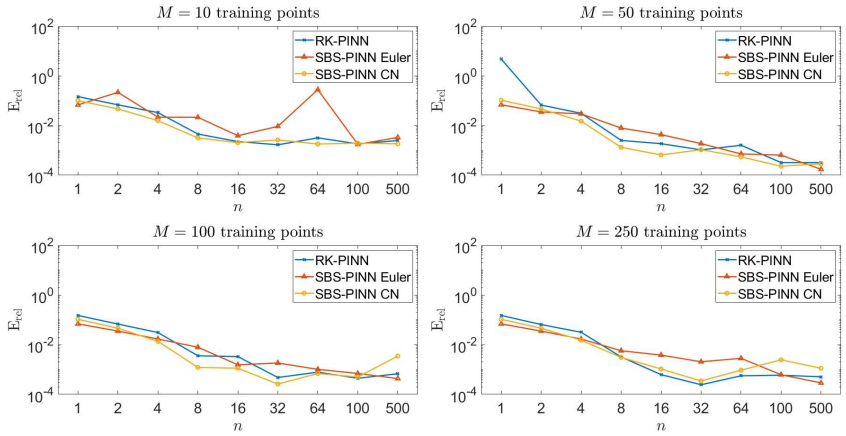


Figure 5.3. Results obtained by RK-PINNs and SBS-PINNs, with both Euler and Crank-Nicolson, for $n = 1, 2, 4, 8, 16, 32, 64, 100, 500$ and several training points, i.e. $M = 10, 50, 100, 250$; the initial learning rates is 5E-4, for the three methods, and the number of epochs is 30000.

In all the figures and tables, the relative errors are computed at the last time grid point t_e, comparing the numerical solutions with the reference one determined as described in Section 4. The first step of the experimental phase consists of a comparison between RK-PINNs and SBS-PINNs, by choosing several values of n, i.e. $n = 1, 2, 4, 8, 16, 32, 64, 100, 500$, and a different number of training

points (equidistant in the interval $[0,\ d]$), i.e. $M = 10, 50, 100, 250$. In this section, we denote by n the number of stages of the RK method for RK-PINNs, and the number of steps of the one-stage methods related to SBS-PINNs. The Butcher tableau of the RK methods, for RK-PINNs, has been taken by the github repository of Raissi et al.[3]. In the experiments, we exploit a network having 3 hidden layers with 10 neurons each, selecting a starting learning rate equal to 5E-04, and 30000 epochs.

Table 5.1. Relative errors estimated with respect to the benchmark solution presented in Subsection 5.3.1 at time point $t = t_e$, by RK-PINNs and SBS-PINNs with $M = 10$ spatial training points.

n	**RK-PINN**	**SBS-PINN Euler**	**SBS-PINN CN**
1	1.45E-01	6.74E-02	1.00E-01
2	6.78E-02	2.15E-01	4.65E-02
4	3.30E-02	2.16E-02	1.56E-02
8	4.53E-03	2.13E-02	3.15E-03
16	2.25E-03	3.88E-03	2.04E-03
32	1.67E-03	9.20E-03	2.62E-03
64	3.17E-03	2.76E-01	1.80E-03
100	1.82E-03	1.74E-03	1.94E-03
500	2.46E-03	3.29E-03	1.84E-03

Table 5.2. Relative errors estimated with respect to the benchmark solution presented in Subsection 5.3.1 at time point $t = t_e$, by RK-PINNs and SBS-PINNs with $M = 50$ spatial training points.

n	**RK-PINN**	**SBS-PINN Euler**	**SBS-PINN CN**
1	4.69E+00	6.84E-02	1.04E-01
2	6.62E-02	3.50E-02	4.61E-02
4	3.09E-02	2.93E-02	1.49E-02
8	2.50E-03	7.88E-03	1.32E-03
16	1.85E-03	4.28E-03	6.51E-04
32	1.04E-03	1.86E-03	1.06E-03
64	1.60E-03	7.16E-04	5.43E-04
100	3.21E-04	6.40E-04	2.30E-04
500	3.19E-04	1.73E-04	2.85E-04

The obtained results are summarized in Tables 5.1, 5.2, 5.3, and 5.4 and Figure 5.3. Note that, for $n = 500$, the SBS-PINNs achieve lower errors than RK-PINNs for all the values of M. In particular, for $M = 10$ the best performances are those of the Crank-Nicolson PINN; instead, for subsequent values of M,

[3] https://github.com/maziarraissi/PINNs/tree/master/Utilities/IRK_weights

Table 5.3. Relative errors estimated with respect to the benchmark solution presented in Subsection 5.3.1 at time point $t = t_e$, by RK-PINNs and SBS-PINNs with $M = 100$ spatial training points.

n	**RK-PINN**	**SBS-PINN Euler**	**SBS-PINN CN**
1	1.49E-01	6.83E-02	1.04E-01
2	6.72E-02	3.50E-02	4.60E-02
4	3.06E-02	1.68E-02	1.35E-02
8	3.55E-03	7.83E-03	1.21E-03
16	3.29E-03	1.54E-03	1.12E-03
32	4.75E-04	1.82E-03	2.59E-04
64	7.78E-04	1.01E-03	6.84E-04
100	4.41E-04	6.89E-04	4.97E-04
500	6.72E-04	4.30E-04	3.46E-03

Table 5.4. Relative errors estimated with respect to the benchmark solution presented in Subsection 5.3.1 at time point $t = t_e$, by RK-PINNs and SBS-PINNs with $M = 250$ spatial training points.

n	**RK-PINN**	**SBS-PINN Euler**	**SBS-PINN CN**
1	1.49E-01	6.83E-02	1.04E-01
2	6.41E-02	3.50E-02	4.57E-02
4	3.13E-02	1.68E-02	1.50E-02
8	3.20E-03	5.67E-03	3.07E-03
16	6.17E-04	3.78E-03	1.04E-03
32	2.43E-04	2.04E-03	3.43E-04
64	5.55E-04	2.80E-03	9.35E-04
100	5.86E-04	6.13E-04	2.44E-03
500	5.06E-04	2.88E-04	1.12E-03

it is the Euler method that achieves the lowest errors (always for $n = 500$). Note also that, for any value of M, the Crank-Nicolson based SBS-PINN shows satisfactory results already with $n = 16$; that is, unlike other methods, it does not need higher n values to achieve good errors.
We also note that SBS-PINNs need fewer training points than RK-PINNs to achieve errors of the same order of magnitude. That is, for $M = 10, 50, 100$, the Euler and Crank-Nicolson based SBS-PINNs achieve lower errors than the RK-PINNs, for the same n values. For $M = 500$, the RK-PINN seems to perform better than the SBS-PINNs, except for the value $n = 500$; in this case, the best error is given by Euler. These observations suggest that SBS-PINNs are less dependent on the number of training points than RK-PINNs. We emphasize that such results are obtained using a network with a low number of hidden layers and neurons for each.

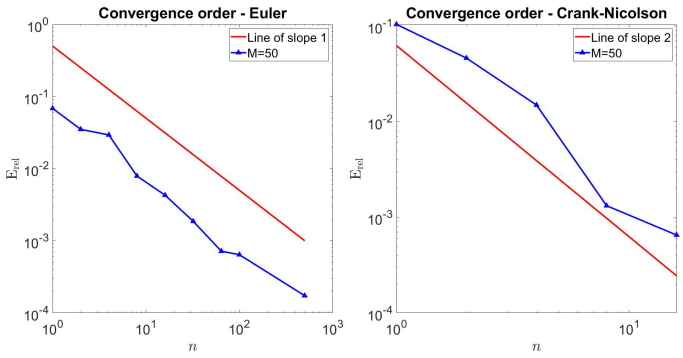


Figure 5.4. Empirical evaluation of the convergence order related to the Euler and Crank-Nicolson SBS-PINNs.

A very interesting comparison is presented in Figure 5.4; it shows the empirical evaluation of the (time) order of convergence of the SBS-PINNs methods, based on the results reported in Table 5.2 for $M = 50$. To realize the plots, we took all values of n for Euler, and up to $n = 16$ for Crank-Nicolson. This choice is made for Crank-Nicolson, since for $n > 16$ the error does not decrease. This may be due to various factors, i.e. by the structure of the network, by the way in which the spatial derivatives involved in the loss are computed by the network, by the training points and so on. Therefore, the solution computed by the network is affected by errors in time and space (compared to the exact or reference solution): for small values of n, it is the error in time that is dominant, and this can allow to observe the the order of convergence of the used methods; instead, for large values of n, the spatial error can prevail. Note, from Figure 5.4, that the Euler and Crank-Nicolson PINNs methods appear to show their effective orders of convergence, which are equal to one and two, respectively. Therefore, it seems that the SBS-PINNs proposed in this manuscript may be able to inherit

the properties of the methods embedded in them, such as consistency, but also stability, given that the numerical solution is well computed by them even for small values of n (i.e. for large values of the step-size δt).

