2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA) | 979-8-3503-4503-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/DSAA60987.2023.10302556

Leveraging patient similarities via graph neural networks to predict phenotypes from temporal data

1st **Dimitrios Proios**
University of Geneva
Geneva, Switzerland
dimitrios.proios@etu.unige.ch

2nd **Anthony Yazdani**
University of Geneva
Geneva, Switzerland
anthony.yazdani@etu.unige.ch

3rd **Alban Bornet**
University of Geneva
Geneva, Switzerland
alban.bornet@unige.ch

4th **Julien Ehrsam**
University of Geneva
& Geneva University Hospitals
Geneva, Switzerland
julien.ehrsam@hcuge.ch

5th **Islem Rekik**
Imperial College London
London, UK
i.rekik@imperial.ac.uk

6th **Douglas Teodoro**
University of Geneva
Geneva, Switzerland
douglas.teodoro@unige.ch

***Abstract*—Several machine learning approaches have been proposed to automatically derive clinical phenotypes from patient data. Nevertheless, methods leveraging similarity-based patient networks remain underexplored for temporal data. In this work, we propose a graph neural network (GNN) model that learns patient representation using different network configurations and feature modes. To explore the sequential nature of time series, features were extracted using a recurrent neural network (RNN) and embedded using information from the network structure via the GNN. Our method improves upon statistical and RNN baselines, with performance boosts up to 1% and 22% accuracy in the inductive and transductive settings, respectively. We also show that network configurations significantly impact performance in the transductive learning setting. Thus, automated phenotyping models based on GNNs could be used to support phenotype-based clinical research and ultimately for personalized clinical decision support.**

***Index Terms*—clinical automated phenotyping, time series, LSTM, similarity graph, graph neural networks**

***Data and Code Availability**:* This paper uses the MIMIC-III dataset [1], which is available on the PhysioNet repository [2]. The experiments are based on the public open source phenotyping benchmark of Harutyunyan *et al.* [3]. All our source code is publicly available at https://github.com/ds4dh/mimic3-benchmarks-GraDSCI23.

I. Introduction

The task of clinical automated phenotyping (automated phenotyping for simplicity) comprises the analysis of patient data to support diagnostic inference and clinical decision making [4]. In its supervised formulation, automated phenotyping refers to the task of associating a phenotype with a diagnosis, given a patient data representation [3]. In its unsupervised formulation, automated phenotyping is equivalent to a clustering task, in which patient data are used to define clinically relevant and homogeneous patient cohorts [5], [6]. Advances in this field can support the automation and characterization of diseases which directly impacts the capabilities of precise medicine [7], [8].

Hospitals host a wealth of data within Electronic Health Records (EHRs), providing rich resources for automated phenotyping. For example, the MIMIC database [2] contains the clinical data of 53,423 patients admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts, from 2008 to 2014. Intelligent algorithms can leverage these data to infer which diseases characterize a patient based on their phenotypical characteristics [9]–[12]. While EHRs are primarily maintained for patient care purposes, numerous studies have demonstrated their potential for secondary analysis in research applications [13]–[15]. However, the inherent challenges presented by the temporality and heterogeneity of such data can make their use cumbersome.

Data-driven methods based on deep learning have been proposed to learn clinical and non-clinical concept representations directly from raw patient data, enabling the association of patients with diverse phenotypes [16], [17], demonstrating promising results for cohort creation [18], [19] and supervised automated phenotyping [8]. In particular, Harutyunyan *et al.* [3] proposed a multi-label, multi-class benchmark to evaluate different approaches in automated phenotyping using patients' multivariate vital signs time series from the MIMIC-III dataset. Among the various deep neural network architectures proposed to solve this benchmark, recurrent architectures, such as Long Short Term Memory (LSTM) channel cells and memory Gated Recurrent Networks achieve the current state-of-the-art results [20], [21] (with reported Macro-AUROC scores of 0.775 and 0.779 accordingly). However, none of the aforementioned methods explicitly leverages associations among patients, which leaves a potential ground for improvement under-explored.

Graph Neural Networks (GNNs) provide a graceful architecture to model complex hierarchical data [11], [22]–[24], and several works [25], [26] have used GNNs to model patient networks. Nevertheless, the application of graph-based techniques and similarity network analyses to multivariate vital signs time series remains under-explored and may be a promising avenue

to improve performance in automated phenotyping. In this work, we present a novel approach for automated phenotyping based on GNNs that leverages patient network structures and temporal data. We formulate automated phenotyping as a node classification problem, which leverages patient similarity patterns learned by GNN models trained both in inductive and transductive settings. We propose several heuristics for patient network configuration and evaluate our approach using the MIMIC-III automated phenotyping benchmark created in [3]. The contribution of the work can be summarised as follows:

- We investigate different graph neural network architectures to predict complex phenotypes and show that transductive GNNs [27] outperform the current state-of-the-art in the assessed automated phenotyping benchmark.
- We propose different graph topologies to create patient networks from vital signs information, achieving stronger predictive performance both in inductive and transductive settings.
- We address the need for models that can capture rare phenotypes using transductive learning, which presents a trade-off between computational resources and sensitivity.

II. Related work

Research studies [3], [28] introduced relevant automated phenotyping benchmarks, that evaluate diagnosis prediction models based on the International Classification of Diseases (ICD) terminology [29]. Many disease definitions of clinical phenotype algorithms are based on expert rules [9], which can be laborious to develop and limited by the number of existing hypotheses.

Recent research has demonstrated that the use of deep learning models with EHR data for clinical risk prediction tasks, such as phenotyping, in-hospital mortality, length-of-stay, readmission, or decompensation prediction, shows significant improvements compared to traditional methods [30]. Moreover, it was shown that these models produce useful dense and continuous representations (embeddings) of patients and clinical concepts [31]. These models were widely applied to the analysis of healthcare data and are based on architectures such as Fully Connected Neural Networks (FCNNs) [32], Convolutional Neural Networks (CNNs) [33], Long-Short-Term Memory Networks (LSTMs) [34], Gated Recurrent Units (GRUs) [35], Word2Vec [10], [36], Transformers [37]–[39], and GNNs [40]–[44] (for systematic reviews, see [31], [45], [46]).

When addressing sequence-based patient representations, research focused on Recurrent Neural Networks (RNNs) [47] due to their ability to model time series data [48]. In particular, LSTMs have been successfully applied in various healthcare applications, such as predicting Intensive Care Unit (ICU) mortality [3], [49], [50], acute decompensation [51]–[53], hospital readmission rates [54], [55], and automated phenotyping [3], [20], [21]. Despite their success, these models have limitations in capturing the complex relationships between patients and their clinical profiles.

To address these limitations, recurrent networks were combined with graph-based methods for the problem of automated phenotyping. In prior works, patient networks were created using spectral graph signal processing [56], [57] or random walk algorithms [58]–[60]. The efficacy of training graphs with data-driven strategies was demonstrated in classic community detection tasks, particularly within the context of disease and comorbidity networks [61]. In recent works, the temporal information available in EHRs is combined with the graph topological information [25], [62]. Roucheteau *et al.* [25] trained a model that combines LSTMs and GCNs using demographics and vital signs in a comorbidity network. Unlike their approach, in our model graph edges do not rely on comorbidities, as these are part of the truth label in the task of automated phenotyping.

