Summary

Sharing data plays a important role in building new predictive models for instance, using real healthcare data to develop tools for early disease detection. Particularly in the medical domain, real-world data is high in demand because it has detailed patient histories, treatment outcomes, and health trends, making it a rich resource for research and development of better diagnostic and predictive tools.

In contrast, sharing data raises general privacy concerns. Individuals are often unwilling to share their personal details, particularly when the data includes sensitive domains such as healthcare history or behavioral patterns. Furthermore, regulations like General Data Protection Regulation(GDPR) and the Health Insurance Portabil- ity and Accountability Act (HIPAA) for protecting sensitive personal information, limit insightful datasets availability for research and development purposes.

These concerns increases when the data is longitudinal, which captures individuals records over time and are unlike static datasets, as they are expose able to subtle and unique behavior patterns. Even with de-identification or anonymisation techniques, the sequence of events in individuals record may still be distinctive enough to risk re-identification. For example, a patient's hospital visits, treatment cycles, and test results over several months might reveal their identity if matched with other available information. This challenge makes it difficult to openly share such data for research or analysis without compromising individual privacy.

To overcome this risk, Synthetic Data Generators (SDG) have emerged as a promising alternative. Synthetic datasets aim to preserve the statistical and structural properties of the original data while ensuring that no real individual's data is exposed. Different techniques have been proposed to generate synthetic data, with Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) among the most widely known. Their "black box" nature makes it hard to understand how decisions are made within these models, posing challenges for sensitive domains that demand transparency and interpretability.

To address these limitations, we investigated synthetic data generation for time series data and propose a non-stationary SDG for longitudinal data (NSSDG-LD). Our algorithm is based on a Dynamic Bayesian Network (DBN) with the capability to capture change points, where the conditional probability within a segment remains sames but varies from segment to segment. Our goal is to balance privacy and utility and to ensure that the generation process remains understandable and controllable. Our method builds on the foundational idea of change point detection from the cpBGe model, which identifies points in time where the behavior of a system changes. Rather than assuming the same statistical relationships hold throughout the entire dataset, we divide the data into segments based on these change points. Within each segment, we learn a DBN that models both the structure and conditional probabilities of the variables. This allows us to capture realistic temporal patterns while keeping the underlying model interpretable and stable.

We evaluate NSSDG-LD on both simulated and real-world datasets. Simulated benchmarks include 2-node, 4-node, and 8-node networks with known change points and structure, allowing for precise validation of inference accuracy. Real-world validation is performed on a longitudinal dataset derived from MIMIC-IV, in which each patient's hospital visits are encoded as categorical event sequences. This enables us to evaluate performance in a real-world clinical context with privacy-sensitive variables and changing care trends.

To measure model performance, we consider six categories of evaluation: (1) structure learning (via AUC against ground truth graphs), (2) change point detection (via F1 score), (3) marginal and (4) pairwise distribution similarity (using TVD and KL divergence), (5) cross-time dependency preservation (via mutual information gap between real and synthetic sequences), and (6) downstream predictive utility (via AUROC for classification tasks). Additionally, we assess privacy using a Membership Inference Attack (MIA), where an attacker attempts to infer whether real records were part of the training data based on the model outputs.

We conducted a comparative evaluation between our proposed NSSDG-LD model and the PARSynthesizer, a state of the art data generation framework specifically designed for longitudinal datasets. While PAR performed slightly higher in utility scores, indicating a closer statistical resemblance to the original data across marginal and pairwise distributions, our model achieved better privacy preservation when tested on MIMIC IV data. Specifically, NSSDG-LD achieved lower recall rates in MIA scenarios, demonstrating a reduced likelihood of real training records being exposed through synthetic data.

In the downstream classification task, we observed that the PAR model showed limited predictive capability and predictions were close to random, implying that it failed to capture meaningful patterns from the original data. In contrast, our proposed NSSDG-LD model achieved better classification performance, as measured by AUROC. This suggests that the synthetic data generated by NSSDG-LD retains more relevant temporal and structural information, enabling machine learning models trained on it to generalize better when evaluated on real-world data.

The privacy offered by our model could be further enhanced by integrating differential privacy techniques. This would add an extra layer of protection. To improve the utility aspect, a fine tuning of the MCMC based structure learning process can be

performed by i.e. initializing with informed graph structures and incorporating domain knowledge, such as fixing known clinical dependencies. Similarly, replacing random change point initialization with statistically or clinically guided estimates.

A Non-Stationary Synthetic Data Generator for Longitudinal Data

Prasun Jhajharia
Aalborg University
Aalborg, Denmark
pjhajh23@student.aau.dk

Mirko Grimm
Aalborg University
Aalborg, Denmark
mgrimm23@student.aau.dk

Abstract

Longitudinal datasets, which capture repeated observations of individuals over time, are important in areas such as healthcare, finance, and education. However, the right to privacy as stated in the General Data Protection Regulation (GDPR) limits the direct sharing of such sensitive data. Synthetic data generation offers a key solution by producing synthetic datasets that captures key statistical characteristics of real data while reducing the risk of exposing sensitive information. While models like Bayesian networks are useful for generating synthetic data and are easy to interpret, they're mainly built for static datasets. This means they often fall short when it comes to handling time based patterns.

In this work, we propose a framework for generating synthetic longitudinal categorical data using segment-wise Dynamic Bayesian Networks (DBNs). Our method detects change points in the temporal data to identify non-stationary segments and learns separate DBNs for each segment. These models are then aggregated by extracting common structural patterns, and segment wise behaviors are grouped via clustering. Synthetic sequences are generated by sampling from these clusters, preserving both temporal coherence and structural realism. We tested our approach on both simulated and real-world datasets, including Electronic Health Records (EHRs). While our model shows lower utility compared to our baselines, it offers better privacy protection demonstrating that a modest trade-off in accuracy can lead to significant gains in protecting sensitive information.

1 Introduction

The growing digitization of data in domains like healthcare, finance, and education has created new opportunities for data driven research and innovation. In particular longitudinal datasets, which track individuals or systems over a period, are valuable for uncovering progression, behavior, and causality [42]. However, in certain fields it is often difficult for analysts and researchers to get access to high quality data for research purposes, as Data protection regulations, such as the GDPR [10] in the European Union and the Health Insurance Portability and Accountability Act (HIPAA) [1, 43] in the United States, increasingly constrain access. These regulations limit how data from individuals can be shared, especially when it involves sensitive health, behavioral, or financial information. Even de-identified data can pose re-identification risks, where repeated records may reveal unique patterns [8].

To address this problem, synthetic data generators (SDG) have emerged as a promising approach, as studies shows that synthetic data have an lower identity disclosure risk compared to de-identified data [39]. Instead of sharing records of real individuals, organizations can release synthetic generated datasets that mimic the statistical structure of the original data while offering stronger privacy protection[33]. Synthetic data allows researchers and developers to test models, perform exploratory analysis, and develop decision support systems without accessing sensitive data[5]. However, generating synthetic data that are both realistic and privacy-preserving is technically challenging, particularly in longitudinal settings where time dependent relationships exist.

Different SDGs have been developed and vary in their method to create synthetic data. Deep generative models such as Generative Adversarial Networks [14] create synthetic data by training a generator to produce realistic samples that can fool a discriminator, which learns to distinguish between real and fake data. Variational Autoencoders [13] offer high realism by learning complex joint distributions. However, these models often operate as black boxes, making their internal decision-making processes difficult to interpret[34, 48]. This lack of transparency poses challenges when validating them in regulated domains such as healthcare and finance, where accountability and explainability are essential[38]. In contrast, marginal based probabilistic models, such as Bayesian Networks, aim to preserve variable level distributions and conditional dependencies explicitly [9, 49]. These methods offer greater transparency and control, making them well suited for domains, where regulatory compliance and explainability are essential.

Unfortunately, most of the existing marginal based methods [27, 50] are limited to static, cross-sectional datasets, where all records are assumed to be independent. This assumption fails in longitudinal settings, where a variable's value often depends on its previous states. For example, a cancer patient undergoing chemotherapy may follow a multi cycle treatment plan, receiving drugs on a repeating schedule, e.g. every two weeks, then followed by lab tests to monitor response and toxicity. Lab results, such as white blood cell

counts, directly affect whether the next cycle should be proceeds as planned or should be delayed. This creates a chain of temporally dependent events involving diagnoses, medications, lab values, and clinical decisions. Ignoring such dependencies during data synthesis can lead to implausible timelines such as a patient receiving continuous chemotherapy without lab monitoring or delays which undermines both data realism and downstream model performance.

