

# Explainable Spatio-Temporal GCNNs for Irregular Multivariate Time Series: Architecture and Application to ICU Patient Data

Óscar Escudero-Arnanz , *Student Member, IEEE*, Cristina Soguero-Ruiz ,  
and Antonio G. Marques , *Senior Member, IEEE*

**Abstract**—In this paper, we present XST-GCNN (eXplainable Spatio-Temporal Graph Convolutional Neural Network), an innovative architecture designed for processing heterogeneous and irregular Multivariate Time Series (MTS) data. Our processing architecture captures both temporal and feature dependencies within a unified spatio-temporal pipeline by leveraging a GCNN that uses a spatio-temporal graph and aims at optimizing predictive performance and explainability. For graph estimation, we propose several techniques, including a novel approach based on the (heterogeneous) Gower distance. Once the graphs are estimated, we propose two approaches for graph construction: one based on the Cartesian product that treats temporal instants homogeneously, and a spatio-temporal approach that considers different graphs per time step. Finally, we propose two GCNN architectures: a standard GCNN with a normalized adjacency matrix and a higher-order polynomial GCNN. In addition to predictive performance, we incorporate intrinsic explainability through architectural design choices, complemented by post hoc analysis using GNNExplainer, aimed at identifying key feature-time combinations that drive the model's predictions. We evaluate XST-GCNN using real-world Electronic Health Record data from the University Hospital of Fuenlabrada to predict Multidrug Resistance (MDR) in Intensive Care Unit patients, a critical healthcare challenge associated with high mortality and complex treatments. Our architecture outperforms traditional models, achieving a mean Receiver Operating Characteristic Area Under the Curve score of  $81.03 \pm 2.43$ . Additionally, the explainability analysis provides actionable insights into clinical factors driving MDR predictions, enhancing model transparency and trust. This work sets a new benchmark for addressing complex inference tasks with heterogeneous and irregular MTS, offering a versatile and interpretable solution for real-world applications.

**Index Terms**—Spatio-temporal graph convolution neural networks, heterogeneous data, irregular multivariate time series, graph learning, multidrug resistance, electronic health records.

## I. INTRODUCTION

GRAPHS have become powerful tools in both Machine Learning (ML) and Signal Processing (SP) due to their capacity to model complex interactions and capture intrinsic relationships within structured data [1]. In real-world applications, data often originates from multiple domains and exhibits heterogeneous characteristics, presenting a significant challenge for graph analysis and estimation [1], [2]. This heterogeneity spans various data types—including numerical, categorical, and textual, among others—collected at different temporal and spatial scales, further complicating traditional graph construction methods [2], [3]. These methods frequently rely on domain-specific adjustments for each variable or data source, leading to inconsistent graph representations and limiting their generalizability [2], [4].

To address the complexities described above, graph learning techniques have emerged as a powerful approach, allowing the inference of graph topologies directly from data, without imposing prior assumptions on the graph structure [5]. However, conventional methods that construct multiple graphs for different data characteristics introduce redundancy and computational inefficiencies [6]. This highlights the need for models capable of estimating a unified graph that integrates relationships among heterogeneous variables while maintaining computational efficiency and explainability [7], [8].

Once a unified graph that captures the relationships within heterogeneous data is estimated, it becomes foundational for subsequent inference and prediction tasks. Graph Convolutional Neural Networks (GCNNs) have proven highly effective in this context, leveraging the graph structure to capture hidden patterns by iterating over nodes and aggregating information from their neighbors [9]. More recent developments, such as Spatio-Temporal Graph Neural Networks (ST-GNNs), combine the strengths of GCNNs with Recurrent Neural Networks (RNNs) to handle both spatial and temporal dependencies in dynamic data [4], [10], [11], [12], [13]. These models have demonstrated significant potential in solving complex problems involving heterogeneous and dynamic datasets [10].

Received 1 November 2024; revised 1 June 2025; accepted 11 September 2025. Date of publication 24 September 2025; date of current version 7 October 2025. This work was supported in part by Spanish AEI (DOI 10.13039/501100011033) under Grant PID2022-136887NBI00 and Grant PID2023-149457OB-I00 and in part by the Autonomous Community of Madrid within the ELLIS Unit Madrid framework. The associate editor coordinating the review of this article and approving it for publication was Prof. Smita Krishnaswamy. (*Corresponding author: Óscar Escudero-Arnanz.*)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Research Ethics Committee of the University Hospital of Fuenlabrada under Application No. Ref. 24/22, EC2091.

The authors are with the Department of Signal Processing, King Juan Carlos University, 28933 Madrid, Spain (e-mail: oscar.escudero@urjc.es).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TSIPN.2025.3613951>, provided by the authors.

Digital Object Identifier 10.1109/TSIPN.2025.3613951

In response to the challenges of heterogeneous temporal data, we propose the eXplainable Spatio-Temporal Graph Convolutional Neural Network (XST-GCNN). This architecture is designed to efficiently capture and integrate Spatio-Temporal (ST) relationships within irregular Multivariate Time Series (MTS). Unlike conventional models, XST-GCNN unifies discrete and continuous data types into a cohesive graph representation, allowing for the modeling of complex, domain-spanning interactions. This approach not only estimates a single graph that encapsulates both spatial and temporal dependencies but also integrates explainability through architectural design and post hoc method, essential for clinical decision-making. By capturing both local and global relationships within medical data, XST-GCNN offers a robust solution for enhancing outcome prediction and supporting decisions in high-stakes environments such as healthcare.

The effectiveness of the XST-GCNN architecture is demonstrated through a case study on real-world medical data from the Intensive Care Unit (ICU) of the University Hospital of Fuenlabrada (UHF), addressing the critical issue of Multidrug Resistance (MDR). Recognized by the World Health Organization as a growing global threat, MDR complicates infection treatments, increases mortality rates, and imposes significant pressures on healthcare systems [14], [15]. It is important to highlight that MDR is an alarming subset of Antimicrobial Resistance (AMR). The dataset, derived from Electronic Health Records (EHRs) of the ICU-UHF, was modeled as irregular MTS and included heterogeneous features such as real-valued and discrete variables. EHRs are frequently used to address a range of healthcare challenges, including early detection of sepsis, prediction of patient deterioration, and personalized treatment planning [16]. The inherent irregularity and variability in EHR data, recorded at varying intervals across different patients, exacerbate the limitations of traditional Time Series (TS) models, which struggle to provide accurate and interpretable predictions in clinical settings [4]. Graph-based models, on the other hand, excel in representing the intricate relationships within clinical data as networks, facilitating more comprehensive analysis and better clinical insights [5].

By capturing detailed, interconnected relationships within biomedical data, graph-based models enhance the integration and analysis of critical healthcare information, such as patient histories, diagnoses, and treatments [5], [17]. When combined with advanced SP techniques, these models significantly improve the extraction of meaningful patterns from clinical data, boosting both explainability and predictive performance [18]. GCNNs, in particular, have revolutionized medical informatics by leveraging graph structures to reveal hidden patterns and improve the explainability of clinical data [19]. This is especially valuable for assessing patient conditions and predicting clinical outcomes. Despite their potential, current research in GCNNs often overlooks the unique challenges posed by irregular MTS, such as varying recording frequencies and the heterogeneous nature of clinical data.

The XST-GCNN architecture proposed in this paper directly addresses these challenges (heterogeneity, irregular MTS, spatial and temporal dependencies, and the need for explainability)

by integrating spatial and temporal dependencies within heterogeneous clinical data, offering a unified, interpretable architecture that enhances clinical decision-making in the context of MDR.

#### A. Related Work

The heterogeneity in real-world data has led to growing interest in architectures that support representation and learning in heterogeneous graphs [20], [21]. Various approaches address this heterogeneity from different perspectives: [21] explores the integration of natural language and code snippets, focusing on structural and semantic heterogeneity across mixed domains, while [20] models recommendation systems using heterogeneous graphs defined by multiple nodes and relation types. In [22], heterogeneity is examined as a combination of node and edge types within information networks, employing meta paths to interpret relational dependencies.

A comprehensive review of the state-of-the-art reveals that, while these studies provide valuable insights into managing diverse data types within heterogeneous graphs, most approaches are limited to independently estimating graphs for each data type—categorical, sequential, or numerical, among others [2], [21]. However, while some GNN frameworks support multiple data types, there is still no unified architecture for the combined integration of continuous and discrete data in a single model designed for classification or prediction tasks [2], [20], [22]. Graph estimation across varied data contexts, as well as developing representations that integrate multiple heterogeneous data types, remains an open area, especially in MTS analysis. Despite recent advancements, challenges persist in learning from heterogeneous graphs that integrate continuous and discrete data. This integration, compounded by temporal irregularity, introduces distinct challenges for inference and learning in graph-based models.

Further exploration of state-of-the-art methods for managing ST relationships in graph-based architectures reveals two dominant approaches: spectral-based and spatial-based methods [2], [23]. ST-GNNs can also be categorized by how they incorporate temporal variation—either through an auxiliary ML model specifically for temporal dependencies or by embedding time directly within the graph structure [23]. Hybrid ST-GNNs often combine spatial modules, such as spectral graph networks, spatial GNNs [11], or graph transformers, with temporal aspects captured by models like RNNs or transformers [23]. Another approach embeds temporal information within the GNN itself, representing time as a signal, axis, subgraph, or through layer-stacking techniques [23], [24]. Despite recent advancements, a significant gap remains in developing models that fully integrate temporal dimensions within the graph structure, allowing GCNNs to process time as an intrinsic part of graph topology and thereby simplifying architectural complexity [2].

After conducting a state-of-the-art review from a methodological perspective, in the clinical domain, traditional ML and Deep Learning (DL) models, such as Neural Networks (NNs), Gated Recurrent Units (GRUs), and transformers, have been widely used to address MDR prediction or simplifications thereof [25],

[26]. While these models often achieve high performance in predicting MDR, many approaches focus on short-term patterns or individual input instances within limited contexts, restricting their ability to capture the complex temporal dependencies crucial for comprehensive MDR prediction [27], [28]. Our previous work on MDR prediction in ICU settings focused on feature selection across independent time points [29], improving explainability yet constraining temporal dependency use [25], [30]. More recently, we implemented a GRU model with explainable artificial intelligence methods adapted for irregular MTS data [26], enhancing explainability but still lacking full integration of ST relationships.

In contrast, recent graph-based approaches have shown significant promise in capturing complex clinical interactions, particularly for AMR and MDR prediction. These methods leverage GNNs to model intricate relationships among clinical, microbiological, and environmental factors, enhancing prediction performance for MDR cases [31], [32]. For instance, GNNs applied to predict MDR infections in Enterobacteriaceae have demonstrated advantages over traditional models [32], while GCNNs have enhanced performance in antiviral drug prediction [31]. However, most studies remain focused on specific organisms or resistance mechanisms rather than providing a broader framework for MDR prediction across multiple pathogens with heterogeneous and irregular MTS data. These limitations underscore the need for models incorporating dynamic temporal dependencies essential in diverse clinical scenarios.

Building on these advancements, our proposed XST-GCNN architecture uniquely integrates ST graph analysis to capture both temporal dynamics and spatial relationships within clinical data. This approach is aligned with recent developments in the medical field, such as [33], which introduced an ST antibiogram predictor, and [34], which applied an ST-GNN for various healthcare applications.

## B. Contributions and Paper Outline

This section outlines our main contributions beyond the state-of-the-art, first methodologically and then in relation to MDR prediction. It also provides a roadmap for the architecture description and validation in the following sections. From the point of view of methodology, we introduce XST-GCNN, a graph-based DL architecture for irregular and heterogeneous MTS, with the following features:

- *Joint modeling of temporal and feature dependencies:* We propose an ST architecture that captures temporal and feature interactions within a unified architecture. The GCNN operates on a graph whose nodes are time-feature pairs, enhancing the representation of complex dependencies often overlooked in traditional methods.
- *Innovative graph estimation and GCNN design:* We explore several graph estimation techniques, including correlation-based methods, graph smoothness, and our novel Heterogeneous Gower Distance (HGD), designed for datasets with discrete and real-valued variables and compatible with Dynamic Time Warping (DTW). These approaches are used to construct two types of graph

representations: i) a Cartesian Product Graph (CPG), which preserves stable feature relationships over time, and ii) an ST Graph (STG), which adapts the feature-to-feature relationship for each time step.

- *Adaptive GCNN architecture for ST data:* At each layer, our GCNN, which supports both CPG and STG representations, can implement either a simple aggregation by processing the data with the normalized adjacency matrix or a more sophisticated higher-order polynomial filter able to deal with heterophilic data. This adaptability allows the model to generalize across diverse datasets, balancing complexity, predictive performance, and robustness.
- *Emphasis on explainability alongside performance:* In addition to high predictive performance, we prioritize explainability. Our model identifies key feature-time step combinations, providing insights into classification outcomes. Explainability stems from the model's transparent design, while explainability uses supplementary techniques to clarify feature impact. Together, these qualities aid clinical decision-making and contribute to building trust in AI-driven predictions.