The performance of any GNN algorithm is recognized to be significantly influenced by the structure of its network connectivity [63], [64]. To that end, we applied data-driven strategies based on different hypotheses and evaluated the resulting graphs on their capacity to train graph neural network models. In the MiME study [26], the authors initiated their process from a fully connected graph to recover the hidden connectivity of the patient graph, refining its sparsity iteratively using self-attention [65]. However, this approach would be computationally intractable in our context, as the fully connected version of our graph demands extensive memory resources and presents scalability challenges for larger data sets. Instead, we explored some heuristics to define the structure of the graph, combining expert rules and data-driven insights. More details are provided in Subsection III-C2.

One distinct advantage of graph-based techniques is their ability to perform inductive and transductive learning [66]. In the inductive learning context, the test set is used neither in the training nor in the hyper-parameter-tuning phases. On the contrary, in the transductive approach, test node features (i.e., not true labels) of the graph are considered during the training phase, avoiding the need to solve a more difficult and generic problem [67]. The major disadvantage of the transductive setting is the need to retrain the model for each new data instance. Leveraging both approaches, we compare the performance of GNNs in different network configurations in the task of automated phenotyping.

III. Methods

A. Study design and dataset description

In this study, we embedded multivariate patient vital signs from the MIMIC-III dataset (see Table I), comparing their statistical profiles, as well as LSTM and GNN layers, whose output was used to perform diagnosis classification. The data consists of N time series $X_i = \{x_1, x_2, ..., x_{T_i}\}$, where each instance $x_t \in \mathbb{R}^{17}$ contains the measurements of 17 vital signs and $T_i \in \mathbb{N}$ is the number of measurements in the i-th time series. N is the number of episodes recorded in the benchmark of [3] (each patient having 1 or more episodes). We associated each episode to a 25-dimensional multi-hot vector that denotes the true labels of any patient time series $y_i \in \{0, 1\}^{25}$, where

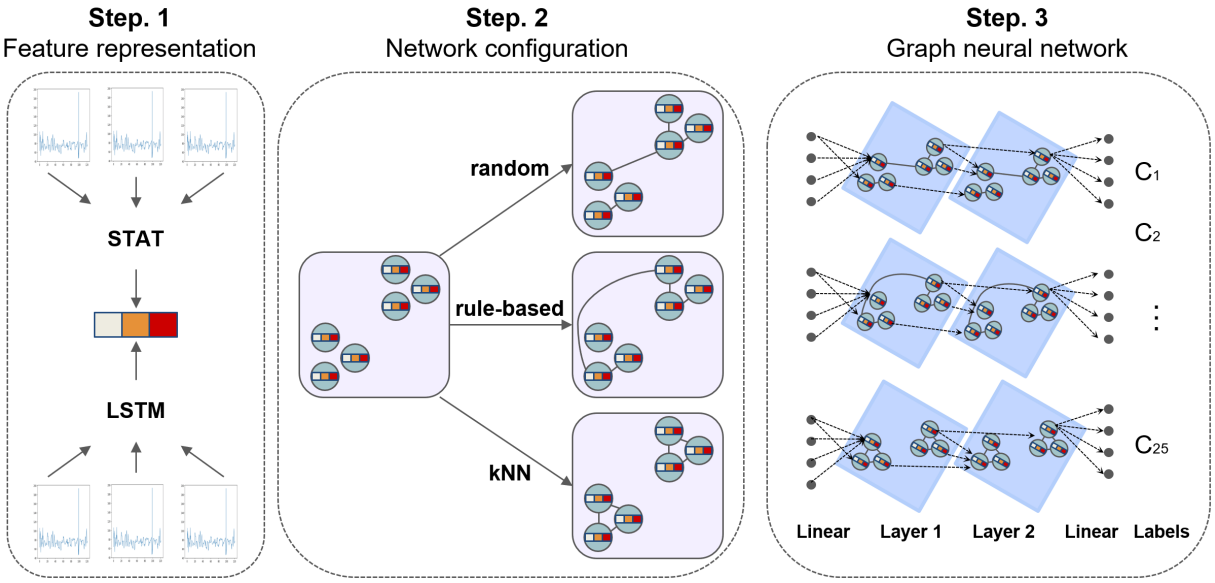

Fig. 1. Data and training pipeline. Left. Patient vital sign time series were used to create two distinct feature sets of 714-dimensional vectors by concatenating their statistical profiles and the 256-dimensional hidden-state embeddings using a pre-trained LSTM network. Center. Homogeneous graphs were built based on patient feature similarities, using different edge-creation strategies. Right. Using the patient similarity network as input, we trained distinct GNNs for each graph to perform multilabel classification in the benchmark of [3].

i is the index of the episode. These true labels do not share the same prevalence. Specifically, for the evaluated test set 'Other upper respiratory' complex phenotype is the least represented (4%), while 'Essential hypertension' is the most prevalent class (41.9%) [3]. For each episode, this multi-hot vector represents diagnoses taken from a particular set of 25 complex diseases, defined by the benchmark of [3], which defines a multi-label classification problem:

$$F_\Theta = X \rightarrow Y \tag{1}$$

where Θ represents the learnable parameters of the predictive model F_Θ, X is the set of all time series $\{X_i\}, X_i \in \mathbb{R}^{T_i \times 17}, i = 1, ..., N$ and $Y \in \{0,1\}^{N \times 25}$ is the matrix of all associated true labels. We optimize the parameter set Θ of the LSTM and the GNN networks with respect to the Binary Cross Entropy Loss function, defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \odot \log(\hat{y}_i) + (y_i - 1) \odot \log(\hat{y}_i - 1)) \cdot \mathbf{1_{25}} \tag{2}$$

where y_i is the i-th row of Y, $\hat{y}_i$ is the prediction of the model associated to the i-th episode, $\odot$ is the Hadamard product, $\mathbf{1_{25}}$ is a 1-vector of size 25, and $\cdot$ is the inner product. This means that the loss is minimized simultaneously for the 25 complex disease labels with a single model.

B. Temporal patient representations

A key challenge when working with time-series data is the dynamic input size that machine learning algorithms need to adjust to. To retrieve fixed-size feature vectors for each episode, we compared two strategies, statistical moment concatenation and LSTM-based hidden state, as shown in Step 1 of Fig. 1.

1) Concatenated Statistical Moments as features: We used a set of statistical features (STAT) based on different time intervals. Specifically, for each of the 17 vital signs we concatenated their average, variance, skewness, min, max, and length, all of them computed over seven different time windows (the first 10%, 25%, 50%, and the last 10%, 20%, 50% of the time-series events, and the whole episode). This resulted in a 714-dimensional (17 features $\times$ 6 moments $\times$ 7 time windows) statistical moment vector V_1.

2) LSTM features: LSTMs are capable of learning trends and long-term dependencies in time series [68]. For this reason, as an alternative to the statistical features, we pre-trained an LSTM network to perform the classification task defined above to embed patient trajectories. LSTMs were used by [20] in the benchmark of [3]. We use this as a baseline model and report its performance in Section IV. After pre-training, we extracted the hidden state from the final time-step of the LSTM output, which we defined as the 256-dimensional vector LSTM feature V_2:

$$(C, H) = \text{LSTM}(X), V_2 = H[T] \tag{3}$$

where C is the set of LSTM cell states for all time steps, H is the set of hidden states and T is the last time step of an ICU episode.