A natural way to model temporal dependencies is through DBNs [12, 28], a temporal extension of Bayesian networks. DBNs capture both intra-slice (within-time) and inter-slice (across-time) dependencies in a structured probabilistic framework [28]. They have been widely used in domains such as speech processing, system monitoring, and disease progression modeling [2, 4, 52]. In modeling diabetes progression, an intra-slice dependency might represent the relationship between a patient's glucose level and insulin dose at a given time point, while an inter-slice dependency could capture how today's glucose level depends on the patient's glucose history and insulin usage from previous visits.

A major limitation of most DBN methods is the assumption of stationarity: that the network structure and transition probabilities remain constant over time. But in reality, system undergo non-stationary changes due to interventions, behavioral shifts, or external events. These shifts, known as change points, mark moments when the underlying process changes, like a patient entering a new disease stage or a user adapting to a platform update. Modeling such structural shifts is essential for generating realistic synthetic sequences.

While existing methods like Bayesian online change point detection [22] and non-stationary DBNs can model evolving network structures [16, 37], they are designed for tasks such as forecasting or latent inference, and are not tailored for generating synthetic data.

Bridging this gap requires adapting change point aware temporal modeling to the synthetic generation setting, to ensuring both realism and segment wise dependency modeling. In this study, we adapt the The non-stationary dynamic change point BGe model (cpBGe) by Grzegorczyk and Husmeier [16] to handle categorical data, transforming it into a non-stationary DBN scored with the Bayesian Dirichlet equivalent (BDe) metric. While the original cpBGe assumes linear-Gaussian processes, our variant models discrete time series data using multinomial distributions with Dirichlet priors, making it suitable for domains like healthcare. Our proposed model, the Non-Stationary Synthetic Data Generator for Longitudinal Data (NSSDG-LD), captures non-stationarity through node-specific change points, allowing CPDs to vary over time. Furthermore, our NSSDG-LD generate synthetic categorical time series by simulating sequences of segments, each defined by learned probabilistic patterns and temporal structures. To evaluate the performance of our SDG, we assess both change point detection accuracy and the utility–privacy trade-off across two settings, a collection of synthetic benchmark datasets with known ground truth for network structure and change point locations, allowing precise quantitative validation, and a real-world domain-specific dataset, the MIMIC-IV clinical database [20, 21], which provides a complex environment for assessing how well the generated data preserve utility while protecting patient privacy.

The remainder of this paper is organized as follows: Section 2 provides related work, followed by background information in Section 3. Section 4 details the proposed methodology, while Section 5 outlines the experimental setup. Section 6 presents the findings, which are further discussed in Section 7. Finally, Section 8 offers concluding remarks, and Section 9 explores directions for future work.

2 Related Work

Synthetic data generation has emerged as a key strategy for privacy preserving data analysis in domains where sharing real data is restricted. Research spans across statistical techniques, probabilistic modeling, and advances in deep generative models. Several reviews [30, 32] works have outlined the evolution of SDGs in the healthcare domain in both tabular and sequential settings. Murtaza et al. [30] highlights how the SDG approach is becoming a popular way to share health data while protecting patient privacy. The review shows that methods based on expert knowledge offer strong privacy but often require a manual setup. On the other hand, data driven methods can struggle to handle complex medical records and rely heavily on having access to real data in the first place. A limitation can be found especially for generating detailed patient timelines with multiple health conditions. It is suggested that these methods require additional research. Longitudinal SDGs are further investigated by Perkonoja et al. [32]. They state that only few of the reviewed SDGs managed to handle all their outlined challenges, e.g. preserving temporal structure, handling unbalanced and irregular data and combining static and time varying variables. In addition, they noted that none of them included built in privacy protection, and most relied on deep learning models. Given these gaps, there is a growing need for more research and models that can explicitly represent temporal dependencies. The allowance of modular control over structural assumptions and interpretable synthetic generation method are further suggested.

DBNs offer a compelling alternative in this regard. Unlike black-box models, DBNs can encode both causal and temporal relationships in a transparent manner, making them suitable for simulating patient trajectories with interpretable temporal logic. Murphy [28] presented foundational work demonstrating DBNs' effectiveness in modeling temporally stable processes. In their classical formulation, DBNs assume stationary structures and consistent of conditional dependencies across time slices. Building on this, Wang et al. [46]

proposed a method for publishing high-dimensional temporal data under differential privacy using DBNs as the core modeling framework. Their method constructs a DBN by selecting highly correlated attributes via mutual information, then builds a temporal dependency graph and applies differential privacy during the synthetic data generation. However, the approach is centered on privacy-preserving data release and assumes that stationary temporal dependencies between variables remain consistent over time.

To address the limitations of stationary DBNs, several non-stationary extensions have been proposed that allow the network structure or parameters to evolve over time. Robinson and Hartemink [36] introduced a model that permits the entire network structure to shift across different segments of the time series. Similarly, Lèbre [26] developed a framework for global structural changes between segments. In contrast, Grzegorczyk and Husmeier [16] proposed a model for continuous data where the overall structure is fixed, but parameters can vary over time through node-specific change points capturing.

While these approaches effectively extend DBNs to accommodate non-stationarity, they differ in their underlying assumptions and range from the type of data supported to the specifications of change points (global vs. local). Notably, they primarily focus on inference rather than data synthesis. This leaves a clear gap at the intersection of change point modeling and longitudinal synthetic data generation, a space this work aims to address.

3 Background

3.1 Dynamic Bayesian Network

DBNs are probabilistic graphical models, which is the extension of static Bayesian networks [6] by modeling time-dependent processes. Formally, DBN is defined as a pair $(B_0, B_{\rightarrow})$, where B_0 is a Bayesian Network that specifies the prior probability distribution over the variables at the initial time step $t = 0$, and $B_{\rightarrow}$ is a Two-Time Slice Bayesian Network (2-TBN) that specifies the conditional probability distribution governing transitions from one time step to the next. The 2-TBN is represented through a directed acyclic graph (DAG) structured across two consecutive time slices. this makes it useful for analyzing sequential processes such as time series data such as stock price over time, monitoring patients in ICUs [28].

The conditional probability distribution (CPD) for a DBN across time steps t can be expressed as follows [28]:

$$P(X_t \mid X_{t-1}) = \prod_{n=1}^{N} P(X_t^n \mid Pa(X_t^n)) \tag{1}$$

where X_t^n represents the n-th node at time t, which may correspond to a variable in the observed, hidden, or input set at time t. The term $\mathrm{Pa}(X_t^n)$ refers to the set of parent nodes of variable X_t^n in the graphical model, which may be drawn from variables in the current time slice X_t, the previous time slice X_{t-1}, or both, depending on the model's structure.

Importantly, nodes within the first slice of the 2-TBN do not possess associated parameters. In contrast, every node in the second slice of the 2-TBN carries a CPD, defining [28]

$$P(X_t^n \mid \mathrm{Pa}(X_t^n)) \quad \text{for all } t > 1 \tag{2}$$

These CPDs govern the probabilistic transitions for all time steps $t > 1$.

Table 1. Table of Notation

Symbol	Meaning
X	The complete set of variables (nodes) in the DBN, representing observed, hidden, or input variables across time steps.
X_t	Set of all random variables (nodes) at time step t.
X_{t-1}	Set of all variables at the previous time step.
X_t^n	n-th variable (node) at time t. For example, if you're modeling vitals, X_t^3 could represent "medication" at time t.
D	Orginal dataset
N	Total number of variables (nodes) in the network.
m	Number of time points in the time series.
G	Graph structure of the Bayesian network, defining parent-child dependencies among variables.
X_n	The n-th node (or variable).
$X_n(t)$	Value of node n at time t.
θ_n	Parameters for node X_n.
$\pi_n(t-1)$	Values of parent nodes of X_n at time $t-1$.
$D_{n,t}$	The value recorded for the n-th variable at time t.

3.2 Non-stationary continuous dynamic Bayesian networks

Grzegorczyk and Husmeier [16] proposed a non-stationary continuous dynamic Bayesian network model, known as change point Bayesian Gaussian equivalent (cpBGe), which allows parameters to evolve over time while maintaining a fixed network structure.

The cpBGe model introduces non-stationarity through a node specific change point process that partitions time series into segments. Importantly, the parent set of each node (i.e., the graph structure) remains fixed across time, enabling information sharing between segments and reducing overfitting risks associated with short time series [16].

The cpBGe model builds upon the foundation of classical DBNs, which define the likelihood of the data given a fixed graph structure G and parameters θ as follows:

$$P(D \mid G, \theta) = \prod_{n=1}^{N} \prod_{t=2}^{m} P\left(X_n(t) = D_{n,t} \mid \pi_n(t-1) = D_{\pi_n,t-1}, \theta_n\right) \tag{3}$$

This equation [16] reflects the Markovian assumption that each variable X_n at time t depends only on its parent variables π_n at the previous time step $t-1$. For every node and each time point from $t = 2$ onward, it computes the probability of the node's observed value, conditioned on the values of its parents one step before. The likelihood is obtained by multiplying these conditional probabilities across all nodes and all time points. This formulation enables the model to capture temporal dependencies.