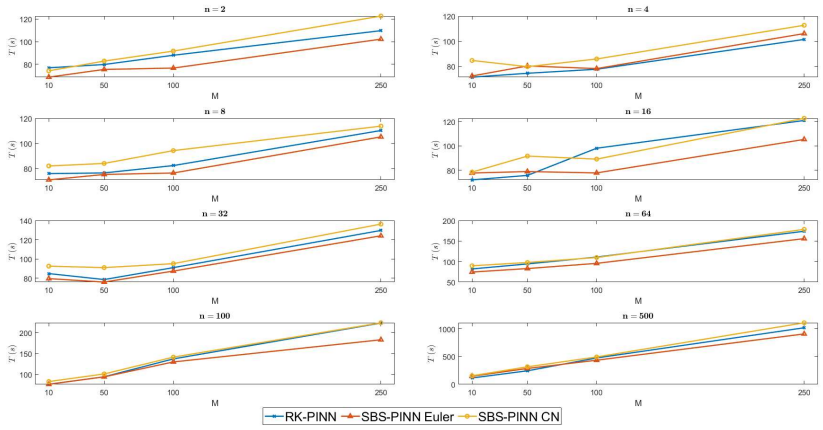


Figure 5.5. Execution times of RK-PINNs and SBS-PINNs based on Euler and Crank-Nicolson for several values of the number of training points M.

Furthermore, let us evaluate the execution time of the SBS and RK-based PINNs. Both PINNs are implemented with Jupyter Notebooks available via Google Colab. Recall that these PINNs share the same workflow, see Section 5.4. Also, recall the SBS-PINNs based on the Crank-Nicolson method require an additional neuron in the output layer (consult Proposition 5.3). The obtained results, in Figure 5.5, show the average of eleven runs for each experiment; the plots are related to $n = 2, 4, 8, 16, 32, 64, 100, 500$, using 30000 epochs. Each plot shows the execution time related to $M = 10, 50, 100, 250$ training points (equally spaced). The results confirm that the computing times of SBS-PINNs and RK-PINNs are similar: actually, SBS-PINNs with Euler seem to be characterized by the lowest effort; SBS-PINNs with Crank-Nicolson take slightly longer times, probably due to the larger size of the output layer. In any case, the computing times of the three PINNs are very close to each other, and this is in agreement with the used workflow. At similar computational cost, it is recalled however that SBS-PINNs are more flexible than RK-PINNs, see e.g. Remark 4. Indeed, for problems defined e.g. over large time intervals, we may need to consider small values of the step-size δt to obtain a sufficiently accurate solution. With SBS-PINNs, δt can be varied immediately without implementation efforts. On the other hand, with RK-PINNs, varying the number of stages means recalculating all the coefficients of the corresponding method; this is not a trivial task.
Finally, in Figure 5.6 the behaviour of SBS-PINNs based on the Euler and

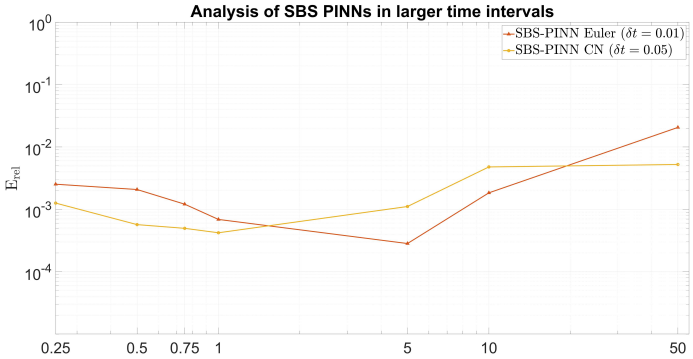


Figure 5.6. relative error at $t = t_e$ by SBS-PINNs applied on the time interval $[0, t_e]$, $t_e = 0.25, 0.5, 0.75, 1, 5, 10, 50$, with fixed step-size δt; the network exploits $M = 50$ (equidistant) spatial training points and 30000 epochs.

Crank-Nicolson methods is evaluated, on long time integrations. In particular, the developed experiment consists of the solution of the DSSCs model for $t \in [0, t_e]$, $t_e = 5, 10, 50$ (in the plot it is also reported the results for $t_e = 0.25, 0.5, 0.75, 1$), with a fixed time-step δt; it is considered $\delta t = 0.01$ for Euler, and $\delta t = 0.05$ for Crank-Nicolson. Figure 5.6 shows the error trend by training the network in the considered intervals; the error is evaluated with respect to the benchmark solution, as already described. For both PINNs (based on Euler and Crank-Nicolson) the error has order of magnitude 10^{-3} for small values of t_e (up to about $t_e = 5$), and 10^{-2} at $t_e = 10, 50$. These results therefore show that the stability of classically used numerical methods is confirmed also through their immersion in PINNs, and so the error remains bounded even for integrations over long times.

5.5.2 *STEP-BY-STEP TIME DISCRETE PINNS FOR INVERSE PROBLEMS*

This subsection shows the use of the proposed approach on inverse problems. That is, the aim lies in identifying the coefficients of parametric PDEs by collecting data related to the phenomena under consideration. The followed approach is similar to the one described in Paragraphs 4.1.1 and 4.2.1 of the manuscript [135]; indeed, there the authors explain how to use discrete PINNs based on RK methods in this context.
Let us thus consider two parameterized versions of the DSSCs model (5.18): one characterized by a parameter λ_1 linked to the non-linear diffusion coefficient, see (5.29); one characterized by a parameter λ_2 linked to the reaction term, see (5.30). The other parameters of the DSSCs model are fixed as described previously. The two versions just mentioned then take the following

form, respectively:

$$\frac{\partial u}{\partial t} - \lambda_1 \frac{\partial}{\partial x}\left(u^\beta \frac{\partial u}{\partial x}\right) - \mu e^{-\nu x} + \xi u^\beta (u-1) = 0, \tag{5.29}$$

$$(x,t) \in [0,\ d] \times [0,\ t_e];$$

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(u^\beta \frac{\partial u}{\partial x}\right) + \lambda_2 \left(-\mu e^{-\nu x} + \xi u^\beta (u-1)\right) = 0, \tag{5.30}$$

$$(x,t) \in [0,\ d] \times [0,\ t_e].$$

The data collected consists of 10000 space points, at $t_e = 1$, on the benchmark solution. The SBS-PINNs are applied with $n = 500$ (Euler), and $n = 16$ (Crank-Nicolson), as these values lead to the best results. The network structure is the one already described in the Subsection 5.5.1. It is considered $M = 50$ spatial training points for the loss related to the physics of the problem and to the initial and boundary conditions. Table 5.5 reports the obtained parameters estimates. Note that, even though strategies to weight physical and data losses are not applied, the results are good: indeed, the Euler SBS-PINN estimates λ_1 and λ_2 with relative errors around 10^{-3} and 10^{-2}, respectively; the same happens for the Crank-Nicolson SBS-PINN, for which the errors are also slightly lower.

Table 5.5. Results in estimating the parameters λ_1 and λ_2 by employing data obtained from benchmark solution; Table (a) refers to problem (5.29); Table (b) to problem (5.30).

(a) Estimation of λ_1 through the approximation $\hat{\lambda}_!$ in the inverse problem (5.29). Numerical experiments require 20000 epochs to train the networks related to SBS-PINNs based on Euler and Crank-Nicolson.

Method	λ_1	$\hat{\lambda}_1$	$\mathrm{E}_{\mathrm{rel}}$
SBS-PINN Euler ($n = 500$)	1.0	1.00526554	5.2655E-03
SBS-PINN CN ($n = 16$)	1.0	0.99702824	2.9718E-03

(b) Estimation of λ_2 through the approximation $\hat{\lambda}_2$ in the inverse problem (5.30). Numerical experiments require 60000 and 30000 epochs to train the networks related to SBS-PINNs based Euler and Crank-Nicolson, respectively.

Method	λ_2	$\hat{\lambda}_2$	$\mathrm{E}_{\mathrm{rel}}$
SBS-PINN Euler ($n = 500$)	1.0	0.90025620	9.9744E-02
SBS-PINN CN ($n = 16$)	1.0	0.96105432	3.8946E-02

To conclude the subsection, let mention that SBS-PINNs can be extended also to other types of problems (not only inverse problems). For example, for a peridynamic problem, in the manuscript [58] the authors consider two concatenated neural networks: one for the estimation of the kernel function; one for computing the time and space dependent solution. Since, like classical

PINNs, also SBS-PINNs solve PDEs (the main difference lies in the fact that they take as input only a point of the space grid, and not also a point of the time grid), the mentioned approach could be used for them as well. Similar considerations apply to general problems with integral discretization, see e.g. [186].