From the point of view of applicability, we validate the architecture in the specific context of MDR prediction using ST data from EHRs, demonstrating its effectiveness in real-world clinical scenarios. Relevant contributions in this regard include:

- *Clinical decision support:* XST-GCNN enhances clinical decision-making by identifying feature-time step combinations that are essential for accurate MDR predictions, thereby improving model explainability and providing actionable insights for clinicians.
- *Superior predictive performance:* XST-GCNN outperforms traditional ML and DL approaches in classifying MDR in ICU patients, demonstrating its effectiveness in improving clinical outcomes.
- *Innovative application in MDR prediction:* This work represents a novel approach in healthcare analytics by applying graph-based methodologies specifically tailored for MDR prediction, highlighting the flexibility and robustness of the XST-GCNN architecture.

The remainder of this paper is organized as follows. Section II details the methods and architectures within the XST-GCNN architecture, Section III describes the case study and experimental validation, and Section IV concludes with key findings and future research directions.

## II. PROPOSED DATA PROCESSING ARCHITECTURE

This section introduces the proposed XST-GCNN architecture, specifically designed to address the challenges of irregular MTS and heterogeneous features, with a focus on EHR data. As shown in Fig. 1, the architecture is meticulously crafted to capture these complexities while enhancing explainability. We begin by defining essential notation, followed by an in-depth discussion of the graph learning and representation techniques employed. The section concludes with two specific GCNN designs tailored to process irregular MTS, yielding predictions that exploit the ST dependencies inherent in the dataset used.

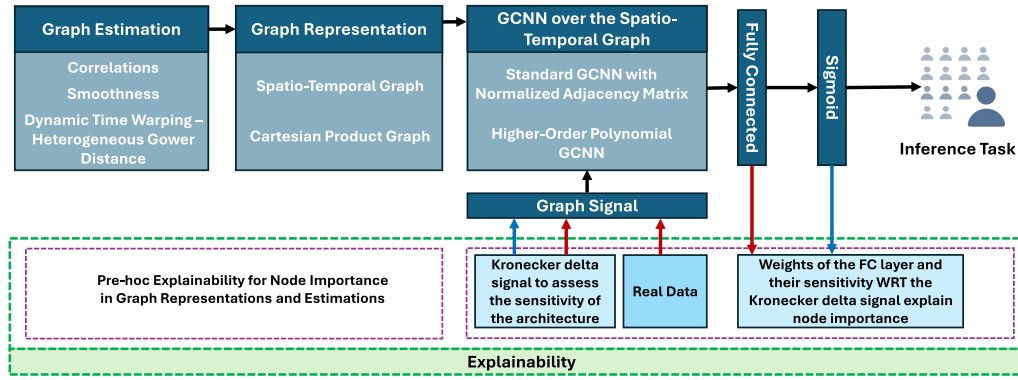


Fig. 1. Proposed XST-GCNN architecture for inference tasks using irregular MTS and heterogeneous features. The architecture employs relatively advanced SP techniques, including graph estimation based on correlations, smoothness constraints, and distance measures such as HGD and DTW. The graph representation is modeled as an STG or CPG, capturing both temporal and spatial dependencies. Two definitions for the GCNN layer are proposed: Standard GCNNs with Normalized Adjacency and Higher-Order Polynomial GCNNs. These layers are followed by LeakyReLU activation and dropout layers before passing through Fully Connected (FC) layers and a sigmoid activation for the final inference task. The architecture also emphasizes explainability, incorporating both pre-hoc and intrinsic methods. Pre-hoc explainability is achieved through node importance analysis during the graph representation and estimation phases. Intrinsic explainability is provided through analysis on both real and synthetic data during and after the architecture training. This includes the consideration of synthetic Kronecker delta signals to assess the sensitivity of the architecture with respect to each of the inputs. The combined approach put forth contributes to improved decision-making and a deeper understanding of the model’s behavior.

### A. Notation

We define our patient dataset as  $\mathcal{D} = \{(\mathbf{X}_p, y_p)\}_{p=1}^P$ , where  $P$  denotes the total number of patients. The  $p$ -th patient is characterized by a feature matrix  $\mathbf{X}_p \in \mathbb{R}^{F \times T}$ , with  $F$  representing the number of features and  $T$  the number of time steps. The  $t$ -th column of  $\mathbf{X}_p$ , denoted by  $[\mathbf{X}_p]_{(:,t)}$ , is a vector comprising the  $F$  features of patient  $p$  at time  $t$ . Conversely, the  $f$ -th row of  $\mathbf{X}_p$ , denoted by  $[\mathbf{X}_p]_{(f,:)}$ , represents the TS of feature  $f$  for the  $p$ -th patient across all  $T$  time steps. Notice that in most clinical applications, some of the features are real-valued while others are binary. This presents several challenges, including the selection of proper metrics to construct graphs, which will be discussed in more detail in subsequent sections. Moving on to the labels, patients are classified based on the presence of MDR pathogens during their ICU stay. Specifically, patients with at least one positive culture for MDR are assigned to the MDR class, while those without are classified as non-MDR. In this binary classification task,  $y_p = 1$  indicates that patient  $p$  developed MDR, and  $y_p = 0$  indicates a non-MDR patient. The model’s task is to predict these labels, with the predicted soft label for the  $p$ -th patient being denoted as  $\hat{y}_p \in [0, 1]$ .

The input MTS considered in this paper is heterogeneous and two-dimensional, which challenges traditional DL approaches. As explained in detail next, our approach is to build an ST graph that jointly models dependencies across features and time steps, and processes the information with a GCNN that leverages convolutions in the ST graph to integrate the information collected in the input MTS.

### B. Graph-Learning Methods

Having established the notation, we now delve into the fundamental concepts of (directed) weighted graphs and the methods employed to estimate these graphs, which are pivotal to our architecture. A weighted graph is represented as  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ , where  $\mathcal{V} = \{1, \dots, N\}$  denotes the set of nodes,  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

represents the set of edges, and  $\mathcal{W} : \mathcal{E} \rightarrow \mathbb{R}_+$  is a weight function assigning positive real values to each edge [35]. These weights indicate the strength or capacity of the connections between nodes.

Graphs can be categorized as either directed or undirected. In a directed graph, edges have a specific orientation, denoted by pairs  $(i, j) \in \mathcal{E}$ , indicating a connection from node  $i$  to node  $j$  [36]. Directed graphs distinguish between out-neighbors and in-neighbors, represented as  $\mathcal{N}_i^{\text{out}} = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$  and  $\mathcal{N}_i^{\text{in}} = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$ , respectively. Conversely, an undirected graph is a special case where each pair of nodes is connected in both directions, represented by a symmetric adjacency matrix [37]. In undirected graphs, the neighboring set of a node  $i$  is defined as  $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ . The graph is commonly represented by a weighted adjacency matrix  $\mathbf{A}$ , an  $N \times N$  matrix where  $[\mathbf{A}]_{ij} > 0$  indicates the weight of the edge  $(i, j) \in \mathcal{E}$  [35]. This matrix may be asymmetric for directed graphs, while it remains symmetric for undirected graphs. In symmetric graphs, another popular matrix is the graph Laplacian  $\mathbf{L}$ , which is defined as  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , where  $\mathbf{D}$  is the (diagonal) degree matrix that satisfies  $\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1})$ .

Within this paper the estimated graphs are weighted and, when considering time dependencies, directed. Two graph representation approaches are considered. In the first one, each node represents a particular feature-time tuple  $(f, t)$ , so that  $\mathcal{V} = \{1, \dots, F\} \times \{1, \dots, T\}$ . In the second one, each node represents a particular feature  $f$ , so that  $\mathcal{V} = \{1, \dots, F\}$ . Depending on the graph representation approach, graph estimation can be performed by either focusing on the information from a specific time step, denoted as  $\mathbf{X}'_t \in \mathbb{R}^{F \times P}$ , or considering the aggregated information from all patients over time, represented by the tensor  $\mathcal{X}' \in \mathbb{R}^{F \times T \times P}$ .

With these considerations in mind, the next step is to discuss the methods employed to estimate the weights associated with the edges of these graphs (see Fig. 1). The proposed methods

for graph computation include: a) correlation-based methods, b) graph smoothness-based methods, and c) HGD-DTW. Thus, the initial step involves a formal introduction to each of these methodologies, with adaptations specific to the MDR context. We differentiate between two distinct approaches for graph estimation: i) analyzing the entire temporal horizon as a unified entity, and ii) examining each temporal step as an independent unit. These methodologies facilitate the assessment of feature relationships by utilizing both aggregated and time-specific data, thereby enhancing the understanding of dynamic interactions within the dataset.

1) *Correlation-Based Methods*: A simple yet effective method to draw links between pairs of nodes is to quantify the level of correlation between the features associated with the nodes. Since we consider a heterogeneous setting where some of the features are binary and some are real-valued, we implement three distinct methods to capture the level of association: a) the Pearson correlation coefficient [38], used when both features are real-valued; b) the Matthews correlation coefficient (aka Phi coefficient [39]), used when both features are binary; and c) the Point-Biserial correlation coefficient [39], used when one of the variables is binary and the other is real-valued. Next, we briefly review each of these three methods. In the following, we assume that we focus on nodes  $i = 1$  and  $j = 2$ , with  $\mathbf{z}_1 = [z_1^{(1)}, z_1^{(2)}, \dots, z_1^{(K)}] \in \mathbb{R}^{1 \times K}$  and  $\mathbf{z}_2 = [z_2^{(1)}, z_2^{(2)}, \dots, z_2^{(K)}] \in \mathbb{R}^{1 \times K}$  denoting the two generic signals associated with those two nodes, and  $K$  representing a generic vector length.

- The *Pearson correlation coefficient* [38] quantifies the linear relationship between two numerical (real-valued) features  $\mathbf{z}_1$  and  $\mathbf{z}_2$ . With  $\bar{z}_1$  and  $\bar{z}_2$  representing their respective means over the  $K$  observations, the normalized Pearson correlation coefficient is simply given by

$$r_{pc}(\mathbf{z}_1, \mathbf{z}_2) = \frac{\sum_{k=1}^K (z_1^{(k)} - \bar{z}_1)(z_2^{(k)} - \bar{z}_2)}{\sqrt{\sum_{k=1}^K (z_1^{(k)} - \bar{z}_1)^2 \sum_{k=1}^K (z_2^{(k)} - \bar{z}_2)^2}}. \quad (1)$$

- The *Phi coefficient* assesses the level of association between two binary features  $\mathbf{z}_1$  and  $\mathbf{z}_2 \in \mathbb{R}^{1 \times K}$  [39]. Let  $n_{ij}(\mathbf{z}_1, \mathbf{z}_2)$  be the frequency of observations corresponding to each binary state combination of  $\mathbf{z}_1$  and  $\mathbf{z}_2$ . Specifically,  $n_{11}(\mathbf{z}_1, \mathbf{z}_2)$  and  $n_{00}(\mathbf{z}_1, \mathbf{z}_2)$  indicate the counts where both vectors simultaneously take the values “1” and “0”, respectively, while  $n_{10}(\mathbf{z}_1, \mathbf{z}_2)$  and  $n_{01}(\mathbf{z}_1, \mathbf{z}_2)$  capture the instances of mixed states. Furthermore, let  $n_1(\mathbf{z}_1)$  and  $n_0(\mathbf{z}_1)$  denote the total counts where the input vector (in this case  $\mathbf{z}_1$ ) is “1” or “0”. Then, the Phi coefficient of the pair  $(\mathbf{z}_1, \mathbf{z}_2)$  is

$$r_{\phi}(\mathbf{z}_1, \mathbf{z}_2) = \frac{n_{11}(\mathbf{z}_1, \mathbf{z}_2)n_{00}(\mathbf{z}_1, \mathbf{z}_2) - n_{10}(\mathbf{z}_1, \mathbf{z}_2)n_{01}(\mathbf{z}_1, \mathbf{z}_2)}{\sqrt{n_1(\mathbf{z}_1)n_0(\mathbf{z}_1)n_1(\mathbf{z}_2)n_0(\mathbf{z}_2)}}. \quad (2)$$

The numerator in (2) reflects the difference in the joint occurrences of concordant states, while the denominator

normalizes this difference by the product of the total occurrences, ensuring a scale-invariant measure of association.