However, estimating the parameters θ for every possible graph can be computationally intensive and unreliable, especially when the data is split into short segments. To address this, the cpBGe model avoids direct parameter estimation by using a conjugate normal-Wishart prior [11], which allows the parameters to be integrated out analytically.

$$P(D \mid G) = \int P(D \mid G, \theta) P(\theta \mid G)\, d\theta = \prod_{n=1}^{N} \Psi(D_n^{\pi_n}, G) \tag{4}$$

Here, $\Psi(D_{\pi_n}^n, G)$ denotes the Bayesian Gaussian equivalent (BGe) score, which evaluates how well the data for node X_n, conditioned on its parent set π_n, aligns with the assumptions of a linear Gaussian model defined by the graph structure G.

The term $D_{\pi_n}^n$ refers to the specific subset of the dataset used to compute this score: it includes all observed values of $X_n(t)$ for time steps $t = 2$ to m, together with the corresponding values of its parent nodes $\pi_n(t-1)$ from the immediately preceding time step.

This child-parent pairing across time enables the model to assess the quality of the local conditional dependency implied by the graph. By computing the BGe score for each node individually using this data, the model can determine how well the overall structure G fits the observed time series, without relying on explicit parameter estimates. This scoring mechanism plays a central role in enabling efficient and reliable graph evaluation within the cpBGe framework.

To introduce non-stationarity, the cpBGe model partitions each node's time series into multiple segments, where each segment assumes constant parameters. The number of segments for each node X_n, denoted K_n, and the location of their boundaries are not fixed in advance but are inferred from the data. This results in a segmented model, where the time series of node X_n is divided into K_n contiguous intervals, each governed by its own parameter vector θ_k^n, for segment index $k = 1, \ldots, K_n$.

The segmentation is encoded by a vector V_n, called the segmentation vector, which assigns a segment label to each time point in the time series of node X_n. The entry $V_n(t)$ indicates the specific segment to which time step t belongs. Given this segmentation, the segment count K, and the parameters θ, the full likelihood of the observed data is expressed as:

$$P(D \mid G, V, K, \theta) = \prod_{n=1}^{N} \prod_{t=2}^{m} \prod_{k=1}^{K_n} P\left(X_n(t) \mid \pi_n(t-1), \theta_n^k\right)^{\delta_{V_n(t),k}} \tag{5}$$

Here, $\delta_{V_n(t),k}$ is the Kronecker delta function, which equals 1 if time point t is assigned to segment k (i.e., $V_n(t) = k$), and 0 otherwise. This ensures that the likelihood contribution at each time step is computed using only the parameters specific to the assigned segment. Each segment k for node X_n is associated with its own parameter vector θ_n^k, which contains the regression coefficients and noise variance used to model the linear Gaussian relationship between $X_n(t)$ and its parent nodes $\pi_n(t-1)$. These parameters are assumed to be constant within a segment but allowed to vary across segments to capture non-stationary behavior.

cpBGe constrains segment assignments to occur over contiguous intervals means that all time points belonging to a given segment must appear in consecutive order on the time axis. This preserves the temporal structure of the data and avoids fragmented or disjointed segmentations. The number of segments K_n for each node is not fixed; instead, it is treated as a random variable governed by a truncated Poisson prior, which biases the model toward simpler segmentations unless the data provides strong evidence for additional changepoints.

As in the stationary case, the model integrates out the segment-specific parameters using the BGe score, yielding the marginal likelihood:

$$P(D \mid G, V, K) = \prod_{n=1}^{N} \prod_{k=1}^{K_n} \Psi(D_n^{\pi_n}[k, V_n], G) \tag{6}$$

Here, $\Psi(D_n^{\pi_n}[k, V_n], G)$ represents the marginal likelihood contribution of the k-th segment of node n, conditioned on its parent set π_n in the graph G. This term is computed using only the subset of the data assigned to the k-th segment of node n, as indicated by the segmentation vector V_n. Formally, $D_n^{\pi_n}[k, V_n]$ denotes the collection of data points $(X_n(t), \pi_n(t-1))$ such that $V_n(t) = k$, meaning time point t is assigned to segment k for node n. This formulation allows exact evaluation of model evidence for a given graph and segmentation under the change-point model, since each segment corresponds to an independent linear-Gaussian model integrated out analytically via the BGe score.

The full posterior distribution over the graph structure G, the segmentation variables V, and the number of segments K is expressed as:

$$P(G, V, K \mid D) \propto P(D \mid G, V, K) \cdot P(G) \cdot P(V \mid K) \cdot P(K) \tag{7}$$

The likelihood term $P(D \mid G, V, K)$ evaluates how well the data fits a model with fixed structure and node-specific changepoints, using the BGe score to integrate out parameters. The prior $P(G)$ assumes a uniform distribution over acyclic graphs with a parent limit to encourage sparsity. $P(K)$ assigns a truncated Poisson prior on the number of segments, favoring simpler models. The term $P(V \mid K)$ is derived from a change-point process that imposes temporal smoothness by probabilistically placing changepoints using ordered uniform samples.

Inference in the cpBGe model is conducted using a Reversible Jump Markov Chain Monte Carlo (RJMCMC)[15] algorithm, which allows the model to explore a space of variable dimension (due to the changing number of segments). At each iteration, the algorithm randomly chooses whether to update the graph structure or the segmentation for a randomly selected node. In the structure update step, a new graph G' is proposed by adding or deleting a single edge. The proposed graph is accepted with the Metropolis-Hastings probability:

$$A(G' \mid G) = \min\left(1, \frac{P(D \mid G', V, K)}{P(D \mid G, V, K)} \cdot \frac{P(G')}{P(G)} \cdot \frac{|\mathcal{N}(G)|}{|\mathcal{N}(G')|}\right) \tag{8}$$

where $\mathcal{N}(G)$ denotes the neighborhood of G under single-edge changes, which refers to all graphs reachable from G by addition or deletion single edge. $P(G)$ represents the prior probability of the current graph structure and $P(G')$ represents the prior probability of the proposed graph structure.

In the segmentation update step, the algorithm proposes one of three moves: a birth move (adding a change point), a death move (removing one), or a reallocation move (shifting a boundary between two segments). These moves alter the segmentation vector V_n and the number of segments K_n for node n. The proposed move is accepted with probability:

$$A = \min\Big(1, \frac{\prod_{k=1}^{K_n'} \Psi(D_n^{\pi_n}[k, V_n'], G)}{\prod_{k=1}^{K_n} \Psi(D_n^{\pi_n}[k, V_n], G)} \cdot \frac{P(V_n' \mid K_n') \cdot P(K_n')}{P(V_n \mid K_n) \cdot P(K_n)} \cdot \frac{q(\text{reverse})}{q(\text{forward})}\Big) \tag{9}$$

where $q(\cdot)$ denotes the proposal distributions for the respective moves. These updates allow the model to adaptively find the most probable network and segmentation structure consistent with the observed data.

The cpBGe model thus provides a principled framework for modeling non-stationary time series where temporal variability is captured through parameter changes rather than structural changes. By combining the analytical tractability of the BGe score with the flexibility of Bayesian change point modeling and RJMCMC inference, the cpBGe framework effectively balances model complexity and generalizability. This makes it particularly well suited for applications involving biological time series, where dynamic changes occur, but the underlying causal relationships are expected to remain structurally consistent over time.

4 Methodology

This section outlines our approach for modeling non-stationary time-series data using a change point aware DBN, drawing foundational inspiration from the cpBGe model proposed by Grzegorczyk and Husmeier [16], as described in Section 3. EHRs often contain a mix of numerical and categorical information, i.e. the MIMIC IV dataset [20, 21]. Only a limited number of numerical columns were suitable for meaningful longitudinal analysis, whereas categorical features were more consistently available. This made categorical data a more practical choice for evaluating the model on real patient. These observations motivated us to use the cpBGe model as the foundation of our method. By using its change point aware learning framework as inspiration, we modified the model to handle categorical time series data, enabling us to better capture temporal dependencies and structural changes in real world clinical records.

The following subsections describes the architecture and inner workings of the our proposed NSSDG-LD model. It is split into change point/structure inference and synthetic data generation.