C. Patient similarity network

The construction process of the patient similarity networks is illustrated in Step 2 of Fig. 1. Using different edge creation strategies, patient networks were defined as homogeneous graphs of the form:

$$G = (V, E) \tag{4}$$

where $V \in R^N$ is the set of vertices (nodes) and $E \subseteq \{(u,v)|u,v \in V\}$ is the set of Boolean edges, including self-loops. We applied different strategies to define node features and edge connectivity structures.

1) Node features: We used both feature sets, V_1, V_2, to create separate models for each. The number of node features is 714 when using statistical moment vectors V_1 and 256 when using LSTM hidden state vectors V_2.

2) Edge definition: We evaluated whether introducing social network connectivity increases model predictive performance in automated phenotyping. Each edge creation strategy and its variations introduce a new graph that we evaluated, based on the benchmark of [3]. The different edge creation strategies were the following: trivial graph with self-loops, random edges sampled from the uniform distribution, k-Nearest Neighbors connectivity [69], and expert-rule-based connectivity, as presented in Table I. The trivial graph strategy was used to control for the impact of applying the message propagation algorithm (see Subsection III-D). In the random strategy, we created $300,000$ random connections for pairs of episodes sampled from the uniform distribution (i.e., any pair of nodes has the same probability of being sampled). In the k-NN strategy, we connected every node to the k nearest nodes of the network, based on the similarity of their features. We compared different ways of computing node feature similarity, for different values of k and using Euclidean distance.

Regarding the expert-rules strategy, we introduced three distinct variations: exact-match, lenient-match, and flexible-match. These different variations are based on the common vital sign abnormality categories (abnormal-low, abnormal-high, normal) derived from [70] (see Table I), which we used to connect pairs of nodes in different ways. For the expert-rules connectivity strategy 17 features are used, 16 vital signs and 2 features (weight and height), as illustrated in Fig. I.

For the exact-match variation, we first computed, for each episode and vital sign feature, whether the episode contains any abnormal-low, abnormal-high, or normal value for this feature. To be categorized as such, an episode was only required that the episode contains at least one abnormal-low, abnormal-high, normal value in the whole time series, respectively. This led to a tensor $C^E \in \{0,1\}^{N \times 15 \times 3}$:

$$C^E_{i,j,k} = \begin{cases} 1 & x_{i,j} \text{ is } k \\ 0 & else \end{cases}, \tag{5}$$

where i, j, and k denote episode index, feature index, and category (abnormal-high, abnormal-low, or normal), respectively. We used C^E to define edges. Each pair of nodes indexed by $\{i_1, i_2 \in \{1, ..., N\}, i_1 \neq i_2\}$, was connected if C_{i_1} and C_{i_2} shared all their elements, i.e., their categorization vector $C_{i_1,j}$ and $C_{i_2,j}$ was the same for all features j.

For the flexible-match, we built the tensor $C^F \in \{0,1\}^{N \times 15 \times 2}$ by merging both categories abnormal-high and abnormal-low into a new category "abnormal", which accounts for the existence of any abnormal vital sign value (i.e., either abnormal-high or abnormal-low) in the time series. Then, we defined edges by applying the same algorithm as in the exact-match variation but using C_F. This means that any pair of nodes that share abnormalities for all vital sign features, irrespective of whether they are too high or too low, was connected.

For the lenient-match variation, we further reduced C^F to $C^L \in \{0,1\}^{N \times 8 \times 2}$ by merging features that belonged to the same group of vital signs (e.g., circulatory system, respiratory system, etc.), as defined in the column Group of Table I. Again, we defined edges by applying the same algorithm as in the exact-match and flexible-match variations but using C_L. This means that any pair of nodes that shares abnormalities across groups of vital signs was connected.

All these strategies lose fine-grained information about time-series, as well as the exact feature values, since they pool whole patient trajectories into a Boolean vector. Still, these strategies infuse the knowledge of medical expert and reduce the dimensionality of the edge-selection problem, which might prune unnecessary or irrelevant connections between patients.

D. Graph neural network model

The creation of GNNs is the last step of the pipeline of Fig. 1. GNNs are based on the message propagation algorithm which exploits information from a node's neighborhood. In the context of node classification problems, GNNs produce node embeddings. In our case, GNNs could take as input any of the graphs defined in the previous section. We use different GNN layers which are well established in the literature and which we briefly outline in the next sections.

a) Graph Convolutional Network - GCN: In the context of node classification, GCN layers operate by aggregating feature information from neighboring nodes to enhance the feature representation of each node. Each GCN layer captures and leverages the local topological structure of the graph. Various functions can be used for the aggregation operation, provided they are either permutation-invariant or permutation-equivariant. Popular choices for these functions include mean, summation, minimum, and maximum. In our case, we used the minimum aggregator since it performed best, based on preliminary results. We compute the k-th layer node embeddings as the result of two steps: AGGREGATE and UPDATE.

$$m_{k,n} = \text{AGGREGATE}(h_{k-1,u} : u \in \mathcal{N}(n)), \tag{6}$$

$$h_{k,n} = \text{UPDATE}(h_{k-1,n}, m_{k,n}, h_{0,n}), \tag{7}$$

where $\mathcal{N}(n)$ represents the neighborhood of node n and $h_{k,n}$ is the set of node embeddings, and $h_{0,n}$ the initial node embeddings prior to the application of GCN. Note that AGGREGATE is a permutation invariant function. Finally, the number of layers k is treated as a hyper-parameter allowing to control the receptive field of the nodes. While the original GCN relies on predefined aggregation functions, GraphSAGE [71] employs a more flexible one and incorporates a neighborhood sampling step to better exploit local neighborhood features. This sampling strategy, which selects a subset of neighbors rather than all of them, mitigates memory usage. We have adopted this approach in all of our GNN models to enhance its efficiency when handling larger graphs.

TABLE I

VITAL SIGN INTERVALS FOR EXPERT-RULE EDGE CREATION. BASED ON THE ABNORMAL CATEGORIZATION FOR EACH VITAL SIGN WE APPLY A SET OF EXPERT RULES TO DERIVE WHETHER TWO PATIENTS SHOULD BE CONNECTED OR NOT IN A GRAPH. VALUE REFERENCE SOURCE: [70].