5.5.3 *Comparison with continuous PINNs*

The final subsection of the experimental phase shows a comparison between the proposed SBS-PINNs and the state-of-the-art continuous (in space and time) PINNs. The latter are implemented exploiting the PINA (Physics Informed Neural network for Advanced modeling) package, deployed by the MathLab group of SISSA[4] [47].
For the SBS-PINNs based on Euler and Crank-Nicolson, the same structure declared previously is considered and $n = 500$, $n = 16$, respectively, as these values lead to the best results. Table 5.6 describes instead the structure of the continuous PINN used. Note that two different continuous PINNs (i.e. structured differently) have been considered: the first has three hidden layers with 10 neurons; the second has two hidden layers with 100 neurons. For both networks: the input layer has two neurons, which are related to spatial (x) and temporal (t) variables; the output layer has one neuron, which is associated to the approximation of the solution evaluated at the input point (x, t). Table 5.7 shows relative errors obtained by executing the SBS-PINNs and continuous PINNs described in Table 5.6. The errors are evaluated at four space points, i.e. $x = 0.25, 0.5, 0.75, 1$. Note that the proposed step-by-step approach leads to lower errors than continuous PINNs. Also, the run times (evaluated through the Jupyter Notebooks, Google Colab) of SBS-PINNs (285 s with Euler and 92 s with Crank-Nicolson) are lower than those of the continuous PINNs (1157 s with Network 1 and 1378 s with Network 2, see Table 5.6). This happens because continuous PINNs need a larger number of (equidistant) training points in space, i.e. $M = 500$, compared to the SBS-PINNs proposed by us, i.e. $M = 50$, to produce the shown results in terms of achieved errors.

5.5.4 *Discussion*

The numerical experiments support the effectiveness of the proposed step-by-step PINN (SBS-PINN) paradigm. Here, a one-stage time integrator is embedded directly into the neural architecture. This enables the network to output the solution at all discrete time levels. This design produces continuous-in-space, discrete-in-time approximations. In the context of historical assets subject to slow, cumulative degradation processes, this formulation is particularly rele-

[4] https://github.com/mathLab/PINA

Table 5.6. Features of the two network structures associated with the implemented continuous PINNs.

	Network 1	**Network 2**
Hidden Layers	3	2
Neurons in each Hidden Layer	10	100
Training Collocation Points	500	500
Training Initial Points	50	50
Training Boundary Points	50 each condition	50 each condition
Epochs	30000	30000
Learning Rate	5E-4	5E-4
Decay Rate	1E-8	1E-8

Table 5.7. Comparison between SBS-PINNs and the continuous PINNs described in Table 5.6.

x	**SBS-PINN Euler**	**SBS-PINN CN**	**Cont. PINN Network 1**	**Cont. PINN Network 2**
0.25	4.69E-04	9.11E-04	8.63E-02	8.54E-02
0.50	2.50E-04	6.31E-04	1.21E-01	1.16E-01
0.75	6.19E-04	3.67E-04	1.33E-01	1.34E-01
1.0	1.73E-04	6.51E-04	1.35E-01	1.44E-01

vant, as it mirrors the discrete acquisition of monitoring data and supports the consistent reconstruction of long-term temporal evolutions of physical state variables on complex geometries. It also preserves a workflow comparable to that of existing time-discrete PINNs. A key first outcome is that SBS-PINNs achieve competitive, and in several configurations improved, accuracy compared to RK-based discrete PINNs, especially when the number of spatial training points is limited. This suggests that the step-by-step formulation reduces sensitivity to the sampling density of the spatial residual points, yielding satisfactory results even with relatively few spatial training points. Moreover, the empirical convergence plots indicate that the SBS-PINNs can inherit the effective temporal order of the embedded integrators (first order for Euler and second order for Crank–Nicolson) until spatial and training-related errors become dominant, which is consistent with the interpretation of the overall error as a combination of time-discretization and approximation/training components. The experiments over extended time horizons further highlight that embedding classical stable schemes within the PINN framework yields bounded errors, even for long-time integrations. While the error increases for large final times, it remains under control. This supports the claim that the stability properties of standard numerical methods carry over to the learning-based setting when the method is enforced step-by-step.
From a practical standpoint, SBS-PINNs offer notable flexibility. The time

step can be changed without reformulating method coefficients. In contrast, RK-PINNs require recomputing the associated tableau if the number of stages varies. This flexibility makes SBS-PINNs appealing when a small time step is needed for accuracy or when the time grid must be refined adaptively.
Finally, the inverse-problem experiments show that the step-by-step approach can effectively couple physics constraints with observational data. This enables identification of PDE parameters, even without specialized loss reweighting strategies. Comparisons to continuous-space–time PINNs show SBS-PINNs achieve lower errors and shorter runtimes due to fewer required collocation points. These results indicate that SBS-PINNs are a promising and efficient alternative for stiff or long-time PDE dynamics and for data-augmented identification tasks.
From a practical point of view, applying step-by-step rules for what happens over time directly handles a main need in predicting how historical objects change. Things like wear, heat, and environmental effects develop over long periods, and we often have only occasional measurements to observe them. The structure that supports separate time points aligns well with timed sensor readings and check-ups, making it easier to update predictions as new data comes in. This approach is especially useful in digital twins for cultural heritage, where it's important to reliably predict what will happen over time in a way that matches real physical changes. This helps assess risks, decide when to act, and support long-term care strategies as conditions and situations change.

Chapter 6

Conclusions and future perspectives

The digital transformation of the Cultural Heritage (CH) domain has profoundly reshaped the documentation, interpretation, and experience of historical artefacts, archaeological sites, and museum collections [12, 124]. As the CH field increasingly leverages digital tools, immersive media, and data-driven technologies, institutions must balance enhanced access and engagement with long-term preservation. Digital tools enable innovative modes of interaction, offering visitors the opportunity to explore heritage spaces through adaptive narratives, personalised itineraries, and hybrid physical–digital experiences [12, 34]. At the same time, these tools provide conservation experts with quantitative insights into material degradation, environmental dynamics, and structural behaviour [50]. Nevertheless, despite these opportunities, the CH domain remains a challenging application context: it is characterised by heterogeneous data, non-standardised documentation practices, strict ethical constraints, and the need to protect fragile, often irreplaceable assets [12, 124]. Additionally, heritage institutions must address the complex challenge of serving audiences with highly diverse expectations, further complicating their mission. Within this scenario, two main research challenges emerge, each introducing distinct methodological and technological difficulties: enhancing CH enjoyment for cultural users and ensuring effective predictive maintenance of cultural assets. In line with the motivations and objectives introduced in Chapter 1, this dissertation contributes along three complementary directions:

- a unified, modular framework that enables reuse of data, models, and services;
- contextual, user-centred methods that improve cultural users' enjoyment through context-aware recommendation, path elaboration, and narrative delivery;
- physics-based and physics-informed approaches that support predictive

maintenance through reliable simulations, including methodological advances for time-dependent problems.

The first challenge is to design cultural experiences that are contextualised, adaptive, and user-centred [12, 124]. Visitors increasingly expect digital interactions that reflect their interests, prior knowledge, mobility constraints, and spatial or temporal context [36]. However, cultural content is typically extensive, heterogeneous, and semantically layered, making it challenging to organise and present it in a meaningful, non-intrusive way [12, 34]. To address these issues, research efforts are shifting toward Context-Aware Recommender Systems that leverage user behaviour, contextual variables, and environmental data to generate tailored recommendations [34, 39]. In the CH domain, such systems have been employed to personalise museum visits, cultural tourism itineraries, and on-site navigation, demonstrating improvements in relevance, engagement, and visitor satisfaction [34, 124]. Also relevant is the role of Digital Storytelling, which helps create narrative structures that link objects and places through clear interpretative frameworks and stories [82, 96, 138]. These approaches are increasingly integrated into architectures that combine storytelling with recommendations, to generate adaptive cultural paths that dynamically reorganise visit itineraries for different user profiles and contexts [12, 82, 96].

From a methodological perspective, visitor-oriented enhancement introduces additional constraints that are often underestimated in generic recommender system deployments. First, contextual modelling in CH must remain transparent and non-invasive: personalisation should not undermine the visitor's agency nor reduce the interpretability of suggested cultural paths, especially when recommendations are intertwined with narrative structures [12, 34]. Second, the system must operate with data that is often heterogeneous and incomplete. User traces may be sparse or ethically restricted, and knowledge bases may differ in standards and semantic granularity [12, 124]. Third, CH experiences are intrinsically multi-objective. Recommendations must balance enjoyment, learning, accessibility, crowd management, and the coherence of the interpretative journey [36]. In this dissertation, these aspects motivate the adoption of a unified architectural perspective in which recommendation is not an isolated service, but a component that can reuse shared representations and contextual information, integrating non-linear storytelling and adaptive interaction patterns [12, 34, 82, 96].

In parallel with visitor-oriented enhancements, the second main issue is the critical task of monitoring and preserving heritage structures, artefacts, and materials. CH asset decay is often complex, occurs in ways that are hard to measure or predict, and is influenced by the environment. Standard numerical methods and purely data-driven methods often struggle with missing data, noise, and the changing nature of historical materials. Recent advances in

Scientific Machine Learning (SciML) offer new opportunities for predictive maintenance [51]. Physics-Informed Neural Networks (PINNs) merge physical laws and observational data. They model structural behaviour and predict degradation trends over time [5, 51]. PINNs have been successfully applied to structural and continuum mechanics problems, including plate and frame structures, showing improved generalisation and physical consistency compared to black-box models [5]. Combined with digital twins and monitoring infrastructures, SciML and PINNs support predictive maintenance strategies. These approaches move from reactive inspections to continuous, model-based risk assessment, offering more reliable guidance for protecting and managing CH assets [50, 75, 149].