4.1 Change point and structure inference

The first step is to separate an EHR into multiple dataset, where each dataset contains one Patient. As illustrated in Figure 1, the NSSDG-LD algorithm begins by taking a multivariate time series dataset, $D \in \mathbb{R}^{N \times m}$ as input, where N is the number of variables and m is the number of time points. Each time series is first discretized into a fixed number of categories using a binning strategy such as quantile based binning, resulting in a categorical dataset $D \in \{0, 1, \ldots, C-1\}^{N \times m}$, where C is the number of discrete states per variable. Temporal dependencies are captured by modeling the conditional distribution $P(X_n(t+1) \mid \pi_n(t))$, where π_n denotes the parent set of node X_n. After prepossessing, the algorithm randomly initializes a Bayesian network structure G and a set of node-specific change points that partition each node's timeline into K_n stationary segments. This change point framework, including the use of node-specific segmentations and posterior scoring, follows the cpBGe model, but is adapted here for categorical data. In each iteration of the MCMC sampling loop, the algorithm proposes one of two moves: a graph structure update (adding, removing, or reversing an edge in G) or a changepoint update (birth, death, or reallocation of a changepoint for a given node). For each proposed configuration, the model fit is evaluated using the

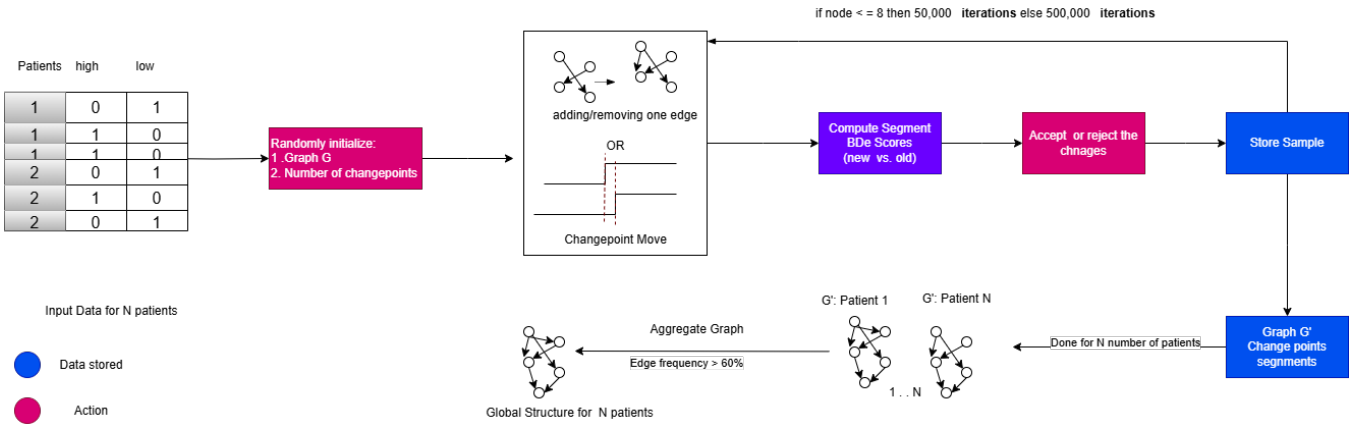

Figure 1. NSSDG-LD Change point and structure inference

Bayesian Dirichlet equivalent (BDe)[29] score [19], which computes the marginal likelihood over categorical data under a Dirichlet-multinomial model. In [35] they used BDe score for categorical data. Let X_c be a child node with categorical outcomes $\{0, \ldots, C-1\}$, and let the data be grouped by unique parent configurations $j = 1, \ldots, q$. For each configuration j, the log marginal likelihood is computed as [29, 47] :

$$P(D_j \mid \boldsymbol{\alpha}) = \frac{\Gamma\left(N_j + \sum_{c=1}^{C} \alpha_c\right)}{\Gamma\left(\sum_{c=1}^{C} \alpha_c\right)} \cdot \prod_{c=1}^{C} \frac{\Gamma(\alpha_c)}{\Gamma(N_{jc} + \alpha_c)} \tag{10}$$

where:

- N_{jc} is the count of observations where the child takes value c given parent configuration j,
- $N_j = \sum_{c=1}^{C} N_{jc}$ is the total count for configuration j,
- $\boldsymbol{\alpha}$ is a Dirichlet prior over child categories.

The full local score for a child node X_c with parent set $\pi(X_c)$ is obtained by summing the log marginal likelihoods across all q parent configurations:

$$\log P(D \mid \pi(X_c)) = \sum_{j=1}^{q} \log P(D_j \mid \boldsymbol{\alpha}). \tag{11}$$

In the presence of changepoints, each node's timeline is partitioned into segments within which the conditional distribution is assumed stationary. The BDe score is computed separately for each segment using the segment-specific data and corresponding parent configurations. These scores are then aggregated across all segments and nodes to obtain the total marginal likelihood The total likelihood score, together with the prior and proposal probabilities, is used to compute the Metropolis-Hastings acceptance ratio. If the proposed move is accepted, the updated structure and segmentation are added to the posterior sample. This MCMC sampling is then repeated for N iterations. After convergence, a posterior summary is generated by examining the sampled graphs and segmentation.he algorithm then selects the maximum a posteriori (MAP) sample from the patient's MCMC run as the final representation, which includes the graph structure G^{MAP}, segmentation vectors V_n^{MAP} and segment counts K_n^{MAP}, for each node X_n. This entire process is then repeated for each individuals patient's longitudinal record. To construct a population-level consensus graph, directed edges that appear in at least 60% [1] of patient-level graphs are retained. This threshold balances sensitivity (capturing common structural patterns across patients) and specificity (excluding patient-specific or noise-driven edges). Such an approach is particularly useful in settings with limited sample sizes, missing data, or sparsely connected networks, where relying on a single global model may fail to capture inter-individual variation [45]. A different method, selecting the highest-scoring structure from all individual, can introduce variability, potentially reflecting noise or unstable trends in the data rather than true underlying dependencies.

4.2 Synthetic Data Generation Expansion

Following the consensus graph construction, the NSSDG-LD framework proceeds with the synthetic data generation, as can be seen in figure 2, by extracting all segmentation vectors V_n^{MAP} and segment counts K_n^{MAP}, for each node X_n from each patient's MAP sample.

For each segment k, the algorithm estimates a Conditional Probability Distribution (CPD), if a node X_n has no parents, the CPD is the empirical marginal distribution:

$$P(X_n = c) = \frac{\text{count}(X_n = c)}{\text{segment length}}$$

[1] "The 60% threshold reflects a majority agreement across patients, allowing the consensus graph to emphasize consistent dependencies while tolerating minor individual differences"

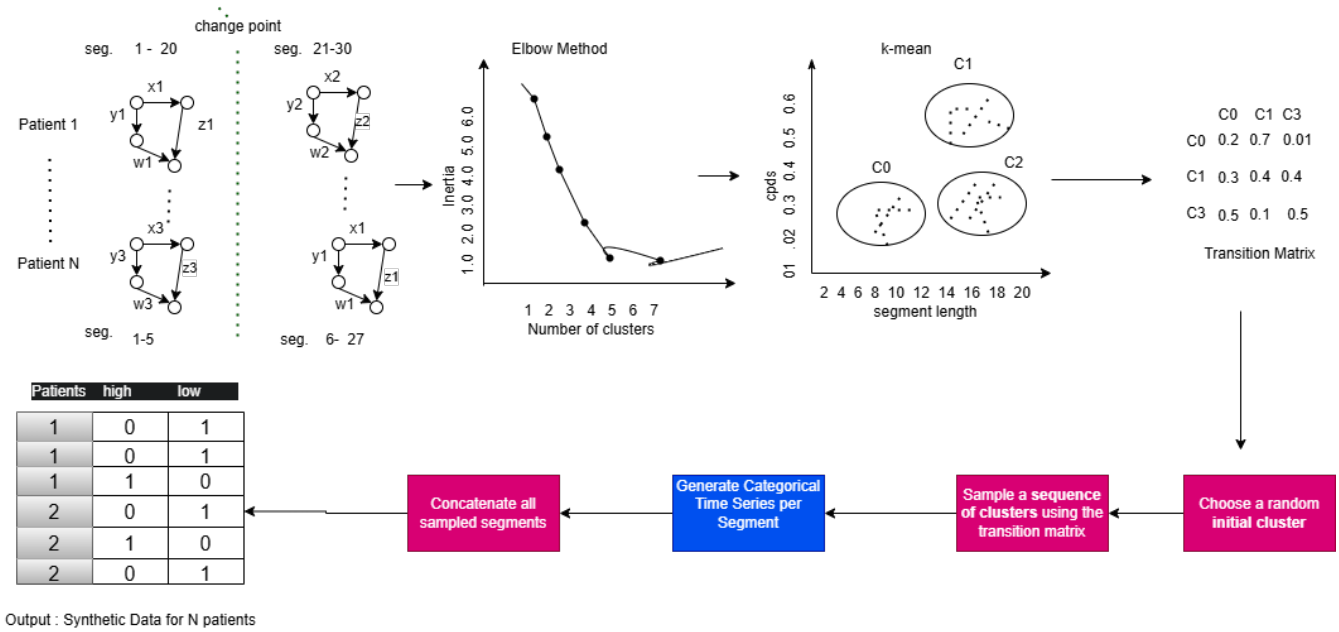

Figure 2. NSSDG-LD Synthetic data generation

If X_n has parents π_n, the CPD is modeled as a Dirichlet-multinomial, computed via normalized joint frequency tables:

$$P(X_n \mid \pi_n) = \frac{N_{j,c}}{\sum_{c'} N_{j,c'}}$$

where $N_{j,c}$ counts how often $X_n = c$ occurs under parent configuration j. To preserve segment duration, the corresponding segment length ℓ_k is appended, forming an augmented vector:

$$\mathbf{z}_k = [\mathrm{vec}(P_k),\ \ell_k]$$

To reduce the redundancy of highly similar CPDs, we perform clustering over the CPD vectors $\mathbf{z}_k$. If we didn't cluster the CPDs, each segment would end up with its own unique distribution. This would make the synthetic data almost a one-to-one copy of the original, which defeats the purpose of generating new, generalizable data. Clustering helps us group similar segment behaviors together, so instead of memorizing and repeating exact patterns, the model learns broader types of behavior. This makes the synthetic data both more efficient to generate and better at preserving privacy. The clustering process is conducted using cosine similarity, implemented via L2-normalized K-Means clustering using the scikit-learn library [31, 40]. Segment length ℓ_k is treated as an additional feature with a configurable weight. The number of clusters K is chosen using the elbow method, by identifying the inflection point in the within-cluster sum of squares (inertia). Furthermore, the algorithm records the empirical start distribution over clusters and a transition matrix $T \in \mathbb{R}^{K \times K}$ estimating the distribution of cluster-to-cluster transitions.

Finally synthetic categorical time-series data were generated using the learned generative model. Simulation began by sampling an initial cluster from the empirical start distribution and the segment length was drawn from the empirical distribution of segment durations within the selected cluster. For each time step within a segment, the value of each variable was simulated sequentially using the corresponding CPD and the values of its parent variables from the previous time step, thereby preserving the conditional dependencies encoded in the DBN structure. At the end of each segment, the next cluster was selected according to the learned transition probability matrix, and the simulation continued. This process was repeated until the synthetic sequence reached the desired length.

5 Experimental Setup

This section explains the evaluation of the NSSDG-LD algorithm. It covers the datasets we used, how we measured performance, and the overall steps in our experimental process. Our goal was to evaluate how well the model captures both structural and temporal patterns, as well the utility and privacy of the generated data.

5.1 Datasets

5.1.1 Two-Node. The two-node dataset represents the simplest synthetic network considered and is reproduced from the cpGBe paper [16]. As illustrated in Figure 3, the structure consists of a self-loop on X ($X \rightarrow X$), which induces autocorrelation in the time series of X, and a directed edge from X to Y, introducing a dependency of Y on X. The

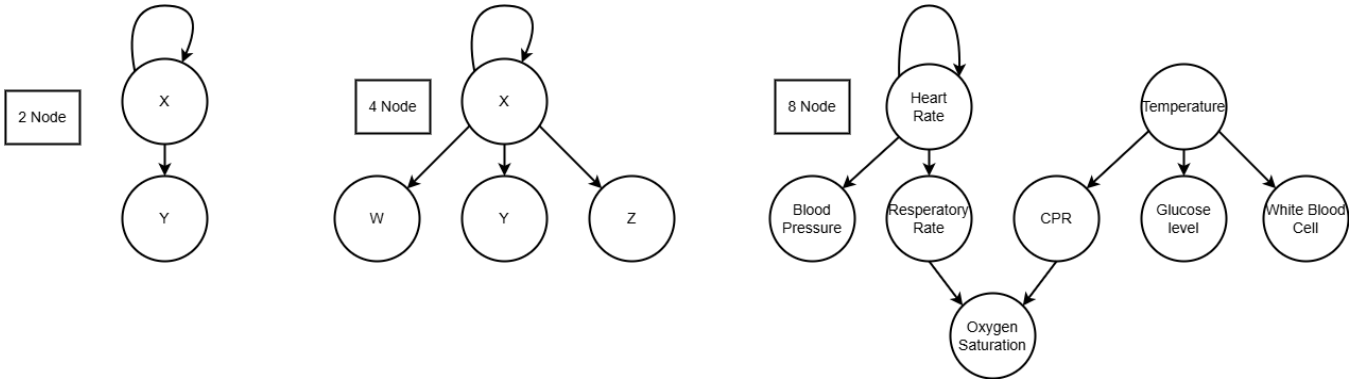

Figure 3. Network Structures

influence from X to Y is governed by a piecewise linear process with a time-dependent coefficient $\beta(t)$. The generative model is defined as:

$$X(t+1) = \sqrt{1-\varepsilon^2} \cdot X(t) + \varepsilon \cdot \phi_X(t+1), \tag{12}$$

$$Y(t+1) = \beta(t) \cdot X(t) + c \cdot \phi_Y(t+1), \tag{13}$$

where:

- $\varepsilon \in [0, 1]$ controls the autocorrelation in X,
- $\phi_X(t), \phi_Y(t) \sim \mathcal{N}(0, 1)$ are independent Gaussian noise terms,
- $\beta(t)$ is a piecewise constant coefficient.
- c is chosen to satisfy a desired signal-to-noise ratio (SNR):

$$c = \frac{\hat{\sigma}(\beta(t)X(t))}{\text{SNR}},$$

where $\hat{\sigma}(\beta(t)X(t))$ is the empirical standard deviation estimated from noise-free simulations.

5.1.2 Four-Node. The four-node dataset extends the previous design to assess the model's performance under more complex multivariate interactions. Figure 3 shows that the network comprises nodes X, Y, W, and Z with three edges and one selfloop. The data is generated according to:

$$X(t+1) = \sqrt{1-\varepsilon^2} \cdot X(t) + \varepsilon \cdot \phi_X(t+1), \tag{14}$$

$$Y(t+1) = \beta_Y(t) \cdot X(t) + c_Y \cdot \phi_Y(t+1), \tag{15}$$

$$W(t+1) = \beta_W(t) \cdot X(t) + c_W \cdot \phi_W(t+1), \tag{16}$$

$$Z(t+1) = \beta_Z(t) \cdot X(t) + c_Z \cdot \phi_Z(t+1), \tag{17}$$

5.1.3 Eight-Node. Furthermore, another extension of previous datasets is represented in this eight node dataset, inspired by real-world healthcare indicators. The variables include: HeartRate, BloodPressure, RespiratoryRate, OxygenSaturation, Temperature, WhiteBloodCell, CRP, and GlucoseLevel. HeartRate acting as a root node following an autoregressive process:

$$X_{\text{HeartRate}}(t+1) = \sqrt{1-\varepsilon^2} \cdot X_{\text{HeartRate}}(t) + \varepsilon \cdot \phi(t),$$

The remaining nodes are generated through linear dependencies on their parent nodes, see Figure 3, perturbed by independent Gaussian noise scaled by a fixed factor.

5.1.4 MIMIC IV. For our experiments, we derived a categorical time-series dataset from the Medical Information Mart for Intensive Care (MIMIC-IV) database [2]. The dataset was structured such that each patient instance consists of at least 15 hospital visits, and each visit is described using 10 categorical attributes 3. These attributes were extracted from tables such as microbiologyevents, admissions, transfers, and other clinical event logs available in MIMIC-IV. Categorical variables encode clinical activities and outcomes, such as tests performed, specimen types, lab results, care units, admission type, and test interpretations. A snippet of the dataset is shown in the Appendix (see table 3) To illustrate the data encoding process, consider an example involving a blood culture test. When such a test is requested, a blood sample is collected from the patient and sent to the microbiology lab. The spec_type_desc indicates the specimen type (e.g., "blood"). If no bacterial growth is observed in the sample, the remaining result columns are recorded as NULL. However, if bacteria are cultured, the org_name column records each detected organism—resulting in multiple rows for the same specimen. If antibiotic susceptibility is tested for the identified organisms, each tested antibiotic is listed in the ab_name column, with associated sensitivity metrics such as dilution_text, dilution_value, and interpretation. This structure inherently creates a categorical multi-row representation per test-visit episode, which we encoded using a visit-wise transformation strategy into fixed-length categorical features. Additionally, we computed the length of hospital stay per visit using admission and discharge times, measured in hours. For classification purposes, each visit was assigned a binary label:

- Assign label 1 ("Serious") if the visit duration exceeds 72 hours and not discharged to the "Home" or died.