Variable	Group	Abnormal Low	Normal	Abnormal High	Min	Max	Mean	St.Deviation
Oxygen saturation (%)	1	<95	95-100	-	0	6.30×10^6	99.3	3.50×10^3
Respiratory rate ($/min$)	1	<12	12-20	>20	0	2.30×10^6	20.4	1.10×10^3
Heart rate (bpm)	2	<60	60-100	>100	-88	8.60×10^4	86.4	4.60×10
Systolic blood pressure (mmHg)	2	<90	90-120	>120	-11	1.40×10^5	122.3	1.40×10^2
Diastolic blood pressure (mmHg)	2	<60	60-80	>80	-16	1.14×10^5	61.8	2.40×10^2
Mean Blood Pressure (mmHg)	2	<60	60-110	>110	-135	1.10×10^5	79.6	1.09×10^2
Capillary Refill Time (sec.)	2	-	[0-3)	≥ 3	0	1	0.1	0.30
Glasgow coma scale eye opening	3	<4	4	-	0	4	3.2	1.10
Glasgow coma scale verbal response	3	<5	5	-	1	5	3	1.90
Glasgow coma scale motor response	3	<6	6	-	1	6	5.2	1.40
Glasgow coma scale total	3	<15	15	-	3.0	15	11.5	3.70
Temperature (°C)	4	<36.5	36.5-37.5	>37.5	-3	5.30×10^3	37	5.30
Blood pH	5	<7.35	7.35-7.45	>7.45	0	7.50×10^2	7.2	1.90
Body mass index (kg/m^2)	6	<18.5	[18.5-25)	≥ 25	0	1.60×10^5	33.2	1.90×10^2
Glucose (mg/dL)	7	<70	70-108	>108	0	9.90×10^5	144.2	1.40×10^3

b) Chebysev convolution network - ChebConv: It is well established in the literature that expressing features in the frequency domain can help facilitate the classification of signals [72]. ChebConv [27] is a variant of GCN that performs spectral transformation of the node features by applying a localized convolution filter to the graph Laplacian, $\mathcal{L} = D - I$, where D is the Degree matrix $D_{ii} = \sum_j W_{ij}$ and I is the Identity matrix. Additionally ChebConv applies an efficient pooling strategy to provide a summarized representation by rearranging nodes as a binary tree in order to make graph connectivity coarser.

c) Cluster-GCN - CGCN: Motivated by the polynomial computational complexity associated with the number of GNN layers, ClusterGCN [74] leverages clustering and sampling neighborhood, mitigating issues related to exponential neighborhood expansion.

The parameters of the edge selection strategies are presented in Table II and illustrated in Figure 2. The k-NN graph has a smaller variance in the number of edges per node, while expert strategies offer a non-uniform distribution, indicating the existence of "social" nodes. The distribution of edges per node is illustrated in a qualitative manner in Fig. 2.

For each possible combination of input features (statistical moments versus LSTM hidden states), edge construction strategies (trivial graph, uniformly sampled random edges, k-NN, and all variations of expert-rule-based), GNN layer (GCN, ChebConv, CGCN), we trained GNNs and LSTM parameters (when applicable) to perform multi-label node classification, as required by the benchmark of [3]. We trained the GNN models both in inductive and transductive settings. We used additional linear layers before and after the GNN layers since we empirically found that it enhances performance. We experimented with zero, one, two, and three linear layers before and after the graph layers. We used a two-phase training approach for our hybrid LSTM-GNN model, similar to the method described in [75]. This approach allowed us to reuse the pre-trained LSTM model for temporal embedding retrieval and facilitated the exploration of diverse graph topologies and architectures.

IV. RESULTS

A. Automated phenotyping - multilabel classification

We trained distinct models on the training set and evaluated them on the benchmark's test set [3], using consistent preprocessing and imputation strategies. We estimated the mean and standard deviation of the performance by evaluating the models based on 1,000 bootstrap samples, and the corresponding 95% confidence intervals. In addition to the micro, macro, and weighted Area Under the Receiving Operating Characteristic (AUROC) metrics reported in the literature, we also report the micro Accuracy, macro Precision, Recall, and F1 metrics, taking into consideration that the class prevalence is highly imbalanced. We achieved state-of-the-art results in the transductive setting and outperformed the reported baselines in the inductive setting using a two-phase training approach involving LSTM and GNN. In the inductive setting, the best performance is achieved using ChebConv on the trivial graph with LSTM node features. In contrast, for the transductive setting, the best performance is achieved by CGCN on the lenient-match connectivity strategy using STAT node features. The best LSTM model we trained reproduces the baseline performance reported by [20] on the benchmark. For all GNNs, neighbor sampling [76] was applied due to the graph size being too large to fit in memory. Hyper-parameter tuning was performed using the Optuna framework on the validation set [77]. For the transductive setting, 200 trials were conducted to explore the optimal model parameters. Through this extensive experimentation, the model with the optimal configuration comprises one pre-processing linear layer, a two-layered GNN, and a post-processing linear layer. In our experiments, including pre-processing and post-processing linear layers in the architecture consistently led to performance improvements. For the best transductive model the 'minimum' function was used for aggregation, the hidden dimensionality of layers was set to 2663, and a maximum of 78 nodes was used for neighbor sampling. Models were trained over 10

TABLE II
KEY PARAMETERS OF THE DIFFERENT EDGE SELECTION STRATEGIES.

Descriptive network metric	Random	Exact	Flexible	Lenient	k-NN
Number of edges	2.99×10^5	2.61×10^4	7.01×10^6	1.63×10^8	3.45×10^5
Average Centrality	3.41×10^{-4}	2.96×10^{-4}	7.90×10^{-3}	1.85×10^{-2}	3.93×10^{-4}
Average Degree	1.43×10	1.20×10	3.34×10^2	7.78×10^5	1.64×10
Average Clustering coefficient	3.41×10^{-4}	3.89×10^{-5}	9.70×10^{-3}	5.30×10^{-4}	1.89×10^{-1}

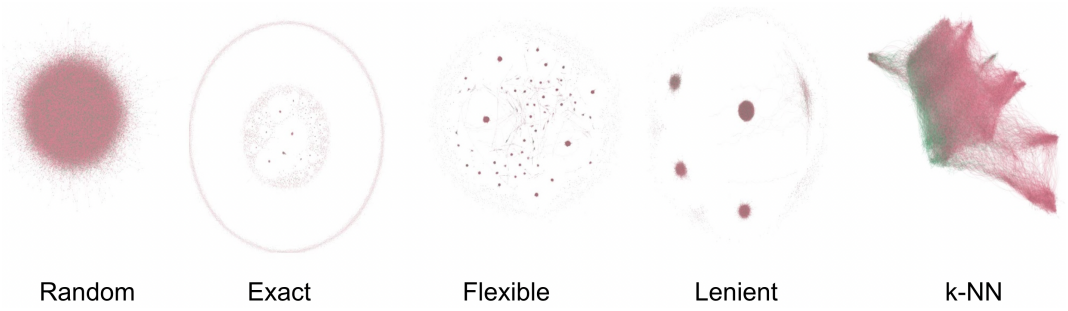

Fig. 2. Comparison of edge strategies. The number of edges used for the visualization is based on a subgraph of $100,000$ edges. The layout was created using the Force-Atlas-2 Gephi algorithm [73]. Green and red correspond to the true label for the complex disease of "Acute and unspecified renal failure" (green: positive and red: negative diagnosis).

epochs using the Adam optimizer [78] with a learning rate of 9×10^{-4} and weight decay of 10^{-4}.

B. Inductive GNN performance

We trained all graph-based models using the LSTM hidden state as node features, maintaining the same training and test splits. During training, a subgraph using just the training nodes was used. The evaluation metrics of the inductive models are reported in Tables III and IV. We note that our GNN models outperform both baseline models (LSTM, Logistic regression). Consistently, LSTM node features achieve superior performance as compared to static node features in all evaluated metrics with the exception of the Recall metric. The best model is ChebConv GNN with trivial connectivity, which outperforms all other combinations in almost all metrics. One exception is the GCN with random connectivity, which obtains the best recall score. These findings suggest that the evaluated connectivity strategies had minimal impact in the inductive setting. Overall, in the inductive setting, the different network configurations did not augment the capacity of our models compared to the trivial graph, however, this is not the case for the transductive setting. Finally, we employed the chi-square McNemar's test to compare the best inductive model against the LSTM baseline, yielding a p-value of $4\times\ 10^{-9}$.