- Finally, the *Point-Biserial coefficient* [39] is used when one feature (say  $\mathbf{z}_1$ ) is real-valued and the other one (say  $\mathbf{z}_2$ ) is binary. Let  $s_{z_1}$  be the standard deviation of the numerical feature  $\mathbf{z}_1$ ,  $\bar{z}_1^1(\mathbf{z}_1, \mathbf{z}_2)$  the mean of feature  $\mathbf{z}_1$  for the entries where  $\mathbf{z}_2$  is “1”, and  $\bar{z}_1^0(\mathbf{z}_1, \mathbf{z}_2)$  its counterpart for the entries where  $\mathbf{z}_2$  is “0”. Moreover, as in (2), the terms  $n_1(\mathbf{z}_2)$  and  $n_0(\mathbf{z}_2)$  denote the number of “1’s” and “0’s” in  $\mathbf{z}_2$ , respectively. Then, the Point-Biserial coefficient of the pair  $(\mathbf{z}_1, \mathbf{z}_2)$  is

$$r_{pb}(\mathbf{z}_1, \mathbf{z}_2) = \frac{\bar{z}_1^1(\mathbf{z}_1, \mathbf{z}_2) - \bar{z}_1^0(\mathbf{z}_1, \mathbf{z}_2)}{s_{z_1}} \sqrt{\frac{n_1(\mathbf{z}_2)n_0(\mathbf{z}_2)}{(n_1(\mathbf{z}_2) + n_0(\mathbf{z}_2))^2}}. \quad (3)$$

The next step is to use the coefficients in (1), (2), and (3) to build the graph. We consider first the approach where a different graph is learned for every  $t$ . To that end, we focus on  $\mathbf{X}'_t \in \mathbb{R}^{F \times P}$ , which is one slice of the tensor  $\mathcal{X}'$ . The first step is to remove (mask) the columns of  $\mathbf{X}'_t$  associated with patients that, for time step  $t$ , do not have information, giving rise to the matrix  $\mathbf{X}'_{\text{masked},t} = \mathbf{X}'' \in \mathbb{R}^{F \times P_t}$  with  $P_t \leq P$ . Since the rows of  $\mathbf{X}'_{\text{masked},t}$  represent features, we compute the adjacency of the feature-to-feature graph  $\mathbf{A}_t$  by: a) computing an  $F \times F$  matrix  $\mathbf{W}_t$  whose entry  $(f, f')$  is obtained using<sup>1</sup> (1)–(3) and b) setting to zero all the entries  $[\mathbf{W}_t]_{ij}$  such that  $|[\mathbf{W}_t]_{ij}| \leq \eta_t$ , with  $\eta_t$  being a pre-specified threshold. Finally, this procedure is repeated for  $t = 1, \dots, T$  giving rise to the set of graphs with adjacency matrices  $\{\mathbf{A}_t\}_{t=1}^T$ . The procedure for the case where a single graph is used to represent the full dataset is quite similar. To that end, we first rearrange the data in tensor  $\mathcal{X}' \in \mathbb{R}^{F \times T \times P}$  into the matrix  $\mathbf{X}'' \in \mathbb{R}^{F \times TP}$ . Then, we remove (mask) the columns of  $\mathbf{X}''$  associated with  $(t, p)$  pairs with missing information, giving rise to the matrix  $\mathbf{X}''_{\text{masked}} = \mathbf{X}'' \in \mathbb{R}^{F \times K}$  with  $K$  representing here the number of columns with data once the missing information was removed. After this, we create a single  $F \times F$  matrix  $\mathbf{W}$ , so that the  $(i, j)$ -th entry of  $\mathbf{W}$  is set to the Pearson/Phi/Point-biserial coefficient between the  $i$ -th and  $j$ -th rows of  $\mathbf{X}''_{\text{masked}}$  (see footnote II-B1). Finally, the entries of  $\mathbf{W}$  whose magnitude is below a threshold are set to zero, giving rise to the *static* adjacency matrix  $\mathbf{A}$ .

2) *Smoothness-Based Graph Estimation*: Smoothness-based graph estimation aims to learn graphs where the given signals are smooth [40], [41], [42]. While different ways to measure signal variability exist, the most popular in the graph SP literature is  $\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i,j} [\mathbf{A}]_{ij} (x_i - x_j)^2$  [43]. When a set of  $K$  graph signals is given, the latter can be written as  $\sum_{k=1}^K \mathbf{x}_k^T \mathbf{L} \mathbf{x}_k = \text{tr}(\sum_{k=1}^K \mathbf{x}_k \mathbf{x}_k^T \mathbf{L}) = K \text{tr}(\hat{\mathbf{C}} \mathbf{L})$ , with  $\hat{\mathbf{C}}$  denoting the sample covariance matrix. To ensure that all the features contribute the same, the sample covariance is typically normalized, so that smoothness-based estimation aims at learning a graph that minimizes  $\text{tr}(\hat{\mathbf{C}}_{\text{norm}} \mathbf{L})$ , where  $[\hat{\mathbf{C}}_{\text{norm}}]_{ij} = [\hat{\mathbf{C}}]_{ij} / \sqrt{([\hat{\mathbf{C}}]_{ii} [\hat{\mathbf{C}}]_{jj})}$ .

<sup>1</sup>The particular choice will depend on the nature of the  $(f, f')$  pair, using (1) if both are real-valued, (2) if both are binary, and (3) if they are mixed.

Among the different smoothness-based graph learning methods, we adopt the one in [42], which implements a greedy approach to learn the edges of the graph. More specifically, the method starts with an FC graph (i.e., a graph with  $F(F-1)/2$  edges), and removes the edge that reduces the graph signal variability the most. The process is repeated iteratively until a pre-specified value of edges (or smoothness) is reached.

As we discussed after (3), the next step is estimating the graph in two different setups. In the first one, the goal is to learn a graph for every time  $t$ . To that end, we use as graph signals the columns of  $\mathbf{X}'_{\text{masked},t} \in \mathbb{R}^{F \times P_t}$ , form the sample covariance matrix  $\hat{\mathbf{C}}_{\text{norm},t}$ , and then use that matrix to learn  $\mathbf{A}_t$  via [42]. In the second one, the goal is to learn a single graph  $\mathbf{A}$ . The graph signals in this case are the columns of matrix  $\mathbf{X}''_{\text{masked}} = \mathbf{X}'' \in \mathbb{R}^{F \times K}$ , which are used to learn the single matrix  $\mathbf{C}_{\text{norm}}$ .

3) *Heterogeneous Gower Distance - Dynamic Time Warping*: Another popular way to build graphs is using a distance function so that two nodes are connected if the distance between the signals (features) associated with those nodes is below a given threshold. Taking into account the particularities of our data, here we implement a distance-based graph learning method where we: i) use the HGD, which is an adaptation of the Gower distance [44] we propose and explain below, to measure the distance between heterogeneous features, and ii) when measuring distances between TS (i.e., when learning a single static graph), we combine HGD with DTW, a technique used to measure the dissimilarity between temporal sequences [45], [46]. A key feature in DTW is that the sequences may be misaligned, which is often the case in applications such as speech or EHR data.

First, let us consider two generic one-dimensional vectors  $\mathbf{z}_1 \in \mathbb{R}^K$  and  $\mathbf{z}_2 \in \mathbb{R}^K$ . To enhance consistency in the comparison of heterogeneous variables, we propose specific modifications to the HGD. When both  $\mathbf{z}_1$  and  $\mathbf{z}_2$  are continuous variables, normalization is achieved by defining the maximum values  $z_1^{\max} = \max\{\{z_1^{(k)}\}_{k=1}^K\}$ ,  $z_2^{\max} = \max\{\{z_2^{(k)}\}_{k=1}^K\}$  and  $z_{1,2}^{\max} = \max\{z_1^{\max}, z_2^{\max}\}$ , and then, with a slight abuse of notation, rescaling each variable as  $z_1^{(k)} = z_1^{(k)} \frac{z_{1,2}^{\max}}{z_1^{\max}}$  and  $z_2^{(k)} = z_2^{(k)} \frac{z_{1,2}^{\max}}{z_2^{\max}}$  for all  $k$ . In cases where one variable (say  $\mathbf{z}_1$ ) is binary, and the other one (say  $\mathbf{z}_2$ ) is continuous, the binary variable is rescaled by setting  $z_1^{(k)} = \max\{\{z_2^{(k)}\}_{k=1}^K\}$  if  $z_1^{(k)} = 1$  and  $z_1^{(k)} = \min\{\{z_2^{(k)}\}_{k=1}^K\}$  if  $z_1^{(k)} = 0$ , ensuring that both variables lie within the same range. Then, the HGD between those vectors is defined as

$$\delta_{\text{HGD}}(\mathbf{z}_1, \mathbf{z}_2) = \frac{1}{K} \sum_{k=1}^K \frac{|z_1^{(k)} - z_2^{(k)}|}{R_k}, \quad (4)$$

where  $R_k$  is the dynamic range of the  $k$ -th entry of the vector among all the vectors in the dataset [44]. Using this definition, each of the graphs in  $\{\mathbf{A}_t\}_{t=1}^T$  is learned by computing an  $F \times F$  matrix  $\tilde{\mathbf{W}}_t$  whose  $(i, j)$ -th entry is  $[\tilde{\mathbf{W}}_t]_{ij} = \delta_{\text{HGD}}([\mathbf{X}'_t]_{(i,:)}, [\mathbf{X}'_t]_{(j,:)})$ . After this, we apply an exponential transformation, so that the weights are found as

$$[\mathbf{W}_t]_{ij} = e^{-\beta[\tilde{\mathbf{W}}_t]_{ij}^2}, \quad (5)$$

with  $\beta$  being a temperature parameter used to tune the sensitivity of the graph with respect to the distance. Clearly, since the transformation in (5) is monotonically decreasing, smaller distances give rise to higher weights. Finally, a thresholding operator is applied entry-wise to set to zero edges with a small weight (i.e., nodes that are far apart from each other).

We move now to the setup where we use DTW to learn a single (static) graph. Specifically, let  $\tilde{\mathbf{X}}_f \in \mathbb{R}^{P \times T}$  be the slice of tensor  $\mathcal{X}'$  that contains the values of feature  $f$  for all patients and time steps. Clearly,  $\tilde{\mathbf{X}}_f$  can be understood as an MTS with  $P$  values per time step. The goal here is to use DTW to learn the feature-to-feature graph  $\mathbf{A}$ . Specifically, we build an  $F \times F$  distance matrix  $\tilde{\mathbf{W}}$  whose  $(i, j)$ -th entry is  $[\tilde{\mathbf{W}}]_{ij} = DTW_{\text{HGD}}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j)$ , with  $DTW_{\text{HGD}}$  representing the DTW distance computed using the HGD. To explain how this distance is computed, let us first define the cumulative distance matrix  $\mathbf{M} \in \mathbb{R}^{(T+1) \times (T+1)}$ , whose values are obtained using the following initialization and recursive procedure [47]:

$$[\mathbf{M}]_{1,1} = 0, [\mathbf{M}]_{1,t+1} = \infty, [\mathbf{M}]_{t+1,1} = \infty, \text{ for all } t \quad (6)$$

$$[\mathbf{M}]_{t,t'} = \delta_{\text{HGD}}([\tilde{\mathbf{X}}_i]_{(:,t)}, [\tilde{\mathbf{X}}_j]_{(:,t')}) + \min\{[\mathbf{M}]_{t-1,t'-1}, [\mathbf{M}]_{t-1,t'}, [\mathbf{M}]_{t,t'-1}\}. \quad (7)$$

After filling  $\mathbf{M}$  column by column (or row by row), the DTW distance is obtained as  $DTW_{\text{HGD}}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j) = [\mathbf{M}]_{T+1,T+1}$ ; see, e.g., [47] for additional details and motivation. While most implementations of DTW consider scalar TS and Euclidean distances, the distance in each step can be adapted for the dataset at hand (in this case, HGD). The main advantage of DTW over correlation and smoothness metrics in previous distances is that DTW is able to deal with misaligned data.

### C. Graph-Representation Approaches

The graph-learning methods detailed in the previous section captured relationships between features. The goal in this section is to explain how to leverage those results to deal with graphs able to capture both spatial (i.e., feature-to-feature) and temporal dynamics. The ultimate goal is to develop tractable graph-based representations that effectively capture the intrinsic relationships within irregular MTS data representation (see Fig. 1). To that end, we consider two distinct approaches: one that leverages the time varying graphs  $\{\mathbf{A}_t\}_{t=1}^T$  (labeled as STG), and another one that leverages the static graph  $\mathbf{A}$  (labeled as CPG).

*Spatio-Temporal Graph (STG)*: Our goal here is to describe a graph  $\mathcal{G}_{STG}$  whose nodes represent  $(f, t)$  tuples and, as a result, is represented by the  $FT \times FT$  adjacency matrix  $\mathbf{A}_{STG}$ . The first  $F$  columns (rows) index the features associated with the first time step, the second  $F$  columns (rows) the features associated with the second time step, and so forth. For the STG approach, we consider that: i) the relation among features changes over time, and the strength of this relationship is given by  $\{\mathbf{A}_t\}_{t=1}^T$ , and ii) the value of any feature at time step  $t$  is related to the value of the same feature at the previous time step  $t-1$ . This

results in the following adjacency matrix

$$\mathbf{A}_{STG} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{I} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{I} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{A}_T \end{bmatrix}, \quad (8)$$

where  $\mathbf{I}$  denotes the  $F \times F$  identity matrix that establishes temporal connections between features at consecutive time steps. Notice that the model in (8) is directed, since  $\mathbf{A}_{STG} \neq \mathbf{A}_{STG}^\top$ . If additional information on the time evolution of the MTS exists, e.g., by assuming that the MTS can be modeled as an autoregressive (AR) process of another one, matrix  $\mathbf{I}$  can be replaced with the AR transition matrix.

*Cartesian Product Graph (CPG):* A related representation approach can be implemented using the static feature-to-feature  $\mathbf{A}$  graph as input. As in the previous case, the goal is to build an  $FT \times FT$  adjacency matrix, labeled in this case as  $\mathbf{A}_{CPG}$ . For the CPG approach, we consider that: i) the relation among features does not change over time and the strength of this relationship is given by  $\mathbf{A}$ , and ii) the value of any feature at time step  $t$  is related to the value of the same feature at the previous time step. This results in a matrix  $\mathbf{A}_{CPG}$  that is obtained by replacing  $\mathbf{A}_t$  with  $\mathbf{A}$  for all  $t = 1, \dots, T$  in (8).