[2] https://physionet.org/content/mimiciv/3.1/hosp/#files-panel

- Assign label 0 ("Not Serious") .

5.2 Metrics

5.2.1 Structure Learning Evaluation. To evaluate the learned dynamic structure, we compute the Area Under the ROC Curve (AUC) between the predicted edge probabilities and the ground truth adjacency matrix. For each directed edge, the model estimates the posterior edge probability across MCMC samples. These probabilities are flattened and compared with the binary ground truth. Providing a threshold independent measure of structure recovery [41].

5.2.2 Change point Detection Evaluation. We assess the accuracy of change point detection using the F1 Score. True and predicted change points are matched within a tolerance window of the series length. A match is considered a true positive if a predicted change point falls within the window of a true change point. Based on these matches, we compute precision, recall, and F1 score:

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN}, \quad \text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$$

5.2.3 Marginal Distribution Similarity . To assess how well synthetic data preserves the distribution of categorical variables, we use the TVComplement metric [3]. This metric compares the marginal (1D) distributions of each column in the real and synthetic datasets by computing the Total Variation Distance (TVD). Given the set of all possible category values Ω, the TVD is defined as:

$$TVD(D, S) = 1 - (\frac{1}{2} \sum_{c \in \Omega} |D_c - S_c)$$

, where c describes all the possible categories in a domain of a column, Ω. D and S refer to the real and synthetic frequencies for those categories.

Another method to evaluate the statistical similarity between real and synthetic datasets is the *Kullback–Leibler (KL) divergence*. KL divergence quantifies the dissimilarity between two probability distributions. For each categorical variable, we compute the KL divergence between the empirical marginal distributions of the real and synthetic data, defined as:

$$\text{KL}(D \parallel S) = \sum_{c \in \Omega} D_c \log\left(\frac{D_c}{S_c}\right), \quad \text{with } D_c > 0 \Rightarrow S_c > 0.$$

Here, D_c and S_c denote the relative frequencies of category c in the real and synthetic datasets, respectively. We report the average inverse KL divergence across all columns [3, 24].

5.2.4 Pairwise Distribution Similarity for Categorical Variables. To assess the degree to which synthetic data replicates the joint distribution of categorical variables observed in real data, we utilize a pairwise contingency similarity metric [4]. For any pair of categorical attributes in columns A and B, we compute normalized contingency tables for both the real and synthetic datasets. Each table entry (c_A, c_B) corresponds to the proportion of records in which the combination of categories $c_A \in A$ and $c_B \in B$ occurs. Let D_{c_A,c_B} represent the normalized frequency of the category pair (c_A, c_B) in the real dataset, and let S_{c_A,c_B} denote the corresponding frequency in the synthetic dataset. The similarity score is then computed using the Total Variation Distance between these two joint distributions:

$$\text{score} = 1 - \frac{1}{2} \sum_{c_A \in A} \sum_{c_B \in B} \left| S_{c_A,c_B} - D_{c_A,c_B} \right|$$

5.2.5 Cross-Time Mutual Information Gap. To evaluate whether temporal dependencies are preserved in the synthetic data, we compute mutual information (MI) between each variable and its true parent(s) from the previous time step, as defined by the DBN structure. Let $\text{Pa}(X_i^t) \subseteq \{X_1^{t-1}, \ldots, X_n^{t-1}\}$ be the set of parents of X_i^t. Then, the Cross-Time Mutual Information Gap (CTMIG) is defined as:

$$\text{CTMIG} = \frac{1}{n} \sum_{i=1}^{n} \left| \text{MI}_{\text{real}}(X_i^t, \text{Pa}(X_i^t)) - \text{MI}_{\text{synth}}(X_i^t, \text{Pa}(X_i^t)) \right|.$$

This metric provides a more faithful evaluation of dependency preservation in synthetic data, especially when variables are driven by their parents. It aligns closely with the structural assumptions of Dynamic Bayesian Networks and helps assess temporal realism in generated sequences [23].

5.2.6 Classification Performance (AUROC):. We use the synthetic dataset to train an XGBoost classifier and evaluate its performance by making predictions on the original data. The performance is reported using the Area Under the Receiver Operating Characteristic Curve (AUROC). This metric provides us the classifier's ability to distinguish between classes.

5.2.7 Membership Inference Attack (MIA). To evaluate potential privacy leakage from synthetic data, we employ a membership inference assessment [25]. This approach estimates whether individual records from the original training dataset can be distinguished from non-training records. A random forest classifier is trained to differentiate between synthetic samples and unseen real data drawn from the same distribution. Importantly, the simulated adversary operates under the assumption of no access to the generation process, but full access to both the synthetic dataset and a representative real-world sample.[5]

[3]Total Variation Distance. Last access: Mai 2025. https://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/tvcomplement

[4]Contingency Similarity. Last access: Mai 2025. https://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/contingencysimilarity

[5]2025. MIA. https://github.com/schneiderkamplab/syntheval/tree/main

5.3 PARSynthesizer

It is implemented as the PAR[51] model in the SDV framework, uses a neural network to generate synthetic sequential data. It applies a GRU-based architecture to model dependencies across time steps. Like many deep learning models, the neural network itself is treated as a black box. However, what makes PAR interpretable is that the output of the network is structured, instead of directly generating the next value, it predicts parameters of probability distributions, such as the mean and variance of a Gaussian for continuous variables, or category probabilities for categorical ones. These parameters are then used to sample the next value in the sequence.

5.4 Evaluation Framework

Our framework is evaluated in three distinct phases, each targeting a specific aspect of performance and applicability:

5.4.1 Phase 1: Implementation and Validation. We first implement and validate our models using three synthetic datasets with known ground-truth structures and change points (Section 5.1). In this phase, we compare our implementation of the cpBGe model against our proposed NSSDG-LD framework. Structural recovery is evaluated using the AUC metric, while change point detection performance is assessed using the F1 score (Section 5.2).

To ensure comparability, continuous variables in the NSSDG-LD model are discretized using quantile-based binning. We optimized the binning size with the information bottleneck algorithm [17]. This transformation aligns the data format with the requirements of our categorical modeling framework and ensures a fair comparison by using the same underlying generative process for both models. While binning introduces some loss of information, it reflects practical constraints in real-world categorical settings.

Synthetic patients are divided into five groups based on change point configurations, simulating various temporal segmentation patterns, as detailed in Table 2.

Table 2. Groups with Varying Temporal Segmentation

Group	Time Points (m)	Patients	Change points
1	50–500	50–500	$\left[\frac{m}{2}\right]$
2	50–500	50–500	$\left[\frac{4*m}{5}\right]$
3	50–500	50–500	$\left[\frac{m}{3}, \frac{2*m}{3}\right]$
4	50–500	50–500	$\left[\frac{m}{4}, \frac{m}{2}, \frac{3*m}{4}\right]$
5	50–500	50–500	$\left[\frac{m}{5}, \frac{2*m}{5}, \frac{3*m}{5}, \frac{4*m}{5}\right]$

For each experimental run, we vary both the number of patients and the number of time points between 50 and 500 to evaluate scalability and sensitivity to sequence length.

5.4.2 Phase 2: Synthetic Data Utility Evaluation. In the second phase, we reuse the node-based datasets to evaluate the quality of synthetic data generated by NSSDG-LD. A comprehensive set of utility metrics (Section 5.2) is used to assess distributional similarity and dependency preservation. Based on the results from Phase 1, we select optimal dataset configurations (patient count and time points) for efficient benchmarking. We additionally compare performance against the PARSynthesizer as a baseline model.

5.4.3 Phase 3: Real-World Generalization. To evaluate the practical utility of our method, we apply NSSDG-LD to the MIMIC-IV dataset. This step tests the framework under real-world conditions, characterized by noise, irregular sampling, and high-dimensional categorical time series, which are common in healthcare data. In addition to the utility metrics, we incorporate MIA and a classification task to assess downstream usefulness and privacy preservation. The classification task involves predicting whether a patient is categorized as "serious" or "not serious" based on predefined labels, using features such as the length of hospital stay and discharge location. The model is trained entirely on synthetic MIMIC data generated using the NSSDG-LD framework and evaluated on the original MIMIC data. This setup enables us to test whether the synthetic data captures clinically meaningful patterns. Importantly, the classification does not rely solely on the current patient state but also incorporates historical information from previous visits or time steps, reflecting the temporal dependencies often present in real clinical decision making.