However, predictive maintenance in CH is not just about selecting a modelling method. It is also about integration. Heritage assets are tracked using various types of sensors and methods, often yielding uneven, sparse, and partially observed data. Moreover, physical models available in the literature may be incomplete or too idealised for specific artefacts, requiring calibration against noisy observations and the ability to incorporate domain constraints. In this setting, SciML bridges mechanistic modelling and real-world data [51]. Deploying SciML introduces practical challenges: processing complex geometries, managing multi-source data streams, and providing interpretable outputs that conservation experts can trust [50, 75, 149]. These issues motivate architectural solutions that explicitly separate data acquisition, knowledge representation, and inference modules, enabling reproducibility and scalability across assets and use cases.

Taken together, the two research directions above highlight a fundamental limitation of the current digital CH landscape: enhancement- and conservation-oriented systems are often developed as isolated solutions, leading to fragmented infrastructures and limited reuse of data, models, and evaluation protocols [124, 34, 50]. A primary aim of this dissertation is to address this limitation through a unified, modular, and asset-aware framework that explicitly integrates

- acquisition of heterogeneous data and digital models;
- knowledge organisation for reuse across tasks;
- inference engines tailored to both user-facing and expert-facing objectives;
- visualisation and decision-support services.

Within this framework, recommendation, storytelling, and predictive maintenance are treated as interchangeable inference workflows that share a common digital backbone, enabling scalability across institutions and assets while reducing the engineering cost of deploying new CH services [34, 124].

The experimental evidence discussed throughout the thesis further supports the effectiveness of this architectural choice. On the enhancement side, the proposed approach confirms that recommendation quality and user satisfaction

can be improved through context-aware modelling and the integration of adaptive interaction mechanisms, including non-linear storytelling and immersive media [12, 34, 82, 96, 138, 36, 39]. In particular, the recommendation engine benefits from representing context in a structured form and from combining dimensionality reduction with contextualised biases, resulting in improvements in accuracy and in-situ satisfaction compared to baseline factorisation-based approaches [34, 39]. On the conservation side, the thesis demonstrates that combining physical knowledge with data-driven learning enables reliable simulations for degradation-related dynamics, strengthening predictive maintenance pipelines in scenarios where purely numerical or purely data-driven strategies may be insufficient [50, 51, 5, 75, 149]. Crucially, these results are obtained while maintaining interoperability with complex geometries and real-world acquisition constraints, which are central in CH practice [50, 124].

A further scientific aim of the dissertation concerns the methodological advancement introduced for time-dependent differential problems. While continuous-in-time PINNs provide a framework for blending physical laws and data [51, 5], they may suffer from optimisation issues in evolutionary settings, including imbalance in the loss function and limited enforcement of temporal causality [51]. To address the causality limitation, Chapter 5 introduces a step-by-step time-discrete formulation that integrates classical numerical time-integration schemes into the PINN training loss, resulting in a model that remains continuous in space while being discretised in time. In this approach, implicit one-step methods (e.g., Implicit Euler and Crank–Nicolson) are embedded into the loss, so that the network advances the solution sequentially along the time grid, enforcing causality by construction and improving stability over long horizons [168]. This hybrid methodology provides a principled bridge between established step-by-step schemes and physics-informed learning, and it is particularly relevant for CH predictive maintenance problems where degradation dynamics are time-dependent, potentially stiff, and often observed through sparse measurements [50, 51, 75, 149]. Accordingly, the chapter complements the framework-level contribution with a methodological tool that increases robustness and computational efficiency when tackling time-dependent PDEs within the proposed digital infrastructure.

Overall, the thesis contributes to the digital CH landscape in three synergistic ways. First, it proposes an architectural solution that unifies enhancement-oriented and conservation-oriented services within a coherent pipeline, enabling reuse of data and digital replicas across tasks [34, 50, 124]. Second, it provides methodological components, including SciML-based solvers and PINN/ROM strategies, that increase the reliability of simulations in complex, data-constrained degradation scenarios [51, 5, 75, 149]. Third, it introduces a time-discrete PINN strategy that strengthens temporal causality and long-horizon stability in evolutionary problems [168]. In this perspective, the long-

term value of the dissertation lies in enabling the transition from isolated prototypes toward integrated digital ecosystems where IoT, AI, and physically consistent modelling jointly support both cultural enjoyment and sustainable preservation [34, 50, 124].

Finally, the challenges outlined in this section suggest clear directions for future work. On the enhancement side, a promising line concerns strengthening conversational and interactive interfaces to reduce cognitive overload while increasing the transparency of personalisation and the coherence of narrative-driven recommendations [12, 34, 82, 96]. On the conservation side, future developments include strategies for model discovery and inference of physical dynamics when literature models are missing or incomplete, as well as specialised workflows for Structural Health Monitoring of historic buildings, where monitoring data and physics-informed models can jointly support risk-informed maintenance [50, 51, 75, 149]. In both cases, a key long-term objective is to evolve from isolated prototypes toward integrated digital ecosystems that systematically combine IoT, AI, and physical modelling to support both cultural enjoyment and sustainable preservation.

References

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. *Lect. Notes Comput. Sci.*, 1707:304 – 307, 1999. DOI: 10.1007/3-540-48157-5_29.

[2] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware recommender systems. *AI Mag.*, 32(3):67 – 80, 2011. DOI: 10.1609/aimag.v32i3.2364.

[3] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103 – 145, 2005. DOI: 10.1145/1055709.1055714.

[4] G. Adomavicius and A. Tuzhilin. *Context-aware recommender systems*. 2015.

[5] A. I. F. Al-Adly and P. Kripakaran. Physics-informed neural networks for structural health monitoring: a case study for kirchhoff–love plates. *Data-Centric Eng.*, 5(4), 2024. DOI: 10.1017/dce.2024.4.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.*, 17(4):2347 – 2376, 2015. DOI: 10.1109/COMST.2015.2444095.

[7] T. M. Al-Hasan, A. N. Sayed, F. Bensaali, Y. Himeur, I. Varlamis, and G. Dimitrakopoulos. From traditional recommender systems to gpt-based chatbots: A survey of recent developments and future directions. *Big Data Cogn. Comput.*, 8(4), 2024. DOI: 10.3390/bdcc8040036.

[8] S. J. Anagnostopoulos, J. D. Toscano, N. Stergiopulos, and G. E. Karniadakis. Residual-based attention in physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.*, 421, 2024. DOI: 10.1016/j.cma.2024.116805.

[9] L. Andrade, J. Sousa, H. Aguilar Ribeiro, and A. Mendes. Phenomenological modeling of dye-sensitized solar cells under transient conditions. *Sol. Energy*, 85(5):781–793, 2011. DOI: 10.1016/j.solener.2011.01.014.

[10] L.-M. Angheluță, A. Ignuța Acimov, C. Gora, A. I. Chiricuță, A. I. Popovici, and V. Obradovici. Documenting romania's wooden churches: Integrating modern digital platforms with vernacular conservation. *Heritage*, 8(3), 2025. DOI: 10.3390/heritage8030103.

[11] J. A. Anta, F. Casanueva, and G. Oskam. A numerical model for charge transport and recombination in dye-sensitized solar cells. *J. Phys. Chem. B*, 110(11):5372–5378, 2006. DOI: 10.1021/jp056493h.

[12] L. Ardissono, T. Kuflik, and D. Petrelli. Personalization in cultural heritage: The road travelled and the one ahead. *User Model. User-Adapt. Interact.*, 22(1-2):73 – 99, 2012. DOI: 10.1007/s11257-011-9104-x.

[13] C. Ardito, P. Buono, G. Desolda, and M. Matera. From smart objects to smart experiences: An end-user development approach. *Int. J. Hum. Comput. Stud.*, 114:51 – 68, 2018. DOI: 10.1016/j.ijhcs.2017.12.002.

[14] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Comput. Netw.*, 54(15):2787 – 2805, 2010. DOI: 10.1016/j.comnet.2010.05.010.

[15] C. Aumaitre, C. Rodriguez-Seco, J. Jover, O. Bardagot, F. Caffy, Y. Kervella, N. López, E. Palomares, and R. Demadrille. Visible and near-infrared organic photosensitizers comprising isoindigo derivatives as chromophores: Synthesis, optoelectronic properties and factors limiting their efficiency in dye solar cells. *J. Mater. Chem. A*, 6(21):10074–10084, 2018. DOI: 10.1039/c8ta01826j.

[16] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, et al. Basic research needs workshop for scientific machine learning: Core technologies for artificial intelligence. *Document prepared for Department of Energy Advanced Scientific Computing Research, USA*, 10, 2019.

[17] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recom-

mendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, page 301–304, 2011.

[18] L. Baltrunas and F. Ricci. Context-based splitting of item ratings in collaborative filtering. In *RecSys'09*, page 245–248, 2009.

[19] L. Baltrunas and F. Ricci. Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Model. User-Adapt. Interact.*, 24(1-2):7 – 34, 2014. DOI: 10.1007/s11257-012-9137-9.

[20] A.s Barakat and P. Bianchi. Convergence and dynamical behavior of the adam algorithm for nonconvex stochastic optimization. *SIAM J. Optim.*, 31(1):244 – 274, 2021. DOI: 10.1137/19M1263443.

[21] B. R. Barricelli, E. Casiraghi, and D. Fogli. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 7, 2019. DOI: 10.1109/ACCESS.2019.2953499.