Interestingly, this construction is equivalent to saying that the graph  $\mathcal{G}_{STG}$  is obtained by computing the CPG between the static feature-to-feature graph and the *directed* path graph of  $\mathcal{G}_{dp}$  of length  $T$  [48]. To be more specific,  $\mathcal{G}_{dp}$  is a graph with  $T$  nodes whose adjacency matrix is given by  $[\mathbf{A}_{dp}]_{t,t+1} = 1$  for  $t = 1, \dots, T-1$  and zero for all other entries. Clearly,  $\mathcal{G}_{dp}$  is directed, has only  $T-1$  edges, and encodes the temporal progression. Using standard results of graph-theory, if two graphs are combined using the CPG, the resulting adjacency matrix can be obtained as

$$\mathbf{A}_{CPG} = \mathbf{A}_{dp} \oplus \mathbf{A}, \quad (9)$$

where  $\oplus$  represents the Kronecker sum. One advantage of the ST structure in (9) is that the *spectral* properties of  $\mathbf{A}_{CPG}$  follow directly from those of  $\mathbf{A}_{dp}$  and  $\mathbf{A}$  [48], facilitating the analysis and processing of signals defined over  $\mathbf{A}_{CPG}$ .

The graph-representations introduced in this section can be leveraged to model the data matrix  $\mathbf{X}_p$  (i.e., the information associated with patient  $p$ ) as a graph signal defined over either  $\mathcal{G}_{STG}$  or  $\mathcal{G}_{CPG}$ . Both approaches integrate temporal dynamics within the spatial graph structure. The selection between  $\mathbf{A}_{STG}$  and  $\mathbf{A}_{CPG}$  will depend on the specificities of the application. The STG approach is particularly advantageous when: i) the relations between features are complex and vary significantly over time, and ii) the number of samples (patients) for each time step is sufficiently high so that the time-varying graphs can be effectively estimated. In contrast, the CPG approach is more suitable when: i) the relations between features do not change too much over time, ii) data is limited, and iii) graph spectral tools are important to process the data at hand.

#### D. Graph Convolutional Neural Network

Upon constructing the ST graphs, the next step is to develop graph-based processing and learning architectures capable of incorporating both spatial and temporal dependencies. Considering the heterogeneity found in our input data—numerical and binary variables—as well as the success of NNs models in MDR prediction [25], [26], the strategy proposed in this paper is to create two GCNN architectures that take advantage of the ST graphs described in (8) and (9).

The numerical experiments in Section III will showcase that, by leveraging the ST relationships in the learned graphs, the architectures proposed next significantly enhance predictive performance and explainability, thereby facilitating more informed decision-making in clinical settings.

Succinctly, the goal of our architectures is to predict the output  $\hat{y}_p$  for the input  $F \times T$  matrix  $\mathbf{X}_p$ , which is the data associated with patient  $p$ . To that end, we first vectorize  $\mathbf{X}_p$  and then use  $\text{vec}(\mathbf{X}_p)$  as an input for a GCNN operating over a graph with  $FT$  nodes [cf. (8) and (9)]. Finally, we apply an FC layer to transform the output of the GCNN into the estimated label  $\hat{y}_p$ . Numerous GCNNs have been proposed in the literature [4], two of which are considered in this paper. The first one is the classical GCNN proposed in [49], which at every layer only considers linear averaging among one-hop neighbors. In contrast, the second architecture, at every layer, implements a polynomial graph filter with learnable coefficients [9], [50], enabling the mixing of information from multiple-hop neighborhoods and learning of low-pass/band-pass/high-pass frequency responses tailored to the properties of the dataset at hand. The remainder of this section is organized as follows. We first explain the two GCNNs considered, along with their main differences. Then, we describe our full DL architecture, which incorporates the previous GCNNs as the key processing block.

*GCNN-1: Standard GCNN with Normalized Adjacency Matrix:* This formulation leverages an architecture based on graph convolutional layers utilizing the normalized adjacency matrix. The core layer of this model normalizes the adjacency matrix  $\mathbf{A}$  by adding self-loops and incorporating the degree diagonal matrix  $\hat{\mathbf{D}} = \text{diag}((\mathbf{A} + \mathbf{I})\mathbf{1})$ , leading to

$$\hat{\mathbf{A}} = \hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\hat{\mathbf{D}}^{-\frac{1}{2}}, \quad (10)$$

where  $\mathbf{I}$  is the identity matrix. This normalization is critical for stabilizing the training process and enhancing the effectiveness of information propagation, as evidenced in prior studies on GCNNs. The graph convolution operation at each layer is expressed as

$$\mathbf{H}^{(l+1)} = \hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}, \quad (11)$$

where  $\mathbf{H}^{(l)} \in \mathbb{R}^{FT \times U_l}$  denotes the data matrix at layer  $l$ , and  $\mathbf{W}^{(l)} \in \mathbb{R}^{U_l \times U_{l+1}}$  is the trainable weight matrix. The number of rows in  $\mathbf{H}^{(l+1)}$  coincides with the number of nodes ( $FT$ , for the setup at hand). In contrast, the number of columns in  $\mathbf{H}^{(l+1)}$  represents the number of “synthetic” graph signals generated by layer  $l+1$ . With this in mind, the formalization of this architecture is given by

$$\mathbf{H}^{(1)} = \sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)} + \mathbf{1}\mathbf{b}^{(0)}), \quad (12a)$$

$$\mathbf{H}^{(l+1)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)} + \mathbf{1}\mathbf{b}^{(l)}), \quad \text{for } l = 1, \dots, L-1, \quad (12b)$$

where  $\mathbf{X} \in \mathbb{R}^{FT \times U_0}$  represents the  $U_0$  input graph signals (in our case,  $U_0 = 1$ );  $\sigma$  is a nonlinear scalar activation function applied entry-wise;  $\mathbf{1}$  is a column vector of all ones; and  $\mathbf{b}^{(l)} \in \mathbb{R}^{1 \times U_{l+1}}$  is the learnable bias vector. Overall, the learnable parameters are  $\{\mathbf{W}^{(l)} \in \mathbb{R}^{U_l \times U_{l+1}}\}_{l=0}^{L-1}$  and  $\{\mathbf{b}^{(l)} \in \mathbb{R}^{1 \times U_{l+1}}\}_{l=0}^{L-1}$ .

*GCNN-2: Higher-Order Polynomial GCNN:* While effective in many relevant applications, the architecture in (12) suffers from problems associated with oversmoothing and poor performance dealing with heterophilic datasets [51]. Motivated by this, we propose a second GCNN architecture, which, at every layer, implements (a bank of) polynomial graph filters with learnable coefficients [9], [50]. This formulation extends the standard GCNN by enabling: i) higher-order graph convolutions that linearly mix information from nodes that are multiple hops away and ii) learning generic frequency responses, which mitigates the problems associated with oversmoothing and endows the GCNN to be applied to non-homophilic datasets. Both generalizations open the door to a GCNN able to capture more complex relationships within the graph. The higher-order convolution operation is defined as [9]

$$\mathbf{H}^{(l+1)} = \sum_{k=0}^{K-1} \mathbf{A}^k \mathbf{H}^{(l)} \mathbf{W}_k^{(l)}, \quad (13)$$

with  $\mathbf{A}^k$  denoting the  $k$ -th power of  $\mathbf{A}$ . In contrast with (11), here we apply the adjacency matrix multiple times, which is an effective way to mix information within a  $(K-1)$ -hop neighborhood. Additionally, the number of learnable weights per convolution is  $K \times U_l \times U_{l+1}$ , endowing the architecture with additional degrees of freedom that can be used to learn more general graph-based transformations.

Upon replacing (13) into a GCNN with  $L$  layers, we have that

$$\mathbf{H}^{(1)} = \sigma \left( \sum_{k=0}^{K-1} \hat{\mathbf{A}}^k \mathbf{X} \mathbf{W}_k^{(0)} + \mathbf{1}\mathbf{b}^{(0)} \right), \quad (14a)$$

$$\mathbf{H}^{(l+1)} = \sigma \left( \sum_{k=0}^{K-1} \hat{\mathbf{A}}^k \mathbf{H}^{(l)} \mathbf{W}_k^{(l)} + \mathbf{1}\mathbf{b}^{(l)} \right), \quad \text{for } l = 1, \dots, L-1, \quad (14b)$$

where  $\mathbf{X} \in \mathbb{R}^{FT \times U_0}$  are the  $U_0$  input graph signals;  $\sigma$  denotes the nonlinear entry-wise activation function; and  $\mathbf{b}^{(l)} \in \mathbb{R}^{1 \times U_{l+1}}$  is the bias vector. Since the weight matrix is different for each  $k$ , this GCNN is able to assign positive or negative weights for the information of  $1, \dots, K-1$  neighbors. This contrasts with (11), which always assigns a positive weight to the information of 1-hop neighbors. In a nutshell, (14) enables the model to learn and leverage more sophisticated graph structures, thereby enhancing its capacity to model complex relationships.

The two alternative GCNN definitions presented here provide a way to adapt the ST DL model to the particularities and complexities of the data at hand. Each definition serves different scenarios, with GCNN-1 providing a more straightforward

approach suitable for general tasks, and GCNN-2 offering a more powerful framework for capturing intricate graph-based dependencies.

*Proposed ST graph-based DL architecture:* As already pointed out, our goal is to design a DL architecture for binary classification leveraging an ST graph whose links capture the strength of the relation between feature-time pairs. The (MTS) input to the architecture is the  $F \times T$  matrix  $\mathbf{X}_p$  and the (soft label) output is the scalar  $\hat{y}_p \in [0, 1]$ . The architecture (cf. Fig. 1) is composed of three blocks, which are applied sequentially.

- The first block simply vectorizes the input and replaces missing values with zero, giving rise to the  $FT$ -dimensional vector  $\mathbf{x}_p^{\text{zp}} = \text{zeropadd}(\text{vec}(\mathbf{X}_p))$ .
- The second block implements one of the two GCNNs presented in this section, with  $L$  denoting the number of layers. The GCNN architecture is specified as follows.
  - a) No pooling is implemented and, as a result, the output of all the GCNN layers (including the last one) can be interpreted as  $FT$ -dimensional signals defined over the ST graph.
  - b) The input of the first layer is the graph signal  $\mathbf{x}_p^{\text{zp}} \in \mathbb{R}^{FT \times 1}$ ; the layers  $l = 1, \dots, L-1$  generate as output multiple graph signals, with the number of generated signals per layer  $U_l$  being set to a constant  $U$  whose value is considered a hyperparameter; and the  $L$ -th layer outputs a single signal (i.e., we set  $U_L = 1$ ), so that the output of the GCNN  $\mathbf{h}^{(L)} = \mathbf{H}^{(L)} \in \mathbb{R}^{FT \times 1}$  is a one-dimensional graph signal. For the GCNN-1 architecture, the learnable parameters are the bias vectors  $\{\mathbf{b}^{(l)} \in \mathbb{R}^{1 \times U_l}\}_{l=0}^{L-1}$  along with the weight matrices  $\{\mathbf{W}^{(l)} \in \mathbb{R}^{U_l \times U_{l+1}}\}_{l=0}^{L-1}$ , with  $U_0 = U_L = 1$  and  $U_l = U$  otherwise. For the GCNN-2 architecture, the learnable parameters are the bias vectors  $\{\mathbf{b}^{(l)} \in \mathbb{R}^{1 \times U_l}\}_{l=0}^{L-1}$  along with the weights  $\{\mathbf{W}_k^{(l)} \in \mathbb{R}^{U_l \times U_{l+1}}\}_{l=0}^{L-1}$  for  $k = 0, \dots, K-1$ .
  - c) The activation function applied at each layer is set to a LeakyReLU, which is defined as  $\text{LeakyReLU}(h) = h$  if  $h > 0$ , and  $\alpha h$  if  $h \leq 0$ , where  $\alpha$  is a small positive scalar, typically set to  $\alpha = 0.01$ . This activation function is particularly suitable for GCNN-1, since it has been shown to mitigate the vanishing gradient problem by maintaining a non-zero gradient when  $h$  is negative, thereby enhancing gradient flow during backpropagation.
  - d) After applying the activation function, a dropout mechanism is introduced. Dropout operates by randomly deactivating a fraction  $\pi$  of the features for each node during training, effectively performing model averaging and preventing co-adaptation of feature representations. Mathematically, the operation performed by the Dropout layer can be expressed as  $[\mathbf{H}_{\text{input}}^{(l+1)}]_{i,u} = [\mathbf{H}^{(l)}]_{i,u} [\mathbf{R}_\pi]_{i,u}$ , where  $\mathbf{H}^{(l)}$  represents the output of the  $l$ -th layer,  $\mathbf{H}_{\text{input}}^{(l+1)}$  is the input to the  $(l+1)$ -th layer, and  $\mathbf{R}_\pi$  is a random binary matrix where each entry is independently drawn from a Bernoulli distribution with parameter  $\pi$ . Here,  $[\mathbf{R}_\pi]_{i,u} = 0$  indicates that the

feature  $u$  of node  $i$  is dropped out. Note that if no dropout is applied, we simply have  $\mathbf{H}_{\text{input}}^{(l+1)} = \mathbf{H}^{(l)}$ , as considered in (12) and (14).