C. Transductive GNN performance

The transductive setting was evaluated using the same training and test splits as the inductive setting. However, it was evaluated to maintain connections with the nodes of the test set, as the graph is fully observed, with loss optimization focused only on the subgraph of training nodes. This approach allows this set of models to retain validation and test node feature information, greatly enhancing performance when compared to the inductive setting. Evaluation metrics for the transductive models are presented in Tables V and VI. Transductive models outperformed both the baseline and all the GNN inductive models. Notably, in contrast to the performance of inductive GNNs, the STAT node features surpassed the LSTM node features across all evaluated metrics. The lenient-match connectivity strategy outperforms other strategies in all metrics. We used the chi-squared McNemar's test to compare the best transductive model against the LSTM baseline, resulting in a p-value of 3×10^{-10}. This result can be attributed to the high number of edges using this strategy, wherein test nodes are more likely to connect to training nodes. Amongst all layer types, CGCN achieves the best performance in all metrics. Note that the improved performance in a transductive setting comes at the cost of needing to retrain the model for each new node added to the graph. This presents a trade-off between speed and performance when choosing relevant methods.

Among all graph variations, in the inductive setting, the introduction of edge connectivity did not enhance performance. However, the graph definition had a significant influence in the case of transductive learning, especially in the expert-rule lenient strategy which is characterized by a balanced average Clustering Coefficient, as seen in Table II. To investigate the capacity of derived and baseline models we compare the best GNN models to ensemble Logistic Regression (LR) [3] and LSTM [20] baselines in Figure 3. We used the Area Under the Precision-Recall (AUPRC) metric to evaluate their ability to detect the rarest 'Other upper respiratory disease' phenotype among the labels. Our results indicate that the use of a transductive learning model significantly enhances rare phenotype identification compared to the baselines.

TABLE III

EVALUATION METRIC SCORES FOR INDUCTIVE MODELS (USING STAT NODE FEATURES-BASED GNNS). MEAN PERFORMANCE BASED ON 1,000 BOOTSTRAP SAMPLES AND THE CORRESPONDING 95% CONFIDENCE INTERVALS.

Model	ES	Precision	Recall	F1 score	Accuracy	AUROC		
						Micro	Macro	Weighted
GCN	Trivial	0.3463	0.5716	0.4232	0.7590	0.8094±0.0002	0.7558±0.0002	0.7462±0.0002
CGCN	Trivial	0.3469	0.5719	0.4260	0.7635	0.8116±0.0001	0.7572±0.0001	0.7480±0.0001
CHEB	Trivial	**0.3519**	0.5769	**0.4315**	**0.7662**	**0.8146±0.0001**	**0.7628±0.0001**	**0.7537±0.0001**
GCN	Random	0.2857	**0.5953**	0.3769	0.6901	0.7499±0.0001	0.6982±0.0001	0.6876±0.0001
CGCN	Random	0.2973	0.5950	0.3799	0.6834	0.7492±0.0002	0.6982±0.0002	0.6893±0.0002
CHEB	Random	0.2977	0.5751	0.3758	0.6952	0.7466±0.0001	0.6936±0.0001	0.6856±0.0001
GCN	Exact	0.3244	0.5309	0.3946	0.7459	0.7719±0.0001	0.7112±0.0001	0.7041±0.0001
CGCN	Exact	0.3276	0.5433	0.4025	0.7497	0.7739±0.0002	0.7148±0.0001	0.7076±0.0001
CHEB	Exact	0.3240	0.5543	0.4009	0.7420	0.7735±0.0001	0.7161±0.0001	0.7085±0.0001
GCN	Lenient	0.3242	0.5174	0.3870	0.7420	0.7657±0.0001	0.7083±0.0002	0.6998±0.0001
CGCN	Lenient	0.3164	0.5316	0.3856	0.7332	0.7608±0.0002	0.7066±0.0002	0.6959±0.0002
CHEB	Lenient	0.3242	0.5640	0.4036	0.7377	0.7827±0.0001	0.7307±0.0001	0.7183±0.0001
GCN	Flexible	0.3145	0.5447	0.3886	0.7271	0.7671±0.0001	0.7124±0.0001	0.7039±0.0001
CGCN	Flexible	0.3252	0.5324	0.3928	0.7397	0.7703±0.0001	0.7159±0.0001	0.7052±0.0001
CHEB	Flexible	0.3202	0.5728	0.4006	0.7331	0.7795±0.0001	0.7288±0.0001	0.7159±0.0001
GCN	k-NN	0.3069	0.5730	0.3928	0.7218	0.7725±0.0001	0.7185±0.0001	0.7079±0.0001
CGCN	k-NN	0.3123	0.5588	0.3919	0.7312	0.7734±0.0002	0.7202±0.0002	0.7097±0.0002
CHEB	k-NN	0.3044	0.5730	0.3881	0.7154	0.7685±0.0001	0.7112±0.0001	0.7011±0.0001

TABLE IV

EVALUATION METRIC SCORES FOR INDUCTIVE MODELS. IN THE FIRST TWO ROWS, THE ENSEMBLE LR AND LSTM BASELINES ARE PRESENTED. THE BEST-PERFORMING GNN MODELS ON THE MACRO-AUROC METRIC FOR EACH LSTM NODE FEATURE-BASED NETWORK CONFIGURATION ARE DISPLAYED. MEAN PERFORMANCE IS BASED ON 1,000 BOOTSTRAP SAMPLES, WITH THE CORRESPONDING 95% CONFIDENCE INTERVALS ALSO SHOWN.

Model	ES	Precision	Recall	F1 score	Accuracy	AUROC		
						Micro	Macro	Weighted
LR	–	0.3577	0.5439	0.4214	0.7664	0.8005±0.0001	0.7408±0.0060	0.7323±0.0037
LSTM	–	0.3726	0.5796	0.4427	0.7713	0.8194±0.0001	0.7692±0.0001	0.7563±0.0001
CHEB	Trivial	**0.3730**	0.5515	**0.4411**	**0.7857**	**0.8200±0.0001**	**0.7703±0.0001**	**0.7569±0.0001**
GCN	Random	0.2971	0.5695	0.3774	0.7019	0.7482±0.0001	0.6942±0.0002	0.6822±0.0001
CGCN	Exact	0.3378	0.5749	0.4172	0.7519	0.7995±0.0001	0.7449±0.0001	0.7336±0.0001
CGCN	Lenient	0.3162	**0.5840**	0.3953	0.7144	0.7720±0.0001	0.7185±0.0001	0.7055±0.0001
CGCN	Flexible	0.3374	0.5308	0.3971	0.7433	0.7721±0.0001	0.7194±0.0002	0.7070±0.0002
CHEB	k-NN	0.3178	0.5673	0.3901	0.7154	0.7651±0.0001	0.7095±0.0001	0.6971±0.0001

TABLE V

EVALUATION METRIC SCORES FOR TRANSDUCTIVE MODELS (USING STAT NODE FEATURES-BASED GNNS). MEAN PERFORMANCE IS BASED ON 1,000 BOOTSTRAP SAMPLES, WITH THE CORRESPONDING 95% CONFIDENCE INTERVALS.