[22] M. Bazire and P. Brézillon. Understanding context before using it. *Lect. Notes Comput. Sci.*, 3554 LNAI:29 – 40, 2005. DOI: 10.1007/11508373_3.

[23] A. Belussi, A. Cinelli, A. Dalla Vecchia, S. Migliorini, M. Quaresmini, and E. Quintarelli. Forecasting poi occupation with contextual machine learning. *Lect. Notes Comput. Sci.*, 13389 LNCS:361 – 376, 2022. DOI: 10.1007/978-3-031-15740-0_26.

[24] P. Benner, S. Grivet Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira. *Snapshot-Based Methods and Algorithms*, volume 2. De Gruyter, 2020.

[25] P. Benner, S. Grivet Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira. *System- and Data-Driven Methods and Algorithms*, volume 1. De Gruyter, 2021.

[26] D. Bingham, T. Butler, and D. Estep. Inverse problems for physics-based process models. *Annu. Rev. Stat. Appl.*, 11(1):461 – 482, 2024. DOI: 10.1146/annurev-statistics-031017-100108.

[27] R. G. Boboc, E. Băutu, F. Gîrbacia, N. Popovici, and D. Popovici. Augmented reality in cultural heritage: An overview of the last decade of applications. *Appl. Sci. (Switz.)*, 12(19), 2022. DOI: 10.3390/app12199859.

[28] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui. Towards a semantic construction digital twin: Directions for future research. *Autom. Constr.*, 114, 2020. DOI: 10.1016/j.autcon.2020.103179.

[29] D. Bokde, S. Girase, and D. Mukhopadhyay. Matrix factorization model in collaborative filtering algorithms: A survey. volume 49, page 136 – 146, 2015.

[30] C. Bolchini, C.A. Curino, E. Quintarelli, F.A. Schreiber, and L. Tanca. Context information for knowledge reshaping. *Int. J. Web Eng. Technol.*, 5(1):88 – 103, 2009. DOI: 10.1504/IJWET.2009.025015.

[31] V.K. Bupesh Raja, K. Palanikumar, R. Rohith Renish, A.N. Ganesh Babu, Jashwanth Varma, and P. Gopal. Corrosion resistance of corten steel - a review. volume 46, page 3572 – 3577, 2020.

[32] F. Cao, G. Oskam, G. J. Meyer, and P. C. Searson. Electron transport in porous nanocrystalline tio2 photoelectrochemical cells. *J. Phys. Chem. A*, 100(42):17021–17027, 1996. DOI: 10.1021/jp9616573.

[33] C. Capodiferro, M. De Maria, M. Mazzei, M. Spreafico, O. V. Bik, A. L. Palma, and A. V. Solovyeva. Cultural itineraries generated by smart data on the web. *ISPRS Int. J. Geo-Inf.*, 13(2), 2024. DOI: 10.3390/ijgi13020047.

[34] M. Casillo, F. Colace, D. Conte, M. Lombardi, D. Santaniello, and C. Valentino. Context-aware recommender systems and cultural heritage: a survey. *J. Ambient Intell. Humaniz. Comput.*, 14(4):3109 – 3127, 2023. DOI: 10.1007/s12652-021-03438-9.

[35] M. Casillo, F. Colace, R. Gaeta, A. Lorusso, D. Santaniello, and C. Valentino. Revolutionizing cultural heritage preservation: an innovative iot-based framework for protecting historical buildings. *Evol. Intell.*, 17(5-6):3815 – 3831, 2024. DOI: 10.1007/s12065-024-00959-y.

[36] M. Casillo, F. Colace, A. Lorusso, D. Santaniello, and C. Valentino. Integrating physical and virtual experiences in cultural tourism: An adaptive multimodal recommender system. *IEEE Access*, 13:28353 – 28368, 2025. DOI: 10.1109/ACCESS.2025.3539205.

[37] M. Casillo, B. B. Gupta, M. Lombardi, A. Lorusso, D. Santaniello, and C. Valentino. Context aware recommender systems: A novel approach based on matrix factorization and contextual bias. *Electron. (Switz.)*, 11(7), 2022. DOI: 10.3390/electronics11071003.

[38] Mario Casillo, Massimo De Santo, Rosalba Mosca, and Domenico Santaniello. An ontology-based chatbot to enhance experiential learning in a cultural heritage scenario. *Front. Artif. Intell.*, 5, 2022. DOI: 10.3389/frai.2022.808281.

[39] L. Cecere, F. Colace, M. Lombardi, A. Lorusso, D. Santaniello, and C. Valentino. Cultural heritage enhancement through digital storytelling and context-aware recommender system. page 86 – 91, 2023.

[40] F. Cena, N. Mauro, L. Ardissono, C. Mattutino, A. Rapp, S. Cocomazzi, S. Brighenti, and R. Keller. Personalized tourist guide for people with autism. In *Adjunct Publication of the 28th ACM Conference*

on User Modeling, Adaptation and Personalization, page 347–351, 2020.

[41] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), 2011. DOI: 10.1145/1961189.1961199.

[42] C. Chen, L. Shen, F. Zou, and W. Liu. Towards practical adam: Non-convexity, convergence theory, and mini-batch acceleration. *J. Mach. Learn. Res.*, 23, 2022.

[43] M. Chen, S. Mao, and Y. Liu. Big data: A survey. *Mob. Netw. Appl.*, 19(2):171 – 209, 2014. DOI: 10.1007/s11036-013-0489-0.

[44] Z. Chen, W. Gan, J. Wu, K. Hu, and H. Lin. Data scarcity in recommendation systems: A survey. *ACM Trans. Recomm. Syst.*, 3(3), 2025. DOI: 10.1145/3639063.

[45] F. Colace, M. De Santo, L. Greco, V. Moscato, and A. Picariello. A collaborative user-centered framework for recommending items in online social networks. *Comput. Hum. Behav.*, 51:694 – 704, 2015. DOI: 10.1016/j.chb.2014.12.011.

[46] F. Colace, R. Gaeta, A. Lorusso, M. Pellegrino, and D. Santaniello. New ai challenges for cultural heritage protection: A general overview. *J. Cult. Herit.*, 75:168 – 193, 2025. DOI: 10.1016/j.culher.2025.07.019.

[47] D. Coscia, A. Ivagnes, N. Demo, and G. Rozza. Physics-informed neural networks for advanced modeling. *Journal of Open Source Software*, 8(87):5352, 2023. DOI: 10.21105/joss.05352.

[48] A. G. Cossatin, N. Mauro, F. Ferrero, and L. Ardissono. Tell me more: integrating llms in a cultural heritage website for advanced information exploration support. *Inf. Technol. Tour.*, 27(2):385 – 416, 2025. DOI: 10.1007/s40558-025-00312-8.

[49] D. R. E. Cotton, P. A. Cotton, and J. R. Shipway. Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innov. Educ. Teach. Int.*, 61(2):228 – 239, 2024. DOI: 10.1080/14703297.2023.2190148.

[50] E. J. Cross, S. J. Gibson, M. R. Jones, D. J. Pitchforth, S. Zhang, and T. J. Rogers. Physics-informed machine learning for structural health monitoring. *Procedia Struct. Integr.*, 21:347 – 367, 2022. DOI: 10.1007/978-3-030-81716-9_17.

[51] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *J. Sci. Comput.*, 92, 2022. DOI: 10.1007/s10915-022-01939-z.

[52] H. Dabiri, R. Marini, J. Clementi, P. Mazzanti, G. S. Mugnozza, F. Bozzano, and D. Bompa. Monitoring buildings performance using fea and ml based on the data acquired by insar; a case study of vittoriano building, rome. *Structures*, 74, 2025. DOI: 10.1016/j.istruc.2025.108643.

[53] A. Dahroug, A. Vlachidis, A. Liapis, A. Bikakis, M. López-Nores, O. Sacco, and J. J. Pazos-Arias. Using dates as contextual information for personalised cultural heritage experiences. *J. Inf. Sci.*, 47(1):82 – 100, 2021. DOI: 10.1177/0165551519871823.

[54] H. Dang, M. Tatipamula, and H. X. Nguyen. Cloud-based digital twinning for structural health monitoring using deep learning. *IEEE Trans. Ind. Inform.*, 18(6):3820 – 3830, 2022. DOI: 10.1109/TII.2021.3115119.

[55] M. De Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro. *Semantics-aware content-based recommender systems*. 2015.

[56] A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4 – 7, 2001. DOI: 10.1007/s007790170019.

[57] P. Di Francesco, P. Lago, and I. Malavolta. Architecting with microservices: A systematic mapping study. *J. Syst. Softw.*, 150:77 – 97, 2019. DOI: 10.1016/j.jss.2019.01.001.

[58] F. V. Difonzo, L. Lopez, and S. F. Pellegrino. Physics informed neural networks for an inverse problem in peridynamic models. *Eng. Comput.*, 2024. DOI: 10.1007/s00366-024-01957-5.

[59] M. Economou, H. Young, and E. Sosnowska. Digital storytelling for emotional engagement in museums: design and evaluation of the hunterian antonine wall emotive experiences. *Int. J. Herit. Stud.*, 2025. DOI: 10.1080/13527258.2025.2591613.