- The third block implements an FC layer. Specifically, if  $\mathbf{h}^{(L)} \in \mathbb{R}^{FT}$  denotes the output of the last layer of the GCNN, this block estimates the soft output as

$$\hat{y}_p = \sigma(\mathbf{w}_o^\top \mathbf{h}^{(L)} + b_o) = \frac{1}{1 + e^{-(\mathbf{w}_o^\top \mathbf{h}^{(L)} + b_o)}}, \quad (15)$$

with  $\mathbf{w}_o \in \mathbb{R}^{FT}$  and  $b_o$  being learnable parameters, and the activation function corresponding to a sigmoid (binary softmax). The main reason to consider a simple FC layer is to enhance the explainability of the architecture. The entries of  $\mathbf{w}_o$  assign relevance to each element of  $\mathbf{h}^{(L)}$ , which corresponds to a specific feature–time pair. Larger positive weights indicate stronger association with the MDR class, while larger negative weights reflect relevance to the non-MDR class. This formulation not only facilitates the final binary decision but also provides a clear interpretation of how each feature–time pair contributes to the classification outcome.

The next section assesses the performance of our ST graph-based architectures in a real-world dataset. As explained in detail next, our novel integration of ST dynamics into a GCNN sets a new benchmark for predictive modeling in dynamic environments, particularly in the context of healthcare applications.

### III. RESULTS AND DISCUSSION

In this section, we apply the XST-GCNN architecture to a real-world dataset to classify MDR patients in the ICU setting at UHF. We first describe the dataset and outline the experimental setup, including parameter optimization. Next, we analyze graph properties and pre-hoc explainability, followed by an evaluation of XST-GCNN’s classification performance against state-of-the-art methods. Finally, we explore the model’s explainability, demonstrating how XST-GCNN clarifies the impact of each feature–time pair on classification outcomes, providing insights that support informed clinical decision-making.<sup>2</sup>

#### A. Dataset

This clinical case study uses the XST-GCNN architecture to predict MDR in ICU patients at UHF, aiming to detect the first MDR-positive culture within a 14-day window. The dataset spans 17 years, from January 2004 to February 2020, and includes the longitudinal clinical records of 3,502 ICU patients. Among these, 548 patients had at least one MDR-positive culture during their stay, highlighting a significant class imbalance.

A rigorous anonymization protocol was implemented to ensure patient confidentiality, with ethical approval obtained from the UHF Research Ethics Committee (ref: 24/22, EC2091).

<sup>2</sup>Due to space limitations, only a summary of the numerical results is included in the main manuscript. Additional results, including figures, tables, and graph representations, are available in the supplementary material submitted together with the main manuscript. Furthermore, we also provide additional results, analysis, and the code for the complete set of experiments in our GitHub repository <https://github.com/oscarescuderoarnanz/XST-GCNN>.

Building on this foundation, the primary objective is to solve a binary classification problem by predicting, based on data available within the first  $T = 14$  days, whether a patient will develop MDR. Due to variations in ICU stays—since not all patients remain hospitalized for the same number of days, nor do they develop MDR on the same day—MTS data exhibit irregularities that must be addressed in the analysis. The analysis is confined to ICU stays, excluding pre-admission data, to focus on the transmission dynamics of MDR pathogens within the ICU environment.

To achieve this, microbiological cultures and antibiograms were conducted to identify MDR pathogens, with particular attention to the first MDR-positive culture detected. Patients without MDR were labeled as  $y_p = 0$ , while those with a positive MDR culture within the first 14 days were labeled as  $y_p = 1$ . The 14-day window was chosen for its clinical relevance, aligning with standard infection control practices where the risk of transmission and the application of treatment protocols are most critical.

The dataset’s richness is underscored by the extensive set of  $F = 80$  variables collected for each patient, which are crucial for understanding the factors contributing to MDR development and the overall ICU environment. These variables are organized into three main categories, providing a comprehensive foundation for the analysis:

- *Patient-specific antibiotic therapy*: To monitor daily antibiotic therapy in the ICU for each patient, binary variables were created to indicate whether the patient received specific antibiotic families. These families include Aminoglycosides (AMG), Antifungals (ATF), Intestinal anti-infectives (ATI), Antimalarials (ATP), Carbapenems (CAR), 1st, 2nd, 3rd, and 4th generation Cephalosporins (CF1, CF2, CF3, CF4), Glycylines (GCC), Glycopeptides (GLI), Lincosamides (LIN), Lipopeptides (LIP), Macrolides (MAC), Monobactams (MON), Nitroimidazoles (NTI), unclassified antibiotics (*Others*), Oxazolidinones (OXA), Miscellaneous (OTR), Broad-spectrum Penicillins (PAP), Penicillins (PEN), Polypeptides (POL), Quinolones (QUI), Sulfonamides (SUL), and Tetracyclines (TTC). The variable *Others* denotes the administration of other antibiotics not included in this list.
- *ICU occupancy and co-patient treatments*: This group of variables captures essential environmental factors that reflect the overall health conditions and treatment protocols within the ICU. “Co-patients” are defined as the other patients sharing the ICU with the  $p$ -th patient during the same time interval, excluding the patient under study. The variables include continuous data on the number of co-patients receiving each of the 25 antibiotic families, represented as “family <sub>$n$</sub> ”. Additionally, daily ICU occupancy is documented through three main variables: i) the total number of co-patients (# of pat<sub>tot</sub>), ii) the number of co-patients diagnosed with MDR bacteria (# of pat<sub>MDR</sub>), and iii) the number of co-patients undergoing any form of antibiotic therapy (# of pat<sub>atb</sub>). These variables offer a detailed view of the ICU environment, providing insights

into the overall health status and treatment practices within the unit.

- *Patient health monitoring variables:* This category includes both continuous and binary variables that serve as key indicators of patient health. Continuous variables monitor daily hours of mechanical ventilation, tracheostomy duration, ulcer presence, hemodialysis hours, and the number and types of catheters used, including Peripherally Inserted Central Catheters (PICC), Central Venous Catheters (CVC), and specific insertion sites such as right (R), left (L), subclavian (S), femoral (F), and jugular (J). Additionally, we include the Nine Equivalents of Nursing Manpower Use Score (NEMS), a patient severity scale utilized by nursing staff. Binary variables capture whether the patient received insulin, artificial nutrition, sedation, muscle relaxation, or underwent postural changes. Additionally, organ failures are closely monitored to identify specific dysfunctions—hepatic, renal, coagulation, hemodynamic, and respiratory—with the administration of vasoactive drugs also being recorded. These variables offer critical insights into the patient’s health status and the necessary interventions during their ICU stay.

### B. Experimental Setup and Parameter Optimization

The experimental setup was designed to evaluate the predictive performance of the XST-GCNN architecture. The dataset was divided into a training set ( $\mathcal{D}_{\text{train}}$ ) with 70% of the patients and a test set ( $\mathcal{D}_{\text{test}}$ ) with the remaining 30%. Given the class imbalance—where MDR-positive cases were underrepresented—an undersampling approach was used within  $\mathcal{D}_{\text{train}}$  to balance the class distribution and reduce potential bias [52]. This method was chosen to maintain the integrity of the original data while minimizing the risk of overfitting. Combined with 5-fold cross-validation, this strategy enhanced the model’s generalization and reduced overfitting.

Hyperparameter tuning was performed to optimize the XST-GCNN architecture, aiming to minimize the binary cross-entropy loss. We explored key hyperparameters, including dropout rates {0.0, 0.15, 0.3}, learning rates {1e-3, 1e-2, 5e-2, 0.1}, and learning rate decay {0, 1e-5, 1e-4, 1e-3, 1e-2}. The number of units in the hidden layers ranged from {4, 8, 16, 32, 64}, and the network depth varied between 1 and 6 layers to find the most effective configuration. For the GCNN component within XST-GCNN, which uses polynomial filter banks, we evaluated the polynomial order  $K$  by testing values of 2 and 3, as these represent the definitions of the GCNN we assessed.

Model performance was assessed using four key metrics: sensitivity, specificity, Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) [53], and Area Under the Precision-Recall Curve (AUC-PR) [54]. Sensitivity quantifies the model’s ability to correctly identify MDR cases, whereas specificity measures its accuracy in recognizing non-MDR instances. ROC-AUC provides a global threshold-independent evaluation of the model’s discriminative capacity between the two classes. However, given the pronounced class imbalance in our dataset (roughly 85% non-MDR vs. 15% MDR), AUC-PR offers a

more informative and clinically meaningful metric, as it better reflects the model’s ability to identify early-positive cases—an aspect of particular relevance in critical care settings. All results presented in Section III-C were obtained on the held-out test set ( $\mathcal{D}_{\text{test}}$ ) and evaluated across these four metrics. To ensure the robustness and reliability of the findings, each experiment was independently repeated three times with different random splits of the training and test sets, thereby accounting for variability in data partitioning.

### C. Testing XST-GCNN in a Real-World Dataset

In this section, we assess some properties of the graph derived from our real dataset to understand feature interactions and data relationships. We then compare our approach with state-of-the-art methods and explore the explainability of the best-performing model using both synthetic and real signals.<sup>3</sup>

1) *Graph Properties and Pre-Hoc Explainability:* To understand in more detail the structure of the graphs generated by our XST-GCNN architecture, we analyzed two fundamental metrics in graph theory: edge density and edge entropy. Edge density,  $\eta(\mathcal{G})$ , quantifies graph connectivity by calculating the ratio of existing edges to the maximum possible number of edges [55]. For an undirected graph with  $|\mathcal{V}|$  vertices and  $|\mathcal{E}|$  edges, it is defined as  $\eta(\mathcal{G}) = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$ , where a value of 1 indicates a complete graph and 0 indicates a graph without edges. Edge entropy,  $H(\mathcal{G})$ , measures the complexity of the graph’s structure by assessing the distribution of edges, calculated as  $H(\mathcal{G}) = -\sum_{i=1}^{|\mathcal{V}|} d_i \ln d_i$ , where  $d_i$  represents the normalized weighted degree of the  $i$ -th node [55].

The analyses conducted, which evaluate edge density and edge entropy across various thresholds, demonstrate that selecting a threshold value of 0.975 effectively ensures meaningful graph sparsity while preserving structural integrity. This threshold maintains low edge density and stable entropy, facilitating the identification of key relationships without excessive connectivity. By assessing multiple threshold values, we confirmed that the model preserves its performance and representational capacity with the chosen threshold, validating its robustness and utility in processing heterogeneous and irregular MTS. Additional information regarding the method’s sensitivity to threshold selection is provided in the supplementary material (see Appendix A).

Additionally, the resultant graphs were provided to clinical experts of the UHF, including the head of the ICU, who validated the relevance and accuracy of the ST graph representations. This implies that, even before training the architecture, the selected representation has the potential to help clinical experts to visually assess evolving patterns, highlighting connections between different features across time, and identifying which variables gain or lose relevance throughout the timeline. Further details on these representations and pre-hoc explainability can be found in Appendix B of our supplementary material.

<sup>3</sup>Additional results, figures, and tables can be found in the supplementary material and in the following folder of the GitHub repository <https://github.com/oscarescuderoarnanz/XST-GCNN>

TABLE I

MEAN  $\pm$  STANDARD DEVIATION OF THE PERFORMANCE (ROC-AUC, SENSITIVITY, SPECIFICITY, AUC-PR) ON 5 TEST SETS WHEN CONSIDERING XST-GCNN ARCHITECTURE AND BASELINE MODELS FOR MDR VERSUS NON-MDR PATIENT CLASSIFICATION. THE HIGHEST AVERAGE PERFORMANCE FOR EACH FIGURE OF MERIT IS IN BOLD.