Model	ES	Precision	Recall	F1 score	Accuracy	AUROC		
						Micro	Macro	Weighted
GCN	Trivial	0.3445	0.5714	0.4219	0.7594	0.8100±0.0001	0.7557±0.0001	0.7469±0.0001
CGCN	Trivial	0.3473	0.5715	0.4258	0.7624	0.8111±0.0001	0.7580±0.0001	0.7485±0.0001
CHEB	Trivial	0.3523	0.5773	0.4318	0.7659	0.8136±0.0001	0.7624±0.0001	0.7531±0.0001
GCN	Random	0.3057	0.5770	0.3866	0.7110	0.7604±0.0002	0.7066±0.0002	0.6962±0.0002
CGCN	Random	0.2970	0.6102	0.3802	0.6736	0.7492±0.0001	0.6980±0.0001	0.6889±0.0001
CHEB	Random	0.2936	0.5847	0.3750	0.6875	0.7458±0.0001	0.6917±0.0002	0.6838±0.0002
GCN	Exact	0.5585	0.5996	0.5671	0.8531	0.8729±0.0001	0.8453±0.0001	0.8372±0.0002
CGCN	Exact	0.6620	0.6249	0.6344	0.8838	0.8934±0.0002	0.8716±0.0002	0.8659±0.0002
CHEB	Exact	0.5634	0.6063	0.5719	0.8550	0.8756±0.0002	0.8481±0.0002	0.8399±0.0002
GCN	Lenient	0.9882	0.9654	0.9766	0.9927	0.9995±0.0000	0.9994±0.0000	0.9993±0.0000
CGCN	Lenient	**0.9987**	**0.9957**	**0.9972**	**0.9991**	**0.9999±0.0000**	**0.9999±0.0000**	**0.9998±0.0000**
CHEB	Lenient	0.9890	0.9494	0.9687	0.9898	0.9990±0.0000	0.9988±0.0000	0.9987±0.0000
GCN	Flexible	0.9446	0.8605	0.8994	0.9706	0.9894±0.0000	0.9873±0.0001	0.9873±0.0000
CGCN	Flexible	0.9771	0.9673	0.9721	0.9913	0.9958±0.0000	0.9946±0.0000	0.9951±0.0000
CHEB	Flexible	0.9297	0.8582	0.8916	0.9666	0.9876±0.0001	0.9860±0.0001	0.9848±0.0001
GCN	k-NN	0.6721	0.5542	0.5974	0.8846	0.8987±0.0001	0.8808±0.0002	0.8741±0.0001
CGCN	k-NN	0.8526	0.7778	0.8113	0.9441	0.9598±0.0001	0.9516±0.0001	0.9517±0.0001
CHEB	k-NN	0.7442	0.6909	0.7120	0.9132	0.9389±0.0001	0.9303±0.0001	0.9249±0.0001

TABLE VI

EVALUATION METRIC SCORES FOR TRANSDUCTIVE MODELS. THE BEST PERFORMING GNN MODELS ON THE MACRO-AUROC METRIC FOR EACH LSTM NODE FEATURE-BASED NETWORK CONFIGURATION ARE DISPLAYED. MEAN PERFORMANCE IS BASED ON 1,000 BOOTSTRAP SAMPLES, WITH THE CORRESPONDING 95% CONFIDENCE INTERVALS ALSO SHOWN.

Model	ES	Precision	Recall	F1 score	Accuracy	AUROC		
						Micro	Macro	Weighted
CGCN	Trivial	0.3675	0.5641	0.4380	0.7761	0.8153±0.0001	0.7683±0.0002	0.7546±0.0001
CGCN	Random	0.2919	0.5989	0.3756	0.6788	0.7445±0.0001	0.6909±0.0001	0.6803±0.0001
CGCN	Exact	0.5890	0.5931	0.5787	0.8614	0.8722±0.0001	0.8435±0.0002	0.8362±0.0002
CGCN	Lenient	**0.9975**	**0.9957**	**0.9966**	**0.9989**	**0.9998±0.0000**	**0.9998±0.0000**	**0.9998±0.0000**
CGCN	Flexible	0.9735	0.9596	0.9664	0.9897	0.9951±0.0000	0.9941±0.0000	0.9943±0.0000
CGCN	k-NN	0.8857	0.8456	0.8644	0.9595	0.9750±0.0000	0.9703±0.0001	0.9692±0.0001

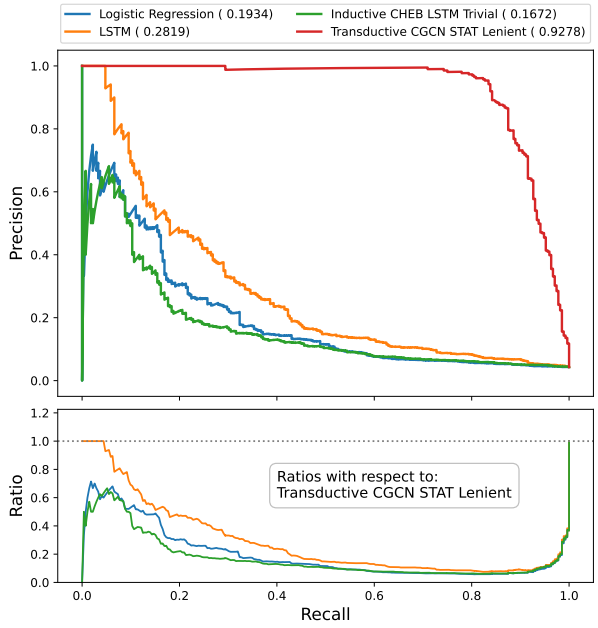

Fig. 3. Area Under the Precision-Recall Curve (AUPRC) comparison for various models including the best inductive and transductive approaches, along with LSTM and Logistic Regression baselines. These are evaluated on the 'Other upper respiratory' disease class, the most rare in the [3] benchmark. The ratio plot indicates the superior performance of the transductive model across all thresholds.

V. DISCUSSION

We constructed homogeneous graphs for the time-series multilabel classification automated phenotyping benchmark, achieving better performance than previously reported methods. While graph connectivity strategies did not enhance performance in the inductive setting, they significantly impacted performance in transductive learning. Additionally, we observed that GNNs trained on networks with a higher number of edges performed significantly better in the transductive setting. We attribute this to the dense connectivity between testing and training node features. However, this comes at the cost of needing to train a new model from scratch to make an inference presenting a sensitivity versus efficiency trade-off. Concerning the impact of node features, LSTM node features significantly outperformed the STAT ones in the inductive setting. However, the trend was reversed in the transductive setting with STAT node features giving a better performance. We hypothesize that this is due to the higher dimensionality (714 versus 256), allowing the GNNs to retain more accurate descriptors of the test node features acting as identifiers. Furthermore, graphs with a higher number of edges demonstrated better performance, especially in the case of rule-based strategies.