[60] H. Garg and M. Dave. Securing iot devices and securelyconnecting the dots using rest api and middleware. In *IoT-SIU 2019*, 2019.

[61] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.*, 79(11):1309 – 1331, 2009. DOI: 10.1002/nme.2579.

[62] G. H. Golub and C. F. Van Loan. Matrix computations, 4th. *Johns Hopkins*, 2013.

[63] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[64] S. Goswami, D. Nag, R. Sengupta, A. Bose, and S. Choudhury. A context-aware collaborative recommendation using knowledge graph. *SN Comput. Sci.*, 6(6), 2025. DOI: 10.1007/s42979-025-

04231-7.

[65] J. Grus. *Data science from scratch: first principles with python*. O'Reilly Media, 2019.

[66] M. Guo, T. Xu, J. Liu, Z. Liu, P. Jiang, T. Mu, S. Zhang, R. R. Martin, M. Cheng, and S. Hu. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media*, 8(3):331 – 368, 2022. DOI: 10.1007/s41095-022-0271-y.

[67] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He. A survey on knowledge graph-based recommender systems. *IEEE Trans. Knowl. Data. Eng.*, 34(8):3549 – 3568, 2022. DOI: 10.1109/TKDE.2020.3028705.

[68] Shuaishuai H., Haowei Y., You Y., Xueting L., and Yuming T. Deep adaptive interest network: Personalized recommendation with context-aware learning, 2024.

[69] J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan kaufmann, 2006.

[70] F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. *Building expert systems*. Addison-Wesley Longman Publishing Co., Inc., 1983.

[71] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Springer International Publishing AG, Cham, 1st ed. 2016 edition, 2015.

[72] C. F. Higham and D. J. Higham. Deep learning: An introduction for applied mathematicians. *SIAM Rev.*, 61(4):860–891, 2019. DOI: 10.1137/18M1165748.

[73] N.-D. Hoang. Image processing-based pitting corrosion detection using metaheuristic optimized multilevel image thresholding and machine-learning approaches. *Math. Probl. Eng.*, 2020, 2020. DOI: 10.1155/2020/6765274.

[74] M. Hong, S. An, R. Akerkar, D. Camacho, and J. J. Jung. Cross-cultural contextualisation for recommender systems. *J. Ambient Intell. Humaniz. Comput.*, 15(2):1659 – 1670, 2024. DOI: 10.1007/s12652-019-01479-9.

[75] J. Huang, J. Hu, Z. Li, Y. Zhang, and Y. Cheng. Physics-informed neural network for predicting the moisture diffusion and parameter inversion in stone heritage. volume 48, page 595 – 600, 2025.

[76] L. Hughes-Noehrer, J. Carlton, and C. Jay. Recommending art online: investigating user engagement and interactions with a digital collection. *Mus. Manag. Curatorsh.*, 2025. DOI: 10.1080/09647775.2025.2512316.

[77] V. Isakov. *Inverse Problems for Partial Differential Equations*. Appl. Math. Sci. (Switz.). Springer International Publishing, 2017.

[78] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. In *UAI 2018*, volume 2, page 876 – 885, 2018.

[79] P. Jackson. Introduction to expert systems. 1986.

[80] A. Jiménez Rios, V. Plevris, and M. Nogal. Bridge management through digital twin-based anomaly detection systems: A systematic review. *Front. Built Environ.*, 9, 2023. DOI: 10.3389/fbuil.2023.1176621.

[81] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nat. Rev. Phys.*, 3(6):422 – 440, 2021. DOI: 10.1038/s42254-021-00314-5.

[82] K. Kasemsarn and F. Nickpour. Digital storytelling in cultural and heritage tourism: A review of social media integration and youth engagement frameworks. *Heritage*, 8(6), 2025. DOI: 10.3390/heritage8060200.

[83] D. E. Klinesmith, R. H. McCuen, and P. Albrecht. Effect of environmental conditions on corrosion rates. *J. Mater. Civ. Eng.*, 19(2):121 – 129, 2007. DOI: 10.1061/(ASCE)0899-1561(2007)19:2(121).

[84] X. Kong and R. G. Hucks. Preserving our heritage: A photogrammetry-based digital twin framework for monitoring deteriorations of historic structures. *Autom. Constr.*, 152, 2023. DOI: 10.1016/j.autcon.2023.104928.

[85] M. Konstantakis, G. Alexandridis, and G. Caridakis. A personalized heritage-oriented recommender system based on extended cultural tourist typologies. *Big Data Cogn. Comput.*, 4(2):1 – 18, 2020. DOI: 10.3390/bdcc4020012.

[86] M. Konstantakis, Y. Christodoulou, J. Aliprantis, and G. Caridakis. Acux recommender: A mobile recommendation system for multi-profile cultural visitors based on visiting preferences classification. *Big Data Cogn. Comput.*, 6(4), 2022. DOI: 10.3390/bdcc6040144.

[87] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Comput.*, 42(8):30 – 37, 2009. DOI: 10.1109/MC.2009.263.

[88] A. Kusiak and M. Chen. Expert systems for planning and scheduling manufacturing systems. *Eur. J. Oper. Res.*, 34(2):113 – 130, 1988. DOI: 10.1016/0377-2217(88)90346-3.

[89] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural. Netw.*, 9(5):987 – 1000, 1998. DOI: 10.1109/72.712178.

[90] N. Laohaviraphap and T. Waroonkun. Integrating artificial intelligence and the internet of things in cultural heritage preservation: A systematic review of risk management and environmental monitoring strategies. *Buildings*, 14(12), 2024. DOI: 10.3390/buildings14123979.

[91] T. Le Bahers, T. Pauporté, P. P. Lainé, F. Labat, C. Adamo, and I. Ciofini. Modeling dye-sensitized solar cells: From theory to experiment. *J. Phys. Chem. Lett.*, 4(6):1044–1050, 2013. DOI: 10.1021/jz400046p.

[92] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436 – 444, 2015. DOI: 10.1038/nature14539.

[93] J. Li, Y. Ma, X. Zhan, and J. Pei. Research of contextual semantic reasoning model based on domain ontology. *Sci. Program.*, 2021, 2021. DOI: 10.1155/2021/4011190.

[94] Y. Li, Y. Du, M. Yang, J. Liang, H. Bai, R. Li, and A. Law. A review of the tools and techniques used in the digital preservation of architectural heritage within disaster cycles. *Herit. Sci.*, 11(1), 2023. DOI: 10.1186/s40494-023-01035-x.

[95] X. Liang, F. Liu, L. Wang, B. Zheng, and Y. Sun. Internet of cultural things: Current research, challenges and opportunities. *Comput. Mater. Contin.*, 74(1):469 – 488, 2023. DOI: 10.32604/cmc.2023.029641.

[96] P. Liu and L. Lan. Museum as multisensorial site: story co-making and the affective interrelationship between museum visitors, heritage space, and digital storytelling. *Mus. Manag. Curatorsh.*, 36(4):403 – 426, 2021. DOI: 10.1080/09647775.2021.1948905.

[97] W. Liu, P. P. Pokharel, and J. C. Principe. The kernel least-mean-square algorithm. *IEEE Trans. Signal Process.*, 56(2):543 – 554, 2008. DOI: 10.1109/TSP.2007.907881.

[98] E. Lucchi. Digital twins for the automation of the heritage construction sector. *Autom. Constr.*, 156, 2023. DOI: 10.1016/j.autcon.2023.105073.

[99] O. M. Machidon, M. Duguleana, and M. Carrozzino. Virtual humans in cultural heritage ict applications: A review. *J. Cult. Herit.*, 33:249 – 260, 2018. DOI: 10.1016/j.culher.2018.01.007.

[100] W. Mai, S. Soghrati, and R.G. Buchheit. A phase field model for simulating the pitting corrosion. *Corros. Sci.*, 110:157–166, 2016. DOI: 10.1016/j.corsci.2016.04.001.

[101] B. Maldon and N. Thamwattana. An analytical solution for charge carrier densities in dye-sensitized solar cells. *J. Photochem. Photobiol. A Chem.*, 370:41–50, 2019. DOI: 10.1016/j.jphotochem.2018.10.018.

[102] B. Maldon, N. Thamwattana, and M. Edwards. Exploring nonlinear diffusion equations for modelling dye-sensitized solar cells. *Entropy*, 22(2), 2020. DOI: 10.3390/e22020248.

[103] S. Mammeri, B. Barros, B. Conde-Carnero, and B. Riveiro. From traditional damage detection methods to physics-informed machine learning in bridges: A review. *Eng. Struct.*, 330, 2025. DOI: 10.1016/j.engstruct.2025.119862.

[104] K. Mani and A. K. B. Shenoy. Machine learning models in web applications: A comprehensive review. *ICT Express*, 2025. DOI: 10.1016/j.icte.2025.09.001.

[105] D. Massimo and F. Ricci. Building effective recommender systems for tourists. *AI Mag.*, 43(2):209 – 224, 2022. DOI: 10.1002/aaai.12057.

[106] P. Mateos and A. Bellogín. A systematic literature review of recent advances on context-aware recommender systems. *Artif. Intell. Rev.*, 58(1), 2025. DOI: 10.1007/s10462-024-10939-4.