	Method	Performance Metrics			
		ROC-AUC	Sensitivity	Specificity	AUC-PR
Baselines	LSTM [56]	75.68 $\pm$ 1.23	67.92 $\pm$ 2.67	80.36 $\pm$ 2.22	46.49 $\pm$ 3.22
	GRU [57]	77.53 $\pm$ 1.42	66.67 $\pm$ 2.35	82.27 $\pm$ 0.79	47.17 $\pm$ 4.48
	Transformer [58]	78.28 $\pm$ 2.01	62.26 $\pm$ 6.72	84.96 $\pm$ 6.33	49.30 $\pm$ 3.70
	Mamba [59]	79.40 $\pm$ 3.58	64.15 $\pm$ 4.08	<b>87.77 <math>\pm</math> 0.16</b>	49.09 $\pm$ 3.61
	G-GRNN [111]	76.09 $\pm$ 2.35	74.84 $\pm$ 3.88	77.33 $\pm$ 3.50	51.31 $\pm$ 3.23
	GCNN-1 (correlations)	69.91 $\pm$ 2.00	62.89 $\pm$ 3.56	66.44 $\pm$ 7.30	35.54 $\pm$ 7.10
	GCNN-1 (smoothness)	72.70 $\pm$ 1.87	63.52 $\pm$ 2.35	70.15 $\pm$ 1.11	42.37 $\pm$ 2.21
	GCNN-1 (HGD-DTW)	74.53 $\pm$ 0.94	61.01 $\pm$ 2.35	74.30 $\pm$ 0.42	42.04 $\pm$ 0.98
	CPG with GCNN-1 (correlations)	76.74 $\pm$ 0.85	72.33 $\pm$ 3.21	72.39 $\pm$ 1.67	44.56 $\pm$ 2.48
	CPG with GCNN-1 (smoothness)	76.80 $\pm$ 0.81	74.84 $\pm$ 0.89	73.63 $\pm$ 0.97	46.21 $\pm$ 3.27
XST-GCNN (our)	CPG with GCNN-1 (HGD-DTW)	77.77 $\pm$ 1.71	75.47 $\pm$ 2.67	72.73 $\pm$ 3.24	46.70 $\pm$ 5.26
	STG with GCNN-1 (correlations)	76.44 $\pm$ 1.01	74.84 $\pm$ 3.88	71.94 $\pm$ 3.55	44.37 $\pm$ 1.25
	STG with GCNN-1 (smoothness)	75.94 $\pm$ 1.20	72.33 $\pm$ 0.89	74.41 $\pm$ 0.99	44.23 $\pm$ 1.89
	STG with GCNN-1 (HGD)	78.17 $\pm$ 1.04	76.10 $\pm$ 3.88	72.28 $\pm$ 2.22	49.08 $\pm$ 4.46
	CPG with GCNN-2 (correlations)	80.59 $\pm$ 4.79	72.33 $\pm$ 4.71	78.79 $\pm$ 8.55	54.43 $\pm$ 4.09
	CPG with GCNN-2 (smoothness)	79.94 $\pm$ 1.67	69.18 $\pm$ 5.41	80.13 $\pm$ 4.59	51.04 $\pm$ 4.46
	CPG with GCNN-2 (HGD-DTW)	76.95 $\pm$ 1.53	72.33 $\pm$ 2.35	72.95 $\pm$ 1.11	49.71 $\pm$ 3.16
	STG with GCNN-2 (correlations)	80.25 $\pm$ 4.07	<b>78.62 <math>\pm</math> 3.21</b>	74.30 $\pm$ 3.73	53.51 $\pm$ 3.34
	STG with GCNN-2 (smoothness)	75.59 $\pm$ 2.88	71.70 $\pm$ 1.54	73.74 $\pm$ 2.35	43.07 $\pm$ 1.75
	STG with GCNN-2 (HGD)	<b>81.03 <math>\pm</math> 2.43</b>	72.33 $\pm$ 2.35	78.68 $\pm$ 1.24	<b>54.98 <math>\pm</math> 2.82</b>

2) *Prediction Results*: The experimental analysis demonstrates the effectiveness of the proposed XST-GCNN architecture in classifying patients as MDR or non-MDR by modeling ST relationships and heterogeneous features in EHR data. Table I reports the results benchmarked against several baselines to assess the predictive performance of different approaches.

To contextualize the performance of XST-GCNN, we compare it with several sequence modeling baselines. These include four temporal architectures—Long Short-Term Memory (LSTM) [56], GRU [57], Transformer [58], and Mamba [59]—along with Gated-Graph RNN (G-GRNN) [111], and a standard GCNN with normalized adjacency matrix. The GCNN baselines were evaluated using the graph learning strategies detailed in Section II-B.

Among temporal models, Mamba achieved the highest ROC-AUC (79.40  $\pm$  3.58%) and specificity (87.77  $\pm$  0.16%), while matching Transformer in AUC-PR (approximately 49%). G-GRNN yielded the best AUC-PR (51.31  $\pm$  3.23%). In contrast, all static GCNN variants underperformed relative to temporal models. Even with the best distance metric (HGD-DTW), static GCNNs reached only 74.53  $\pm$  0.94% in ROC-AUC and 42.04  $\pm$  0.98% in AUC-PR, highlighting their limitations under severe class imbalance. Mamba offered the most balanced overall performance, while G-GRNN may be preferable in settings prioritizing sensitivity.

We then evaluated twelve variants of XST-GCNN, from two graph construction strategies—CPG and STG—combined with three edge-weighting criteria (correlation, smoothness, and HGD or HGD-DTW), and two convolutional backbones: a first-order graph convolution (GCNN-1) and a higher-order polynomial variant (GCNN-2). Within GCNN-1, CPG models performed best with HGD-DTW, reaching a ROC-AUC of 77.77  $\pm$  1.71% and the highest AUC-PR among all GCNN-1 variants (46.70  $\pm$  5.26%), along with a sensitivity of 75.47  $\pm$

2.67%. For STG, the HGD weighting also led to superior results, achieving 78.17  $\pm$  1.04% in ROC-AUC and 49.08  $\pm$  4.46% in AUC-PR, outperforming correlation- and smoothness-based STGs. These results highlight the benefit of temporally informed graph construction in clinical tasks involving irregular time series. Overall, STG with HGD emerged as the most robust GCNN-1 configuration. Nevertheless, all GCNN-1 models were outperformed by their GCNN-2 counterparts, demonstrating the advantage of higher-order spectral filtering in capturing complex ST dependencies.

Switching to GCNN-2 yielded substantial performance gains. For example, CPG with correlation-based edges reached a ROC-AUC of 80.59  $\pm$  4.79% and an AUC-PR of 54.43  $\pm$  4.09%, improving early-positive yield by nearly ten percentage points relative to GCNN-1, while maintaining balanced sensitivity (72.33  $\pm$  4.71%) and specificity (80.13  $\pm$  4.59%). Even stronger results were obtained with STG: using HGD weighting, this configuration achieved the highest ROC-AUC (81.03  $\pm$  2.43%) and AUC-PR (54.98  $\pm$  2.82%), with sensitivity and specificity of 72.33  $\pm$  2.35% and 78.68  $\pm$  1.24%, respectively. These results confirm the effectiveness of combining temporally-aware graph learning with higher-order spectral filtering to enhance discriminative power, particularly under class imbalance.

Therefore, based on the predictive performance analysis, the STG with GCNN-2 (HGD) configuration delivered the best overall performance, surpassing all baselines in ROC-AUC and AUC-PR. This variant attained the highest ROC-AUC (81.03  $\pm$  2.43%) and the highest AUC-PR (54.98  $\pm$  2.82%) across all models evaluated. Compared to Mamba, it improved ROC-AUC by 1.6 percentage points (pp), AUC-PR by 5.9 pp, and sensitivity by 8.2 pp, with a 9.1 pp decrease in specificity—an acceptable trade-off in clinical screening scenarios where minimizing false negatives is critical.

In addition, we complemented the predictive performance analysis with inference time measurements for representative model variants, as shown in Fig. 2 (three right-hand panels). While baseline models are faster due to their lower architectural complexity, the inference times observed for all models—including XST-GCNN—remain within a few seconds per patient. Although this reflects a moderate increase compared to the sub-second latency of simpler baselines, the additional cost is acceptable in practice and justified by the improvements in ROC-AUC and AUC-PR, alongside the clinical relevance of timely and reliable MDR detection.

In summary, the XST-GCNN model configured with STG and GCNN-2 (HGD) establishes a new benchmark for MDR prediction on irregular ICU TS. It outperforms all baselines on the most informative metrics and provides flexible configurations that balance sensitivity and specificity according to clinical needs. The consistent improvements in AUC-PR, despite the 85/15 class imbalance, highlight the value of jointly modeling spatial and temporal dependencies using high-order spectral filtering and HGD-based graph learning.

3) *Explainability*: As shown in the previous section, the proposed XST-GCNN delivers predictive performance for MDR classification, achieving high ROC-AUC and AUC-PR scores while effectively handling irregular MTS and heterogeneous

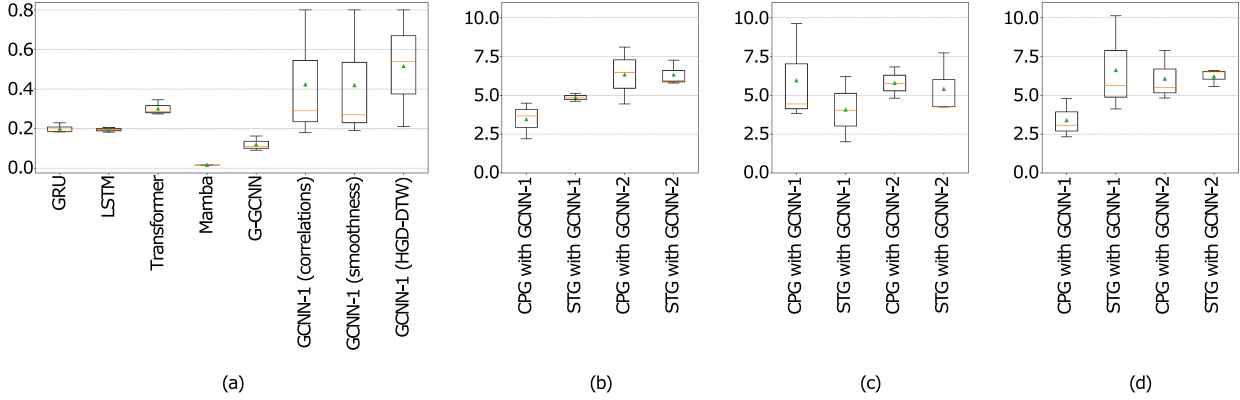


Fig. 2. Inference time comparison (in seconds) across all evaluated architectures. From left to right, the plot presents: (a) baseline architectures (GRU, LSTM, Transformer, Mamba, G-GRNN, and GCNN); (b) the proposed XST-GCNN variants using the correlations strategy; (c) smoothness; and (d) HGD-DTW. The boxplots show the distribution of inference times during the inference phase on the test set, providing a comparative view of the computational efficiency across architectures.

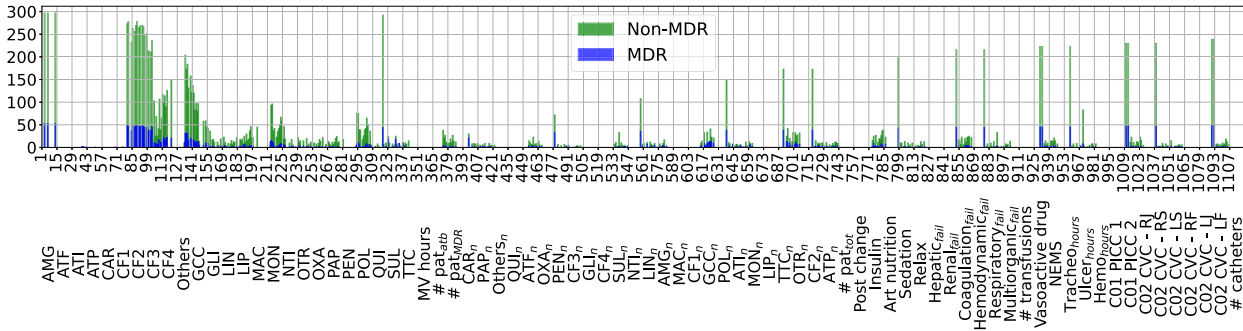


Fig. 3. Bar graph depicts the frequency distribution of variables associated with MDR (blue) and non-MDR (green) cases across a range of clinical features. The x-axis represents different clinical variables, while the y-axis indicates the frequency of occurrence for each variable. This visual comparison highlights the prevalence and variation of specific clinical features between MDR and non-MDR groups, providing insights into potential risk factors and patterns associated with MDR.

clinical variables. Beyond its predictive capacity, the architecture offers intrinsic explainability by identifying the most relevant feature–time interactions contributing to each class prediction. XST-GCNN was evaluated using both real-world EHR data and synthetic inputs, allowing a robust assessment of the model’s explainability across diverse conditions. To further validate these intrinsic explanations, we incorporated a widely adopted post-hoc method for graph-based models, GNNExplainer [60], enabling direct comparison with the intrinsic explanations produced by XST-GCNN. This evaluation was conducted on the  $D_{\text{test}}$  cohort using the best-performing configuration—STG estimated via HGD with GCNN-2—outlined in Section III-C2. The resulting patterns highlighted critical feature–time pairs associated with MDR risk and were reviewed by clinical experts, offering external validation of the explainability outcomes.

a) *Analysis with real patient data:* We start our analysis by trying to identify which  $(f, t)$  pairs are more relevant for classifying patients as either MDR or non-MDR. To that end, we: i) use the absolute value of the product of the input and the weight of FC layer [cf. (15)] as the importance value and ii) deem as relevant the 56 pairs with highest importance value, which represents the 5% of the  $FT = 1120$  pairs. In short, a) for each patient in the test set we compute the 1120 values

at the input of the FC layer and select the top 56  $(f, t)$  pairs; and b) we then repeat the experiment for each test patient and count the number of times each  $(f, t)$  pair is selected. The results are shown in Fig. 3. The x-axis represents the 1120  $(f, t)$  pairs, with the first 14 values being associated with the 14 measurements of feature AMG, the next 14 values with the 14 measurements of feature ATF, and so forth. The y-axis indicates the number of times each pair was selected in the top 5% across test samples, by class. High positive values strongly indicate MDR status, while negative weights correspond to non-MDR relevance, highlighting key variables influencing predictions and aiding clinical decision-making. The frequency distribution in Fig. 3 reveals distinct patterns and risk factors differentiating these patient groups. Additional results and analysis of Fig. 3 are available in the GitHub repository.<sup>4</sup>

The main analysis of the results is as follows. For non-MDR patients, the most relevant variables were concentrated in the initial 4 time steps, particularly within the first 24 hours and again after the first 72 hours of admission. These include specific

<sup>4</sup>The complete analysis can be found at [https://github.com/oscarescuderoarnanz/XST-GCNN/tree/main/XST-GNN\\_Architecture/step3\\_GCNNs](https://github.com/oscarescuderoarnanz/XST-GCNN/tree/main/XST-GNN_Architecture/step3_GCNNs), within each experiment’s folder.

antibiotic treatments and the number of co-patients receiving the same antibiotics, such as POL, OTR, TTC, LIP, GCC, CF2, LIP, ATP, and ATI. Additionally, during these first 24 hours, variables related to the patient's health status—such as hours on hemodialysis, tracheostomies, ulcers, and CO1 PICC 2, Co2 CVC - LJ and RF—emerged as significant. As time progressed from 24 to 72 hours, the total number of patients and those taking antibiotics, as well as NEMS, gained relevance, while after 72 hours post-admission, factors such as catheter types, patient status (NEMS), insulin, mechanical ventilation, respiratory failure, and the number of transfusions became increasingly important.