In this paper, we have demonstrated the significant impact of homogeneous graph definitions, applied through inductive and transductive learning, on the performance of Graph Neural Networks (GNNs) when classifying ICU episodes. Our results have not only improved upon the results reported in the literature but have also contributed to the replicable benchmark established by [3]. By comparing inductive and transductive GNNs, we demonstrated that transductive learning achieves almost perfect results, which might be relevant in certain settings, especially for less prevalent complex diseases. However, since no inherent network structure exists for similarity-based datasets, further research is needed to evaluate the impact of networks in the inductive learning setting.

REFERENCES

[1] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 160035, 2016.

[2] A. E. W. Johnson, T. J. Pollard, and R. G. Mark, "MIMIC-III clinical database (version 1.4)," 2016.

[3] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific Data*, vol. 6, no. 1, jun 2019.

[4] N. Akhoon, "Precision medicine: A new paradigm in therapeutics," *International Journal of Preventive Medicine*, vol. 12, 1 2021.

[5] H. Alzoubi, R. Alzubi, N. Ramzan, D. West, T. Al-Hadhrami, and M. Alazab, "A Review of Automatic Phenotyping Approaches using Electronic Health Records," *Electronics*, vol. 8, p. 1235, Oct. 2019.

[6] T. J. Loftus, B. Shickel, J. A. Balch, P. J. Tighe, K. L. Abbott, B. Fazzone, E. M. Anderson, J. Rozowsky, T. Ozrazgat-Baslanti, Y. Ren, S. A. Berceli, W. R. Hogan, P. A. Efron, J. R. Moorman, P. Rashidi, G. R. Upchurch, and A. Bihorac, "Phenotype clustering in health care: A narrative review for clinicians," *Frontiers in Artificial Intelligence*, vol. 5, p. 184, 8 2022.

[7] A. Oellrich, N. Collier, T. Groza, D. Rebholz-Schuhmann, N. Shah, O. Bodenreider, M. R. Boland, I. Georgiev, H. Liu, K. Livingston *et al.*, "The digital revolution in phenotyping," *Briefings in bioinformatics*, vol. 17, no. 5, pp. 819–830, 2016.

[8] S. Yang, P. Varghese, E. Stephenson, K. Tu, and J. Gronsbell, "Machine learning approaches for electronic health records phenotyping: a methodical review," *Journal of the American Medical Informatics Association*, vol. 30, no. 2, pp. 367–381, 2023.

[9] J. C. Kirby, P. Speltz, L. V. Rasmussen, M. Basford, O. Gottesman, P. L. Peissig, J. A. Pacheco, G. Tromp, J. Pathak, D. S. Carrell, S. B. Ellis, T. Lingren, W. K. Thompson, G. Savova, J. Haines, D. M. Roden, P. A. Harris, and J. C. Denny, "Phekb: a catalog and workflow for creating electronic phenotype algorithms for transportability," *Journal of the American Medical Informatics Association : JAMIA*, vol. 23, pp. 1046–1052, 11 2016.

[10] *Cluster Analysis of Low-Dimensional Medical Concept Representations from Electronic Health Records*. Springer, Cham, 2022.

[11] R. Gouareb, A. Bornet, D. Proios, S. G. Pereira, and D. Teodoro, "Detection of patients at risk of enterobacteriaceae infection using graph neural networks: a retrospective study," *medRxiv*, pp. 2023–06, 2023.

[12] A. Bornet, D. Proios, A. Yazdani, F. Jaume-Santero, G. Haller, E. Choi, and D. Teodoro, "Comparing neural language models for medical concept representation and patient trajectory prediction," *medRxiv*, pp. 2023–06, 2023.

[13] Y. Si, J. Du, Z. Li, X. Jiang, T. Miller, F. Wang, W. J. Zheng, and K. Roberts, "Deep representation learning of patient data from electronic health records (ehr): A systematic review," *Journal of biomedical informatics*, vol. 115, p. 103671, 2021.

[14] L. Caroprese, P. Veltri, E. Vocaturo, and E. Zumpano, "Deep learning techniques for electronic health record analysis," in *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2018, pp. 1–4.

[15] 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 journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[16] J. De Freitas, K. Johnson, E. Golden, G. Nadkarni, J. Dudley, E. Bottinger, B. Glicksberg, and R. Miotto, "Phe2vec: automated disease phenotyping based on unsupervised embeddings from electronic health records. patterns 2 (9), 100337 (2021)," 2021.

[17] S. Gehrmann, F. Dernoncourt, Y. Li, E. T. Carlson, J. T. Wu, J. Welt, J. Foote Jr, E. T. Moseley, D. W. Grant, P. D. Tyler *et al.*, "Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives," *PloS one*, vol. 13, no. 2, p. e0192360, 2018.

[18] C. Shivade, P. Raghavan, E. Fosler-Lussier, P. J. Embi, N. Elhadad, S. B. Johnson, and A. M. Lai, "A review of approaches to identifying patient phenotype cohorts using electronic health records," *Journal of the American Medical Informatics Association*, vol. 21, no. 2, pp. 221–230, 2014.

[19] I. Landi, B. S. Glicksberg, H. C. Lee, S. Cherng, G. Landi, M. Danieletto, J. T. Dudley, C. Furlanello, and R. Miotto, "Deep representation learning of electronic health records to unlock patient stratification at scale," *npj Digital Medicine 2020 3:1*, vol. 3, pp. 1–11, 7 2020.

[20] C. Zang and F. Wang, "Scehr: Supervised contrastive learning for clinical risk prediction using electronic health records," 2021.

[21] Y. Zhang, Q. Wu, N. Peng, M. Dai, J. Zhang, and H. Wang, "Memory-gated recurrent networks," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 12B, pp. 10 956–10 963, 12 2020.

[22] S. Ferdowsi, N. Borissov, J. Knafou, P. Amini, and D. Teodoro, "Classification of hierarchical text using geometric deep learning: the case of clinical trials corpus," 2021.

[23] S. Ferdowsi, J. Copara, R. Gouareb, N. Borissov, F. Jaume-Santero, P. Amini, and D. Teodoro, "On graph construction for classification of clinical trials protocols using graph neural networks," in *International Conference on Artificial Intelligence in Medicine*. Springer, 2022, pp. 249–259.

[24] S. Ferdowsi, J. Knafou, N. Borissov, D. V. Alvarez, R. Mishra, P. Amini, and D. Teodoro, "Deep learning-based risk prediction for interventional clinical trials based on protocol design: A retrospective study," *Patterns*, vol. 4, no. 3, 2023.

[25] E. Rocheteau, C. Tong, P. Veličković, N. Lane, and P. Liò, "Predicting patient outcomes with graph representation learning," 2021.

[26] E. Choi, Z. Xu, Y. Li, M. W. Dusenberry, G. Flores, Y. Xue, and A. M. Dai, "Graph convolutional transformer: Learning the graphical structure of electronic health records," *CoRR*, vol. abs/1906.04716, 2019.

[27] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, pp. 3844–3852, 6 2016.

[28] S. Wang, M. B. A. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes, and T. Naumann, "Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii," in *Proceedings of the ACM Conference on Health, Inference, and Learning*, ser. CHIL '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 222–235.

[29] WHO, "International statistical classification of diseases and related health problems (11th ed.)." 2019.