[107] L. D. McClenny and U. M. Braga-Neto. Self-adaptive physics-informed neural networks. *J. Comput. Phys.*, 474, 2023. DOI: 10.1016/j.jcp.2022.111722.

[108] K. Michalakis and G. Caridakis. Context awareness in cultural heritage applications: A survey. *J. Comput. Cult. Herit.*, 15(2), 2022. DOI: 10.1145/3480953.

[109] T. Miller, I. Durlik, A. Łobodzińska, L. Dorobczyński, and R. Jasionowski. Ai in context: Harnessing domain knowledge for smarter machine learning. *Appl. Sci. (Switz.)*, 14(24), 2024. DOI: 10.3390/app142411612.

[110] M. Mishra and P. B. Lourenço. Artificial intelligence-assisted visual inspection for cultural heritage: State-of-the-art review. *J. Cult. Herit.*, 66:536 – 550, 2024. DOI: 10.1016/j.culher.2024.01.005.

[111] T. M. Mitchell. *Machine Learning*. 1997.

[112] M. Moreno, R. Ortiz, D. Cagigas-Muñiz, J. Becerra, J.M. Martin, A.J. Prieto, M.A. Garrido-Vizuete, J.M. Macías-Bernal, M.J. Chávez, and P. Ortiz. Art-risk 3.0 a fuzzy—based platform that combine gis and expert assessments for conservation strategies in cultural heritage. *J. Cult. Herit.*, 55:263 – 276, 2022. DOI: 10.1016/j.culher.2022.03.012.

[113] A. Motwani, P. K. Shukla, and M. Pawar. Ubiquitous and smart healthcare monitoring frameworks based on machine learning: A comprehensive review. *Artif. Intell. Med.*, 134, 2022. DOI: 10.1016/j.artmed.2022.102431.

[114] R. Mundlamuri, G. R. Gunnam, N. K. Mysari, and J. Pujuri. The evolution of ai: From classical

machine learning to modern large language models. *IEEE Access*, 13:178302 – 178341, 2025. DOI: 10.1109/ACCESS.2025.3621344.

[115] A. P.h Murdan, V. Bucktowar, V. Oree, and M. P. Enoch. Low-cost bus seating information technology system. *IET Intell. Transp. Syst.*, 14(10):1303 – 1310, 2020. DOI: 10.1049/iet-its.2019.0529.

[116] M. Murphy, E. McGovern, and S. Pavia. Historic building information modelling - adding intelligence to laser and image based surveys of european classical architecture. *ISPRS J. Photogramm. Remote Sens.*, 76:89 – 102, 2013. DOI: 10.1016/j.isprsjprs.2012.11.006.

[117] C. Musto, M. de Gemmis, P. Lops, F. Narducci, and G. Semeraro. *Semantics and Content-Based Recommendations*. 2022.

[118] C. A. Navarro, N. Hitschfeld-Kahler, and L. Mateu. A survey on parallel computing and its applications in data-parallel problems using gpu architectures. *Commun. Comput. Phys.*, 15(2):285 – 329, 2014. DOI: 10.4208/cicp.110113.010813a.

[119] D. Nawara and R. Kashef. Context-aware recommendation systems in the iot environment (iot-cars)-a comprehensive overview. *IEEE Access*, 9:144270 – 144284, 2021. DOI: 10.1109/ACCESS.2021.3122098.

[120] A. N. Nikolakopoulos, X. Ning, C. Desrosiers, and G. Karypis. *Trust Your Neighbors: A Comprehensive Survey of Neighborhood-Based Methods for Recommender Systems*. Springer, 2022.

[121] E. Not and D. Petrelli. Blending customisation, context-awareness and adaptivity for personalised tangible interaction in cultural heritage. *Int. J. Hum. Comput. Stud.*, 114:3 – 19, 2018. DOI: 10.1016/j.ijhcs.2018.01.001.

[122] B. O'Regan and M. Grätzel. A low-cost, high-efficiency solar cell based on dye-sensitized colloidal tio2 films. *Nature*, 353(6346):737–740, 1991. DOI: 10.1038/353737a0.

[123] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053 – 1058, 1972. DOI: 10.1145/361598.361623.

[124] G. Pavlidis. Recommender systems, cultural heritage applications, and the way forward. *J. Cult. Herit.*, 35:183 – 196, 2019. DOI: 10.1016/j.culher.2018.06.003.

[125] F.o Piccialli, P. Benedusi, L. Carratore, and G. Colecchia. An iot data analytics approach for cultural heritage. *Pers. Ubiquitous Comput.*, 24(3):429 – 436, 2020. DOI: 10.1007/s00779-019-01323-z.

[126] D. P. Pocobelli, J. Boehm, P. Bryan, J. Still, and J. Grau-Bové. Bim for heritage science: a review. *Herit. Sci.*, 6(1), 2018. DOI: 10.1186/s40494-018-0191-4.

[127] S. J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023.

[128] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *J. Comput. Phys.*, 477, 2023. DOI: 10.1016/j.jcp.2022.111902.

[129] O. I. Psomadaki, C. A. Dimoulas, G. M. Kalliris, and G. Paschalidis. Digital storytelling and audience engagement in cultural heritage management: A collaborative model based on the digital city of thessaloniki. *J. Cult. Herit.*, 36:12 – 22, 2019. DOI: 10.1016/j.culher.2018.07.016.

[130] A. Quarteroni, P. Gervasio, and F. Regazzoni. Combining physics-based and data-driven models: advancing the frontiers of research with scientific machine learning. *Math. Models Methods Appl. Sci.*, 35(4):905 – 1071, 2025. DOI: 10.1142/S0218202525500125.

[131] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. La Matematica per Il 3+2, 92. Springer International Publishing, Cham, 1st ed. 2016. edition, 2016.

[132] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.

[133] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, 1994.

[134] N. W. Rahayu, R. Ferdiana, and S. S. Kusumawardani. A systematic review of ontology use in e-learning recommender system. *Comput. Educ.: Artif. Intell.*, 3:100047, 2022. DOI: 10.1016/j.caeai.2022.100047.

[135] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686 – 707, 2019. DOI: 10.1016/j.jcp.2018.10.045.

[136] S. Rendle. *Item Recommendation from Implicit Feedback*. Springer, 2022.

[137] F. Ricci, B. Shapira, and L. Rokach. *Recommender systems: Introduction and challenges*. 2015.

[138] S. Rizvic, D. Boskovic, and B. Mijatovic. Advanced interactive digital storytelling in digital heritage applications. *Digit. Appl. Archaeol. Cult. Herit.*, 33, 2024. DOI: 10.1016/j.daach.2024.e00334.

[139] D. Roy and M. Dutta. A systematic review and research perspective on recommender systems. *J. Big Data*, 9(1):59, 2022. DOI: 10.1186/s40537-022-00592-5.

[140] G. Rozza, F. Ballarin, L. Scandurra, and F. Pichi. *Real Time Reduced Order Computational Mechanics: Parametric PDEs Worked Out Problems*, volume 5 of *SISSA Springer Series*. Springer Nature Switzerland, Cham, 2024.

[141] S. Russell and P. Norvig. Artificial intelligence: A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25(27):79–80, 1995.

[142] Philipp S., Dominik M., Laurin E., and Niklas K. Towards a problem-oriented domain adaptation framework for machine learning, 2025.

[143] A. Saibene, M. Assale, and M. Giltri. Expert systems: Definitions, advantages and issues in medical field applications. *Expert Syst. Appl.*, 177, 2021. DOI: 10.1016/j.eswa.2021.114900.

[144] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. 2008.

[145] L. Salau, M. Hamada, R. Prasad, M. Hassan, A. Mahendran, and Y. Watanobe. State-of-the-art survey on deep learning-based recommender systems for e-learning. *Appl. Sci.*, 12(23), 2022. DOI: 10.3390/app122311996.

[146] A.L. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 44(1):206 – 226, 2000. DOI: 10.1147/rd.441.0206.

[147] F. Santos, A. Almeida, C. Martins, R. Gonçalves, and J. Martins. Using poi functionality and accessibility levels for delivering personalized tourism recommendations. *Comput. Environ. Urban Syst.*, 77:101173, 2019. DOI: 10.1016/j.compenvurbsys.2017.08.007.

[148] O. M. Schei, A. Møgelvang, and K. Ludvigsen. Perceptions and use of ai chatbots among students in higher education: A scoping review of empirical studies. *Educ. Sci.*, 14(8), 2024. DOI: 10.3390/educsci14080922.

[149] S. Semeraro, F. Vecchi, R. Stasi, and U. Berardi. Physics-informed neural networks for predicting indoor temperature and cooling demand in historic buildings. *J. Build. Eng.*, 115, 2025. DOI: 10.1016/j.jobe.2025.114392.

[150] I. Serbouti, J. Chenal, S. A. Tazi, A. Baik, and M. Hakdaoui. Digital transformation in african heritage preservation: A digital twin framework for a sustainable bab al-mansour in meknes city, morocco. *Smart Cities*, 8(1), 2025. DOI: 10.3390/smartcities8010029.