6 Experimentation

6.1 Phase 1

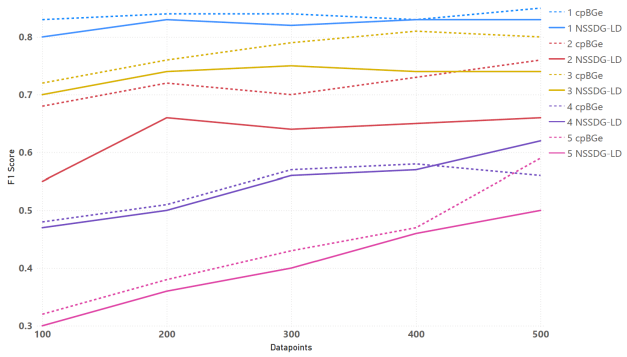

Figure 4. Change point evaluation on the 2-Node dataset

Across all groups, we observe in Figure 4 a consistent upward trend in the F1 scores with increasing dataset size up to 500 data points for both modeling approaches, indicating a improved detection performance with more data points per patients. However, after certain increase of data points

a saturation takes place. An exception for this can be seen for Group 4 and 5, where the score continuous increases. Overall, the continuous model slightly outperformed the categorical model, particularly in more complex scenarios, where multiple nodes and dependencies are in place. This shows clearly that performances of algorithms start at a lower score for Group 4 and 5.

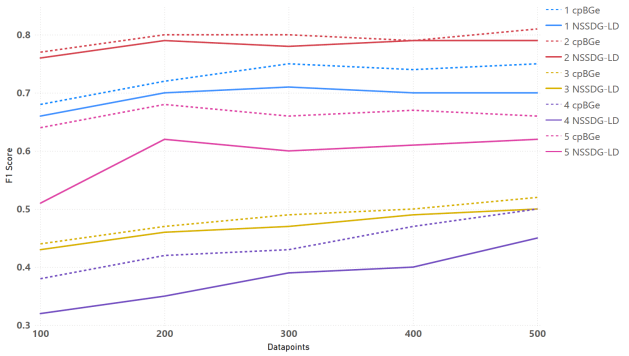

Figure 5. Change point evaluation on the 4-Node dataset

Group 1 in Figure 5, the simplest scenario with a single change point, yielded the highest F1 scores for both models ranging from 0.77 to 0.81 (cpBGe) and 0.76 to 0.79 (NSSDG-LD). This suggests that both models effectively capture simple temporal dynamics, especially with larger datasets. Group 5, which featured the most complex configuration with four change points, recorded the lowest F1 scores overall. Starting at 0.38 (cpBGe) and 0.32 (NSSDG-LD) for datasets with 100 data points,the performance increased modestly to 0.5 (cpBGe) and 0.45 (NSSDG-LD) at 500 data points. This highlights the challenges posed by increased segmentation complexity, potentially due to noisier boundaries or less distinguishable transitions. Overall the performance of the algorithm depends on the complexity of the change points and the structure of the dataset. The 2-Node dataset, seen in Figure 4 achieved the highest scores and the 8-Node dataset in Figure 13 the lowest.

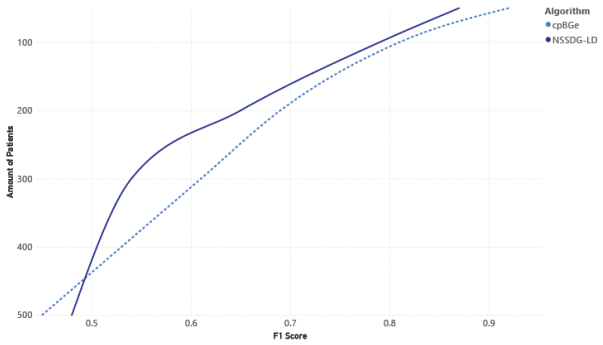

Figure 6. Combined Change point evaluation of all three datasets with different amounts of patient

In contrast to increasing the data points per patient, we run as well an experiment where data points stay at constant amount and the number of patients increases. Figure 6 shows that the opposite appears. With increasing patients the F1 score drops rapidly. This shows a clear correlation between these two factors.

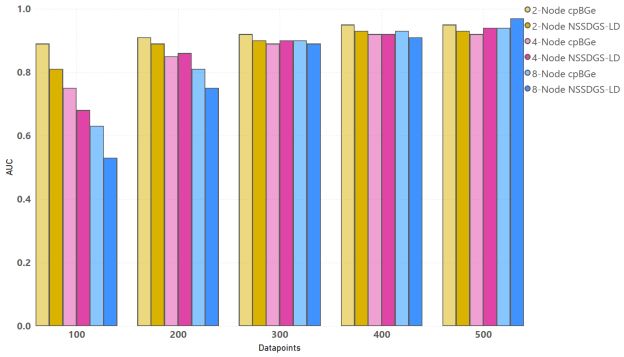

Figure 7. Structure evaluation on all three datasets

Furthermore, we evaluate the structural recovery performance of our proposed algorithms under both continuous and categorical settings across synthetic datasets of varying complexity. Performance is measured using the Area Under the ROC Curve (AUC), which reflects the model's ability to accurately distinguish true structural edges from false ones. Across all configurations, a clear trend emerges in Figure 7 : increasing the number of data points consistently improves structure recovery, regardless of the algorithm type or network size. For the 2-node datasets, both cpBGe and NSSDG-LD achieve clearly high AUC scores (Score = 0.9) with as few as 200–300 data points, and reach above 0.9 at 300 data points for the continuous case and 400 for the categorical one. In the 4-node setting, we observe a mild performance gap at smaller sample sizes. CpBGe yields an AUC of 0.75 at 100 points, improving to above 0.9 by 300 data

points. NSSDG-LD starts slightly lower (0.68 at 100 points), but quickly closes the gap, achieving above 0.9 at 400 data points. The 8-node datasets highlight the growing challenge of structural recovery in higher-dimensional settings. While both models exhibit upward AUC trends with increasing data, they no longer reach perfect recovery within the 500-point limit. The continuous algorithm improves from 0.63 (100 points) to 0.94 (500 points), whereas the categorical counterpart ranges from 0.53 to 0.97 over the same span. This gap underscores the higher data demands and increased complexity of categorical inference in large graphs. Nevertheless, both models achieve AUC > 0.9 with 400 or more points, indicating strong structure learning capacity in data rich regimes.

6.2 Synthetic Data Evaluation

This section presents the performance comparison between the NSSDG-LD and PAR algorithms across all three datasets with known ground. Performance was evaluated using four utility metrics on the created synthetic data.

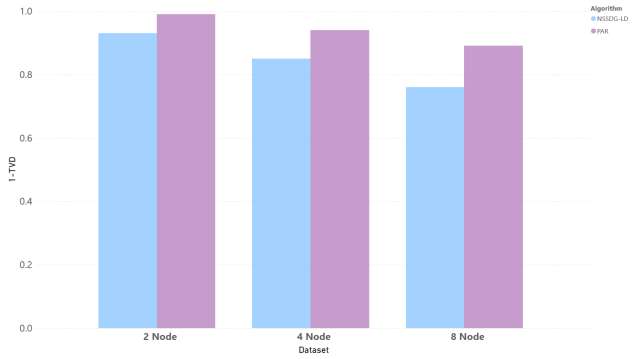

Figure 8. 1-Way TVD Evaluation on all three datasets.

On the 2-node dataset, both NSSDG-LD and PAR perform comparably well across all metrics, though PAR consistently achieves slightly higher values. For instance, PAR yields a 1-TVD of 0.99,seen in figure 8, slightly higher than NSSDG-LD's 0.93, suggesting it more accurately captures marginal distributions. Similarly in figure 10, 9 and 11, PAR maintains slightly superior performance in 2-TVD for 2 Node dataset (0.83 vs 0.81), KL (0.98 vs 0.97), and MI (0.92 vs 0.97). In contrast, NSSDG-LD's performs slightly better in 2-TVD for the 4 and 8 Node dataset.

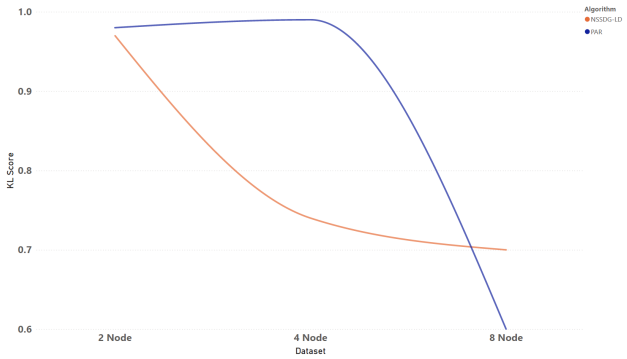

Figure 9. KL evaluation on all three datasets.