[30] J. R. A. Solares, F. E. D. Raimondi, Y. Zhu, F. Rahimian, D. Canoy, J. Tran, A. C. P. Gomes, A. H. Payberah, M. Zottoli, M. Nazarzadeh *et al.*, "Deep learning for electronic health records: A comparative review of multiple deep neural architectures," *Journal of biomedical informatics*, vol. 101, p. 103337, 2020.

[31] Y. Si, J. Du, Z. Li, X. Jiang, T. Miller, F. Wang, W. J. Zheng, and K. Roberts, "Deep representation learning of patient data from electronic health records (ehr): A systematic review," *Journal of biomedical informatics*, vol. 115, p. 103671, 2021.

[32] Z. Che, D. Kale, W. Li, M. T. Bahadori, and Y. Liu, "Deep computational phenotyping," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Aug. 2015.

[33] Y. Cheng, F. Wang, P. Zhang, and J. Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, jun 2016.

[34] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Duggan, J. Irvine, Q. Le, K. Litsch, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum, K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. D. Howell, C. Cui, G. S. Corrado, and J. Dean, "Scalable and accurate deep learning with electronic health records," *npj Digital Medicine*, vol. 1, no. 1, may 2018.

[35] Y. Zhang, H. Zhou, J. Li, W. Sun, and Y. Chen, "A time-sensitive hybrid learning model for patient subgrouping," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[36] Y. Choi, C. M. Y.-I. Chiu, and D. Sontag, "Learning low-dimensional representations of medical concepts," *AMIA Summits on Translational Science Proceedings*, vol. 2016, p. 41, 2016.

[37] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi, "Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction," *NPJ digital medicine*, vol. 4, no. 1, p. 86, 2021.

[38] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[39] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi, "Behrt: transformer for electronic health records," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[40] C. Liu, F. Wang, J. Hu, and H. Xiong, "Temporal phenotyping from longitudinal electronic health records," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Aug. 2015.

[41] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "GRAM," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Aug. 2017.

[42] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, "KAME," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, Oct. 2018.

[43] Y. Cen, Z. Hou, Y. Wang, Q. Chen, Y. Luo, Z. Yu, H. Zhang, X. Yao, A. Zeng, S. Guo, Y. Dong, Y. Yang, P. Zhang, G. Dai, Y. Wang, C. Zhou, H. Yang, and J. Tang, "Cogdl: A toolkit for deep learning on graphs," 2021.

[44] M. Liu, Y. Luo, L. Wang, Y. Xie, H. Yuan, S. Gui, H. Yu, Z. Xu, J. Zhang, Y. Liu, K. Yan, H. Liu, C. Fu, B. Oztekin, X. Zhang, and S. Ji, "Dig: A turnkey library for diving into graph deep learning research," *J. Mach. Learn. Res.*, vol. 22, no. 1, jan 2021.

[45] X. Liu, H. Wang, T. He, Y. Liao, and C. Jian, "Recent advances in representation learning for electronic health records: A systematic review," in *Journal of Physics: Conference Series*, vol. 2188, no. 1. IOP Publishing, 2022, p. 012007.

[46] J. Xu, X. Xi, J. Chen, V. S. Sheng, J. Ma, and Z. Cui, "A survey of deep learning for electronic health records," *Applied Sciences*, vol. 12, no. 22, p. 11709, 2022.

[47] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[48] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.

[49] W. Ge, J.-W. Huh, Y. R. Park, J.-H. Lee, Y.-H. Kim, and A. Turchin, "An interpretable icu mortality prediction model based on logistic regression and recurrent neural networks with lstm units." in *AMIA Annual Symposium Proceedings*, vol. 2018. American Medical Informatics Association, 2018, p. 460.

[50] K. Yu, M. Zhang, T. Cui, and M. Hauskrecht, "Monitoring icu mortality risk with a long short-term memory recurrent neural network," in *Pacific Symposium on Biocomputing 2020*. World Scientific, 2019, pp. 103–114.

[51] E. Harrison, M. Chang, Y. Hao, and A. Flower, "Using machine learning to predict near-term mortality in cirrhosis patients hospitalized at the university of virginia health system," in *2018 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, 2018, pp. 112–117.

[52] B. Balkan, V. Subbian *et al.*, "Decompensation in critical care: early prediction of acute heart failure onset," *JMIR Medical Informatics*, vol. 8, no. 8, p. e19892, 2020.

[53] L. Zhang, X. Chen, T. Chen, Z. Wang, and B. J. Mortazavi, "Dynehr: Dynamic adaptation of models with data heterogeneity in electronic health records," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, 2021, pp. 1–4.

[54] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[55] B. K. Reddy and D. Delen, "Predicting hospital readmission for lupus patients: An rnn-lstm-based deep-learning methodology," *Computers in biology and medicine*, vol. 101, pp. 199–209, 2018.

[56] D. Lee, X. Jiang, and H. Yu, "Harmonized representation learning on dynamic ehr graphs," *Journal of biomedical informatics*, vol. 106, p. 103426, 2020.

[57] A. Rai, P. Pradhan, J. Nagraj, K. Lohitesh, R. Chowdhury, and S. Jalan, "Understanding cancer complexome using networks, spectral graph theory and multilayer framework," *Scientific reports*, vol. 7, no. 1, pp. 1–16, 2017.

[58] H. P. Sajjad, A. Docherty, and Y. Tyshetskiy, "Efficient representation learning using random walks for dynamic graphs," *arXiv preprint arXiv:1901.01346*, 2019.

[59] T. Wu, Y. Wang, Y. Wang, E. Zhao, and Y. Yuan, "Leveraging graph-based hierarchical medical entity embedding for healthcare applications," *Scientific reports*, vol. 11, no. 1, p. 5858, 2021.

[60] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 807–815, 2009.

[61] B. Malone, A. Garcia-Duran, and M. Niepert, "Learning representations of missing data for predicting patient outcomes," 2018.

[62] S. Liu, T. Li, H. Ding, B. Tang, X. Wang, Q. Chen, J. Yan, and Y. Zhou, "A hybrid method of recurrent neural network and graph neural network for next-period prescription prediction," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 2849–2856, 2020.

[63] Y. Dong, S. Wang, Y. Wang, T. Derr, and J. Li, "On structural explanation of bias in graph neural networks," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 316–326.

[64] Y. Ma, X. Liu, N. Shah, and J. Tang, "Is homophily a necessity for graph neural networks?" 2021.

[65] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 5999–6009, 6 2017.

[66] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[67] V. Vapnik, *Estimation of Dependences Based on Empirical Data*. Springer New York, 2006.

[68] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[69] F. Preparata and M. Shamos, *Computational Geometry: An Introduction*, ser. Monographs in Computer Science. Springer New York, 2012.

[70] "Statpearls - ncbi bookshelf."

[71] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *CoRR*, vol. abs/1706.02216, 2017.

[72] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013.

[73] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software," *PLOS ONE*, vol. 9, p. e98679, 6 2014.

[74] W. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," *CoRR*, vol. abs/1905.07953, 2019.

[75] M. T. Do, N. Park, and K. Shin, "Two-stage training of graph neural networks for graph classification," 2020.

[76] P. Hu and W. C. Lau, "A survey and taxonomy of graph sampling," 2013.

[77] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," *CoRR*, vol. abs/1907.10902, 2019.

[78] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.