For patients with MDR infections, the 56 most relevant nodes were primarily identified within the initial 24 hours, with certain variables demonstrating importance across consecutive time points. Key variables in this critical period included the administration of antibiotic therapies and the number of co-patients receiving antibiotics from the same families, such as ATI, GCC, OTR, and TTC, with co-patient relevance observed for 21 out of 23 antibiotic families at the first time step, highlighting broader environmental impact. Additionally, patient health monitoring variables, including the number of catheters, CO2 CVC - RF, NEMS scores, hours on mechanical ventilation, and indications of hemodynamic, respiratory, and multi-organ failure, emerged as critical, correlating strongly with a more severe prognosis. The total number of MDR co-patients, overall ICU occupancy, and the total number of patients receiving antibiotics were also significant indicators of patient outcomes. Other important variables over time included co-patients receiving ATP and OTR in the last five time steps, CO2 CVC - LF from  $t_4$  to  $t_8$ , and CO2 CVC - RF from  $t_9$  to  $t_{13}$ , reflecting the increased complexity and severity commonly associated with MDR patients.

*b) Analysis with synthetic signals:* We analyze the model's sensitivity by evaluating the activation response to individual input nodes (feature-time steps) in our pre-trained architecture. More specifically, as illustrated in Fig. 1, we generate 1120 Kronecker delta inputs and, for each of them, evaluate the impact in each of the entries of the output of the GNN  $\mathbf{h}^{(L)}$  as well as in the decision made by the fully connected layer by computing the value of  $\mathbf{w}_o^\top \mathbf{h}^{(L)}$  in (15). This sensitivity analysis contributes to understanding the influence of each input value on the model's predictions, verifying that the relationships learned from real data are faithfully represented in the model's behavior.

The complete details and visualization of the results obtained can be accessed in our GitHub repository. Below, we summarize the main insights derived from analyzing the values  $\mathbf{w}_o^\top \mathbf{h}^{(L)}$ , focusing on three cases: (i) large values below zero, (ii) values above zero, and (iii) values close to zero. In case (i), large positive weights show higher relevance within the first 48 hours for variables such as C02 CVC - RJ and RS, insulin, vasoactive drugs, and multi-organ failure. We have information on antibiotics taken by co-patients, specifically AMG and ATF, as well as whether a patient receives antibiotics like ATP, LIN, LIP, OXA, Others, PEN, and QUI. After the first 48 hours of patient admission, antibiotics such as ATP, CF1, CF3, MAC, MON, NTI, POL, and QUI begin to gain greater importance. Health monitoring variables in this case include C02 CVC - LF, coagulation failure, and hemodynamic, hepatic, respiratory, and multi-organ conditions. Additionally, insulin administration,

postural changes, and vasoactive drugs remain relevant. In case (ii), large negative values initially highlight the first 48 hours, with notable relevance for information on antibiotics taken by co-patients, such as CAR, CF3, Others, PAP, and QUI, as well as the total number of patients and those receiving antibiotics. CF1 and OXA also contribute significantly. For health monitoring variables, transfusion count, hours with C02 CVC - LJ, and presence of postural changes emerge as key factors, alongside indicators of hemodynamic and hepatic failure. After 48 hours, the relevance shifts towards variables like antibiotics AMG, ATF, CAR, CF3, GLI, LIP, Others, PEN, QUI, and SUL concerning co-patients, among which CF2 is uniquely significant. Health status variables gain importance, including hours with C02 CVC - LF, and instances of coagulation, multi-organ, and respiratory failure, along with information on insulin, postural changes, relaxation, sedation, and vasoactive drugs. Lastly, case (iii) encompasses values close to zero, corresponding to many node-time steps with minimal impact on class determination, indicating a stable or low-relevance state across most temporal intervals. For further details and a comprehensive visualization of these findings, please refer to our GitHub repository: [https://github.com/oscarescuderoarnanz/XST-GCNN/tree/main/XST-GNN\\_Architecture/step3\\_GCNNs](https://github.com/oscarescuderoarnanz/XST-GCNN/tree/main/XST-GNN_Architecture/step3_GCNNs).

These analyses underscore the importance of managing specific interventions, especially those related to vascular access and antibiotic administration, to effectively manage patients at risk of developing MDR infections. These insights not only enhance predictive performance but also provide clinicians with a deeper understanding of the critical factors influencing patient outcomes, facilitating more informed and effective decision-making.

*c) GNNExplainer analysis:* GNNExplainer [60] is a model-agnostic post-hoc explainability method that estimates the relevance of input components by learning soft masks over node features and/or graph structures. For the targeted XST-GCNN configuration (STG estimated via HGD with GCNN-2), GNNExplainer was configured to generate node-level relevance scores for the binary classification task. Each node encodes a clinical feature at a specific time step, resulting in a matrix of dimension  $F \times T$  per patient. The explanations mask the relative contribution of each feature-time pair to the model's output. To produce robust class-wise explanations, GNNExplainer was applied to all test-set instances separately for MDR and non-MDR classes. The resulting relevance maps were then aggregated within each class to generate average heatmaps that highlight consistent predictive patterns across the population.

Fig. 4 presents the resulting explanation scores: panel (a) shows the average explanations for non-MDR patients, panel (b) for MDR patients, and panel (c) the point-wise difference between the two. In the non-MDR map, antibiotic therapy variables—especially ATI and CAR—show the highest importance scores, with ATF becoming more significant during the final time steps. Moderate relevance is noted for CF3, LIP, and PAP, which are also associated with antibiotic treatment. Furthermore, ICU occupancy and co-patient treatment variables (e.g., Others, SUL, CF1, TTC) demonstrate intermediate levels of activation, whereas health-monitoring parameters (e.g., insulin dosage, coagulation failure) exhibit negligible influence.

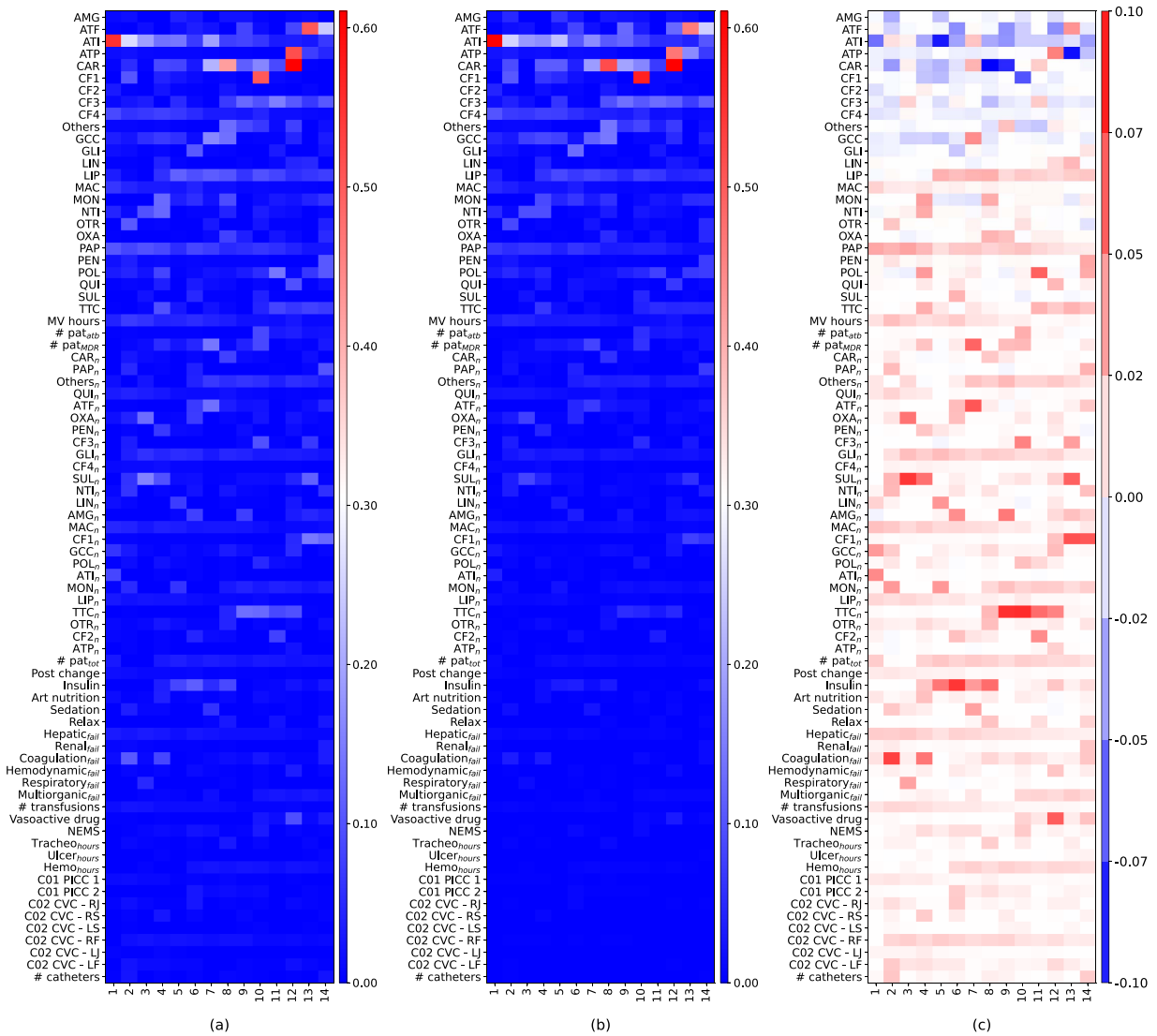


Fig. 4. Feature–time explanation scores obtained with GNNExplainer. Columns represent (left) non-MDR patients, (centre) MDR patients, and (right) their difference (*non-MDR* – *MDR*). Each heat-map cell shows the relative importance of a clinical variable (rows) at a specific time step (columns) for the final prediction.

For MDR patients, explanations remain centered on ATI, CAR, and ATF during the last three time steps, with moderate salience for CF3, GCC, and LIP. Other categories, such as patient monitoring and ICU occupancy, have negligible contributions. The differential heatmap highlights key discriminative features: antibiotics such as AMG, ATF, ATI, CAR, and ATP are more salient for MDR classification, whereas non-MDR cases are relatively more influenced by co-patient and monitoring variables. The sparse distribution of high-scoring regions suggests that only a subset of feature–time pairs is clinically decisive, supporting the model’s explainability and relevance. Notably, these results were reviewed and validated by the head of the ICU at UHF, who confirmed the clinical coherence of the patterns identified—particularly regarding antibiotic pressure—but also highlighted the absence of variables from other relevant categories such as ventilation status, transfusions, or organ failure. These missing patterns, expected from a clinical perspective, reflect a lower degree of granularity in the GNNExplainer output compared to the richer intrinsic attributions captured by XST-GCNN.

*d) Clinical relevance of explainability:* The intrinsic explainability of XST-GCNN proves both technically robust and clinically actionable. In non-MDR cases, it highlights early warning signals, including specific antibiotic regimens, co-treatment patterns, and markers of patient status such as catheterization and mechanical ventilation—key for timely intervention. In MDR cases, it identifies variables associated with cumulative antibiotic pressure, ICU occupancy, and clinical severity, such as sedation, postural adjustments, and multi-organ failure. Compared to the post-hoc GNNExplainer, which predominantly surfaces antibiotic-related features, XST-GCNN captures a more comprehensive set of clinically relevant factors. This includes overlooked but critical variables such as ventilation parameters, transfusions, and organ failure indicators. The difference in granularity was validated through expert review by the head of the ICU at UHF, who confirmed both the coherence of the patterns and the added value of the intrinsic explanations. Such detailed attribution supports more informed clinical decisions, enhancing the management and outcomes of MDR patients.