[151] A. Shabani, M. Skamantzari, S. Tapinaki, A. Georgopoulos, V. Plevris, and M. Kioumarsi. 3d simulation models for developing digital twins of heritage structures: Challenges and strategies. *Procedia Struct. Integr.*, 37(C):314 – 320, 2021. DOI: 10.1016/j.prostr.2022.01.090.

[152] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim. Internet of things (iot) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5g-iot scenarios. *IEEE Access*, 8:23022 – 23040, 2020. DOI: 10.1109/ACCESS.2020.2970118.

[153] T. Shen and B. Li. Digital twins in additive manufacturing: a state-of-the-art review. *Int. J. Adv. Manuf. Technol.*, 131(1):63 – 92, 2024. DOI: 10.1007/s00170-024-13092-y.

[154] D. Shin, J.-W. Lee, J. Yeon, and S.-G. Lee. Context-aware recommendation by aggregating user context. page 423 – 430, 2009.

[155] E. Smith. *Introduction to the Tools of Scientific Computing*, volume 25. Springer Nature, 2022.

[156] D. Song, E. Yang, G. Guo, L. Shen, L. Jiang, and X. Wang. Multi-scenario and multi-task aware feature interaction for recommendation system. *ACM Trans. Knowl. Discov. Data*, 18(6), 2024. DOI: 10.1145/3651312.

[157] G. Sperlí. A cultural heritage framework using a deep learning based chatbot for supporting tourist journey. *Expert Syst. Appl.*, 183, 2021. DOI: 10.1016/j.eswa.2021.115277.

[158] M. Srifi, A. Oussous, A. A. Lahcen, and S. Mouline. Recommender systems based on collaborative filtering using review texts-a survey. *Information*, 11(6), 2020. DOI: 10.3390/INFO11060317.

[159] M. S. B. Storeide, S. George, A. Sole, and J. Y. Hardeberg. Standardization of digitized heritage: a review of implementations of 3d in cultural heritage. *Herit. Sci.*, 11(1), 2023. DOI: 10.1186/s40494-023-01079-z.

[160] Sybren Stüvel. Import bpy: Modern add-on development with blender. In *SIGGRAPH Labs '25*, 2025.

[161] T. Szandała, A. K. Bhoi, P. K. Mallick, C. Liu, and V. E. Balas. *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*, pages 203–224. Springer Singapore, Singapore, 2021.

[162] J. Terven, D. Córdova-Esparza, and J. Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.*, 5(4):1680 – 1716, 2023. DOI: 10.3390/make5040083.

[163] A. Thananchana, K. Noinan, and S. Wicha. The designing of cultural-based tourism recommendation system with community collaboration. page 510 – 513, 2022.

[164] S. A. Thorat, G. Ashwini, and M. Seema. Survey on collaborative and content-based recommendation

systems. page 1541 – 1548, 2023.

[165] G. Trichopoulos, M. Konstantakis, G. Alexandridis, and G. Caridakis. Large language models as recommendation systems in museums. *Electronics*, 12(18), 2023. DOI: 10.3390/electronics12183829.

[166] A. M. Turing. *Computing machinery and intelligence*. 2009.

[167] M. Unger, A. Tuzhilin, and A. Livne. Context-aware recommendations based on deep learning frameworks. *ACM Trans. Manage. Inf. Syst.*, 11(2):1–15, 2020. DOI: 10.1145/3386243.

[168] C. Valentino, G. Pagano, D. Conte, B. Paternoster, F. Colace, and M. Casillo. Step-by-step time discrete physics-informed neural networks with application to a sustainability pde model. *Math. Comput. Simul.*, 230:541 – 558, 2025. DOI: 10.1016/j.matcom.2024.10.043.

[169] I. Vasic, R. Quattrini, R. Pierdicca, A. Mancini, and B. Vasic. 3vr: Vice versa virtual reality algorithm to track and map user experience. *J. Comput. Cult. Herit.*, 17(3), 2024. DOI: 10.1145/3656346.

[170] N.a M. Villegas, C. Sánchez, J. Díaz-Cely, and G. Tamura. Characterizing context-aware recommender systems: A systematic literature review. *Knowl.-Based Syst.*, 140:173 – 200, 2018. DOI: 10.1016/j.knosys.2017.11.003.

[171] L. Wang, S. Joty, W. Gao, X. Zeng, and K.-F. Wong. Improving conversational recommender system via contextual and time-aware modeling with less domain-specific knowledge. *IEEE Trans. Knowl. Data Eng.*, 36(11):6447 – 6461, 2024. DOI: 10.1109/TKDE.2024.3397321.

[172] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen. A survey on large language model based autonomous agents. *Front. Comput. Sci.*, 18(6), 2024. DOI: 10.1007/s11704-024-40231-1.

[173] S. Wang, S. Sankaran, and P. Perdikaris. Respecting causality for training physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.*, 421, 2024. DOI: 10.1016/j.cma.2024.116813.

[174] S. Wang, Y. Teng, and P. Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.*, 43(5):3055 – 3081, 2021. DOI: 10.1137/20M1318043.

[175] S. Wang, H. Wang, and P. Perdikaris. Improved architectures and training algorithms for deep operator networks. *J. Sci. Comput.*, 92(2), 2022. DOI: 10.1007/s10915-022-01881-0.

[176] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207 – 244, 2009.

[177] C. Willberg, S. Duczek, J.M. Vivar-Perez, and Z.A.B. Ahmad. Simulation methods for guided wave-based structural health monitoring: A review. *Appl. Mech. Rev.*, 67(1), 2015. DOI: 10.1115/1.4029539.

[178] J. Wu, X. He, X. Wang, Q. Wang, W. Chen, J. Lian, and X. Xie. Graph convolution machine for context-aware recommender system. *Front. Comput. Sci.*, 16(6), 2022. DOI: 10.1007/s11704-021-0261-8.

[179] P. Xu, X. Zhu, and D. A. Clifton. Multimodal learning with transformers: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(10):12113 – 12132, 2023. DOI: 10.1109/TPAMI.2023.3275156.

[180] X. Xu, Z. Xu, P. Yu, and J. Wang. Enhancing user intent for recommendation systems via large language models. volume 13635, 2025.

[181] S. Yang. Application of collaborative filtering algorithm based collaboration and collaborative environment in the inheritance of traditional ceramic culture. page 566 – 571, 2024.

[182] X. Yang, P. Grussenmeyer, M. Koehl, H. Macher, A. Murtiyoso, and T. Landes. Review of built heritage modelling: Integration of hbim and other information techniques. *J. Cult. Herit.*, 46:350 – 360, 2020. DOI: 10.1016/j.culher.2020.05.008.

[183] Z. Yang, K. Li, C. Lin, L. R. Devereux, W. Zhang, C. Shao, J. M. Cole, and D. Cao. Predicting device parameters for dye-sensitized solar cells from electronic structure calculations to reproduce experiment. *ACS Appl. Ener. Mat.*, 3(5):4367–4376, 2020. DOI: 10.1021/acsaem.0c00060.

[184] J. Yoon and C. Choi. Real-time context-aware recommendation system for tourism. *Sensors*, 23(7), 2023. DOI: 10.3390/s23073679.

[185] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Comput. Methods Appl. Mech. Eng.*, 393, 2022. DOI: 10.1016/j.cma.2022.114823.

[186] L. Yuan, Yi-Q. Ni, Xiang-Y. Deng, and S. Hao. A-pinn: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *J. Comput. Phys.*, 462, 2022. DOI: 10.1016/j.jcp.2022.111260.

[187] M. H. Zafar, E. F. Langås, and F. Sanfilippo. Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review. *Robot. Comput.-Integr. Manuf.*, 89, 2024. DOI: 10.1016/j.rcim.2024.102769.

[188] Qi Zeng, Yash Kothari, Spencer H. Bryngelson, and Florian Schäfer. Competitive physics informed

networks. In *ICLR 2023*, 2023.

[189] Z. Zhang, A. Dang, J. Huang, and Y. Chen. Advancing conservation methods of the great wall cultural heritage through digital twin. *IEEE Internet Comput.*, 29(1):48 – 55, 2025. DOI: 10.1109/MIC.2025.3539367.

[190] J. Zhao, W. Wang, Z. Zhang, Q. Sun, H. Huo, L. Qu, and S. Zheng. Trusttf: A tensor factorization model using user trust and implicit feedback for context-aware recommender systems. *Knowl. Based Syst.*, 209:106434, 2020. DOI: 10.1016/j.knosys.2020.106434.

[191] Y. Zheng, R. Burke, and B. Mobasher. The role of emotions in context-aware recommendation. In *RecSys' Decisions 2013*, 10 2013.

[192] Y. Zheng, B. Mobasher, and R. Burke. Carskit: A java-based context-aware recommendation engine. page 1668 – 1671, 2016.

[193] Z. Zhou, G. Hu, C. Hu, C. Wen, and L. Chang. A survey of belief rule-base expert system. *IEEE Trans. Syst. Man Cybern.: Syst.*, 51(8):4944 – 4958, 2021. DOI: 10.1109/TSMC.2019.2944893.

[194] Z. Zou, Z. Wang, and G. E. Karniadakis. Learning and discovering multiple solutions using physics-informed neural networks with random initialization and deep ensemble. *arXiv preprint arXiv:2503.06320*, 2025. DOI: 10.48550/arXiv.2503.06320.