#### IV. CONCLUSIONS AND FUTURE WORK

In this work, we proposed a novel graph-based DL architecture specifically designed to process irregular and heterogeneous MTS data. Our approach jointly modeled dependencies between features and time through an ST architecture, where a GCNN operates on a graph that integrates both temporal and feature dimensions. A primary contribution was our innovative use of HGD for graph estimation, effectively modeling the complexities of heterogeneous data by accurately representing both categorical and real-valued features. Additionally, we explored and compared various methods for defining the GCNN and estimating the graph structure, evaluating their impact on model performance. Beyond predictive performance, we emphasized explainability by designing inherently interpretable architectures and complemented it with post-hoc analysis to clarify the model's decision process, conducting detailed analyses to illuminate the model's decision-making process, and facilitating more informed decisions.

We validated the XST-GCNN model through a real-world case study focused on predicting MDR in ICU patients by leveraging ST information embedded in EHR data. The proposed model delivered substantial improvements over conventional state-of-the-art machine learning and deep learning approaches, highlighting both its predictive capabilities and practical relevance for healthcare analytics. In particular, the XST-GCNN variant that combined STG graphs estimated via HGD with a higher-order polynomial GCNN achieved a ROC-AUC of  $81.03 \pm 2.43\%$ , sensitivity of  $72.33 \pm 2.35\%$ , specificity of  $78.68 \pm 1.24\%$ , and AUC-PR of  $54.98 \pm 2.82\%$ . These results surpassed those of the best-performing baseline, Mamba, which reached  $79.40 \pm 3.58\%$  (ROC-AUC),  $64.15 \pm 4.08\%$  (sensitivity),  $87.77 \pm 0.16\%$  (specificity), and  $49.09 \pm 3.61\%$  (AUC-PR). Compared to Mamba, XST-GCNN achieved a gain of 1.6 pp in ROC-AUC and nearly 6 points in AUC-PR, while offering a considerably more balanced trade-off between sensitivity and specificity. Inference-time evaluations further showed that, although XST-GCNN is approximately three times slower than the lightweight Mamba model, its latency remains well within acceptable bedside constraints and is more than compensated by the significant improvement in early-positive detection. Beyond enhancing binary classification performance, XST-GCNN also produced interpretable insights into the contributions of specific feature–time pairs, reinforcing its potential for clinical decision support and model transparency.

Further analysis of the estimated graphs and their clinical relevance confirmed that the architecture effectively captured critical patterns and variables essential for MDR prediction. The most relevant explainability findings obtained were related to the early administration of certain antibiotics, such as CAR, and the number of co-patients receiving similar treatments within the first 24 hours, which were highly predictive of MDR outcomes. Additionally, variables associated with organ failure, including decreased renal function and respiratory failure, were identified as key indicators. These patterns were consistently observed in both real-world ICU data and synthetic tests, highlighting the model's ability to detect meaningful clinical signals that are

essential for predicting MDR status. A complementary post-hoc analysis using GNNExplainer partially reproduced the core antibiotic-related patterns, albeit with reduced granularity. Its focus remained predominantly on antibiotic therapy variables, while it largely failed to capture clinically expected features from ICU occupancy and physiological monitoring domains.

Looking ahead, we will focus on extending XST-GCNN to domains where irregular MTS, heterogeneous inputs, and explainability are essential. While designed to be domain-agnostic, evaluating the model on tasks with more explicit spatial structure—such as fMRI analysis, melanoma progression prediction, or traffic flow modeling in urban networks—will allow us to assess its capacity to capture ST patterns across diverse contexts. These applications offer rich spatial signals and meaningful temporal dynamics, providing a valuable testbed for validating the generalizability and explainability of the proposed architecture. In addition, motivated by the explainability results, we plan to integrate more advanced mechanisms such as Graph Attention Networks [7] to enhance the model's ability to identify the most relevant relationships in the data. This extension aims to improve explainability in both clinical and non-clinical settings. We also intend to explore GNN-specific explainability techniques to obtain more fine-grained insights into graph-level decisions and better align the model's outputs with expert knowledge. Another important area of focus will be optimizing the computational efficiency of XST-GCNN, making it suitable for deployment in resource-constrained environments. This will increase its accessibility to a wider range of healthcare institutions, particularly those with limited computational resources. Finally, we plan to explore combining the current architecture with RNNs by replacing the FC layer with an RNN. This modification will further enhance the model's capacity for processing sequential data, allowing it to capture temporal dependencies in more complex datasets better.

In conclusion, XST-GCNN represents a significant advancement in the application of GNNs to clinical data, particularly for the prediction of MDR infections. By addressing the outlined challenges, including explainability and efficiency, and refining the approach, this research sets the stage for the development of more effective and reliable predictive models, with the potential to significantly impact patient care and clinical outcomes.

#### REFERENCES

- [1] X. Zhang and Q. Wang, "A graph-assisted framework for multiple graph learning," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 10, pp. 162–178, 2024.
- [2] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [3] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," 2020, *arXiv:2002.07962*.
- [4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
- [5] X.-M. Zhang, L. Liang, L. Liu, and M.-J. Tang, "Graph neural networks and their current applications in bioinformatics," *J. Biomed. Inform.*, vol. 12, 2021, Art. no. 690049.
- [6] E. Işufi, F. Gama, and A. Ribeiro, "EdgeNets: Edge varying graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7457–7473, Nov. 2022.

- [7] P. Velickovic et al., “Graph attention networks,” *Stat*, vol. 1050, no. 20, pp. 10–48550, 2017.
- [8] D. Bhaskar et al., “Inferring dynamic regulatory interaction graphs from time series data with perturbations,” in *Proc. Learn. Graphs Conf.*, 2024, pp. 22–1.
- [9] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, “Graphs, convolutions, and neural networks: From graph filters to graph neural networks,” *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, Nov. 2020.
- [10] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” 2018, *arXiv:1709.04875*.
- [11] L. Ruiz, F. Gama, and A. Ribeiro, “Gated graph recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 68, pp. 6303–6318, 2020.
- [12] E. Rossi et al., “Temporal graph networks for deep learning on dynamic graphs,” 2020, *arXiv:2006.10637*.
- [13] L. Yu, L. Sun, B. Du, and W. Lv, “Towards better dynamic graph learning: New architecture and unified library,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 67686–67700.
- [14] K. W. K. Tang, B. C. Millar, and J. E. Moore, “Antimicrobial resistance (AMR),” *Brit. J. Biomed. Sci.*, vol. 80, 2023, Art. no. 11387.
- [15] World Health Organization, “WHO bacterial priority pathogens list, 2024: Bacterial pathogens of public health importance to guide research, development and strategies to prevent and control antimicrobial resistance,” May 2024. Accessed: May 27, 2024. [Online]. Available: <https://www.who.int/publications/i/item/9789240093461>
- [16] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, “Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis,” *IEEE J. Biomed. Health Inform.*, vol. 22, no. 5, pp. 1589–1604, Sep. 2018.
- [17] A. Kallipolitis et al., “Medical knowledge extraction from graph-based modeling of electronic health records,” in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.*, 2023, pp. 279–290.
- [18] L. Murali, G. Gopakumar, D. M. Viswanathan, and P. Nedungadi, “Towards electronic health record-based medical knowledge graph construction, completion, and applications: A literature study,” *J. Biomed. Inform.*, vol. 143, 2023, Art. no. 104403.
- [19] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [20] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, and P. S. Yu, “A survey on heterogeneous graph embedding: Methods, techniques, applications and sources,” *IEEE Trans. Big Data*, vol. 9, no. 2, pp. 415–436, Apr. 2023.
- [21] H. Phan and A. Jannesari, “Heterogeneous graph neural networks for software effort estimation,” in *Proc. 16th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas.*, 2022, pp. 103–113.
- [22] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, and Q. Wang, “Interpretable and efficient heterogeneous graph convolutional network,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1637–1650, Feb. 2023.
- [23] Z. A. Sahili and M. Awad, “Spatio-temporal graph neural networks: A survey,” 2023, *arXiv:2301.10569*.
- [24] H. Liu et al., “TodyNet: Temporal dynamic graph neural network for multivariate time series classification,” *Inf. Sci.*, vol. 677, 2024, Art. no. 120914.
- [25] S. Martínez-Aguero et al., “Interpretable clinical time-series modeling with intelligent feature selection for early prediction of antimicrobial multidrug resistance,” *Future Gener. Comput. Syst.*, vol. 133, pp. 68–83, 2022.
- [26] Ó. Escudero-Arnanz, C. Soguero-Ruiz, J. Álvarez-Rodríguez, and A. G. Marques, “Explainable artificial intelligence techniques for irregular temporal classification of multidrug resistance acquisition in intensive care unit patients,” *IEEE Trans. Bio. Eng.*, 2025.
- [27] M. Tharmakulasingam, W. Wang, M. Kerby, R. L. Ragione, and A. Fernando, “TransAMR: An interpretable transformer model for accurate prediction of antimicrobial resistance using antibiotic administration data,” *IEEE Access*, vol. 11, pp. 75337–75350, 2023.
- [28] M. Nigo et al., “Deep learning model for personalized prediction of positive mrsa culture using time-series electronic health records,” *Nat. Commun.*, vol. 15, no. 1, 2024, Art. no. 2036.
- [29] Ó. Escudero-Arnanz et al., “Temporal feature selection for characterizing antimicrobial multidrug resistance in the intensive care unit,” in *Proc. Eur. Conf. Artif. Intell.*, 2020, pp. 54–59.
- [30] Ó. Escudero-Arnanz et al., “On the use of time series kernel and dimensionality reduction to identify the acquisition of antimicrobial multidrug resistance in the intensive care unit,” 2021, *arXiv:2107.10398*.
- [31] J. Pi, P. Jiao, Y. Zhang, and J. Li, “MDGNN: Microbial drug prediction based on heterogeneous multi-attention graph neural network,” *Front. Microbiol.*, vol. 13, 2022, Art. no. 819046.
- [32] R. Gouareb, A. Bornet, D. Proios, S. G. Pereira, and D. Teodoro, “Detection of patients at risk of multidrug-resistant enterobacteriaceae infection using graph neural networks: A retrospective study,” *Health Data Sci.*, vol. 3, 2023, Art. no. 0099.
- [33] X. Fu et al., “Spatial-temporal networks for antibiogram pattern prediction,” in *Proc. IEEE Int. Conf. Healthcare Inform.*, 2023, pp. 225–234.
- [34] A. Senthilkumar, M. Gupte, and S. Shridevi, “Dynamic spatial-temporal graph model for disease prediction,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, pp. 952–960, 2022.
- [35] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, “Graph filters for signal processing and machine learning on graphs,” *IEEE Trans. Signal Process.*, vol. 72, pp. 4745–4781, 2024.
- [36] D. B. West, *Introduction to Graph Theory*. Englewood Cliffs, NJ, USA: Prentice Hall, 2001.
- [37] R. Diestel, “Graph theory,” in *Reinhard Diestel (eBooks)*, Berlin, Germany: Springer, 2024.
- [38] I. Cohen et al., “Pearson correlation coefficient,” in *Noise Reduction Speech Process*. Berlin, Germany: Springer, 2009, pp. 1–4.
- [39] S. Malik and R. Singh, “A family of estimators of population mean using information on point bi-serial and phi correlation coefficient,” 2013, *arXiv:1302.1658*.
- [40] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.
- [41] V. Kalofolias, “How to learn a graph from smooth signals,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 920–929.
- [42] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, “Learning sparse graphs under smoothness prior,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 6508–6512.
- [43] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [44] J. Podani, “Extending Gower’s general coefficient of similarity to ordinal characters,” *Taxon*, vol. 48, no. 2, pp. 331–340, 1999.
- [45] M. Müller, “Dynamic time warping,” *Inf. Retrieval Music Motion*, vol. 2, pp. 69–84, 2007.
- [46] Ó. Escudero-Arnanz et al., “dtwParallel: A Python package to efficiently compute dynamic time warping between time series,” *SoftwareX*, vol. 22, 2023, Art. no. 101364.
- [47] S. Seto, W. Zhang, and Y. Zhou, “Multivariate time series classification using dynamic time warping template selection for human activity recognition,” in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2015, pp. 1399–1406.
- [48] A. Sandryhaila and J. M. F. Moura, “Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure,” *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, Sep. 2014.
- [49] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, *arXiv:1609.02907*.
- [50] V. N. Ioannidis, A. G. Marques, and G. B. Giannakis, “Tensor graph convolutional networks for multi-relational and robust learning,” *IEEE Trans. Signal Process.*, vol. 68, pp. 6535–6546, 2020.
- [51] Y. Yan, M. Hashemi, K. Swersky, Y. Yang, and D. Koutra, “Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks,” in *Proc. IEEE Int. Conf. Data Mining*, 2022, pp. 1287–1292.
- [52] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [53] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.
- [54] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.
- [55] E. D. Kolaczyk and G. Csárdi, *Statistical Analysis of Network Data With R*, vol. 65. Berlin, Germany: Springer, 2014.
- [56] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [57] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 2014, *arXiv:1412.3555*.
- [58] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
- [59] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” 2023, *arXiv:2312.00752*.
- [60] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, “GNNExplainer: Generating explanations for graph neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 9240–9251.