As the number of nodes increases, a separation in the algorithm robustness emerges. NSSDG-LD's performance declines more noticeably, especially on KL and MI metrics. The KL score drops from 0.97 to 0.74, and MI drops from 0.97 to 0.73, indicating a degradation in its ability to capture joint distributions and preserve the overall statistical utility of the data. Meanwhile, PAR sustains high performance, showing minimal loss: KL remains nearly perfect at 0.99, and MI holds at 0.92. This suggests that PAR is more resilient to increases in network complexity. Even in 1-TVD and 2-TVD, PAR continues to keep the higher scores (1-TVD: 0.94 vs 0.85, 2-TVD: 0.61 vs 0.63), though the margin is slightly smaller in the latter, indicating that both models struggle more with capturing higher-order dependencies.

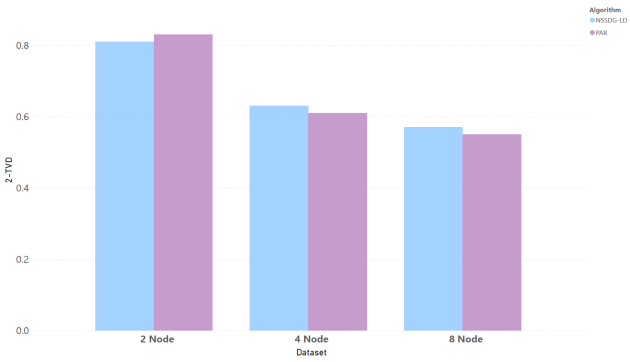

Figure 10. 2-Way TVD evaluation on all three datasets.

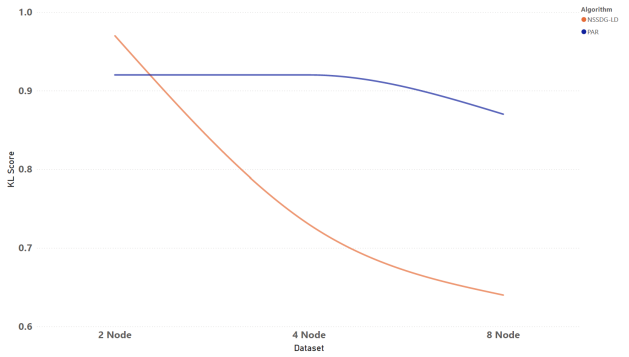

Figure 11. MI evaluation on all three datasets.

In the 8-node scenario, performance degradation becomes evident for both algorithms. NSSDG-LD registers the lowest scores across the board: 1-TVD drops to 0.76, KL falls to 0.70, and MI dips to 0.64. PAR, on the other hand, maintains relatively high 1-TVD (0.89) and MI (0.87), though it does experience a decline in KL (0.60) and 2-TVD (0.55). This suggests that while PAR is affected by increased complexity, it continues to generalize better and generate higher-quality synthetic data than NSSDG-LD. Interestingly, the gap between 2-TVD scores in figure 10 of NSSDG-LD and PAR narrows progressively as complexity increases (difference of 0.02 at 4-node and only 0.02 again at 8-node), potentially indicating a saturation effect where both models struggle equally in modeling two-step temporal or structural relationships in larger graphs.

6.3 Real-World Generalization

Algorithm	1-TVD	2-TVD	KL	MU	MIA	AUROC
NSSDG-LD	0.71	0.61	0.49	0.56	0.36	0.59
PAR	0.84	0.64	0.47	0.79	0.47	0.43

Figure 12. Utility and Privacy evaluation on generated synthetic data.

The table 12 presents a comparative evaluation of the two synthetic data generation algorithms, PAR and NSSDG-LD, on a real world based EHR across six key metrics that collectively assess both privacy protection and data utility. From a privacy perspective, NSSDG-LD demonstrates a clear advantage. It achieves a lower MIA score of 0.36 compared to 0.47 for PAR, indicating a reduced risk of privacy leakage. This suggests that data generated by NSSDG-LD is less susceptible to attacks aiming to infer whether specific individuals were included in the original dataset, an important consideration when working with sensitive domains such as healthcare. In terms of distributional similarity, PAR performs better. It produces higher values for both 1-TVD and 2-TVD 0.84 and 0.64 respectively compared to 0.71 and 0.61 for NSSDG-LD. The results suggest that PAR more accurately replicates the statistical structure of the original dataset, which is essential for ensuring that synthetic data can be used reliably in statistical analysis and exploratory modeling. Although NSSDG-LD exhibits slightly better performance in terms of KL divergence (0.47 vs. 0.49), the difference is minimal and does not significantly offset the broader pattern of PAr stronger distributional fidelity. A notable strength of Par lies in its preservation of inter-variable relationships, as evidenced by a substantially lower MI of 0.56, compared to 0.79 for PAR. This indicates that NSSDG-LD more effectively retains the dependency structures present in the real data, which is particularly important for applications that rely on feature interactions. Most importantly, NSSDG-LD achieves a significantly higher AUC score (0.59 vs. 0.43), reflecting superior performance in a downstream classification task. This demonstrates that models trained on NSSDG-LD synthetic data are better able to generalize to real data, confirming its practical utility for machine learning applications

7 Discussion

7.1 Change point Detection Performance

Across all dataset groups, the F1 score tends to increase with more data, reflecting the general principle that larger datasets enable more accurate estimation of temporal dependencies and structural transitions [18]. However, performance saturates beyond approximately 200 -300 data points, particularly in simpler networks. This reflects the point at which the model has already recovered the essential structure with high confidence. Beyond this threshold, additional data contributes little new information, leading to diminishing returns in accuracy [7]. Groups with simple segmentation patterns, such as Group 1 (single change point), achieved the highest F1 scores. This confirms that both continuous and categorical models are effective in low complexity scenarios. In such cases, the posterior distribution over change point configurations is less multimodal, and the algorithm can quickly converge to the correct change point allocation. The MCMC allocation operator faces fewer segment boundaries to evaluate, reducing the probability of incorrect proposals and making acceptance more stable. In contrast, the performance drop in more complex groups, especially Group 5 with four change points reflects the increased difficulty in detecting fine-grained transitions [44]. With more true change points, the number of potential segmentation grows, making MCMC stuck in local modes more often. Additionally, shorter segments offer less information for estimating model parameters, which makes it harder for the MCMC sampler to distinguish between good and bad segmentation proposals. This reduces the effectiveness of each move and limits the algorithm's ability to explore the space of change

point configurations. The categorical models generally underperformed compared to their continuous counterparts, which can be explained both statistically and algorithmically. Discretizations introduces information loss, especially when transitions in the underlying continuous signal are subtle. Once quantized into categories, minor distributional shifts may fall into the same bin, effectively hiding the change point. Consequently, the BDe score used in the categorical model becomes less sensitive to weak temporal shifts. From an algorithmic perspective, the categorical model relies on computing Dirichlet-based marginal likelihoods for each segment, conditioned on discrete parent configurations. When segments are small or parent configurations are numerous, the counts become sparse, reducing the discriminatory power of the scoring function and results in high variance in BDe score get estimated in an inconsistent acceptance rates.

7.2 Structural Recovery Accuracy

Structure learning performance, as measured by AUC, shows strong dependency on both data availability and network dimensionality. In small-scale networks, such as the 2-node case, both the continuous and categorical models achieved near perfect AUCs with as few as 200–300 data points. In this low-dimensional setting, the graph space is small (only four possible directed edges), and MCMC sampling efficiently explores most of the graph configurations within the iteration budget. Additionally, the BDe score can be estimated accurately even with limited samples. However, as the number of nodes increases to 4 or 8, we observe a clear drop in structure recovery accuracy, especially for the categorical model. This can be attributed to two key factors. First, although the structure learning phase does not explicitly estimate CPDs, it relies on the BDe score, which evaluates graph structures by marginalizing over all possible discrete conditional distributions. This score depends critically on observed counts of child-parent configurations. As the number of nodes and parents increases, the number of possible parent configurations grow. This leads to sparser observations per configuration, reducing the reliability of the BDe score and diminishing its ability to differentiate between competing graph structures. Second, the MCMC graph operator proposes edge additions, deletions, or reversals stochastically while enforcing a maximum fan-in constraint. As dimensionality increases, the space of possible graphs expands rapidly and the score differences between competing structures become less pronounced. Lower contrast in BDe scores reduces selection during sampling, making convergence slower. The result is, that more data is required to reliably identify structure as node count and category complexity grow. In contrast, the continuous model benefits from smoother likelihood, when evaluating structural changes, as real-valued data allows for more fine-grained updates to covariance structures or local likelihoods. Nevertheless, both algorithms demonstrated strong behaviors. With enough data (400 points), even the categorical model achieved AUC 0.90 in 8-node networks, showing that the algorithm can recover true structures with high fidelity when the data to parameter ratio becomes favorable.

7.3 Synthetic data generation

The evaluation results highlight a performance difference between the proposed NSSDG-LD algorithm and the baseline PARSynthesizer across synthetic benchmark datasets and EHR data. In smaller scale datasets (e.g., 2-node and 4-node cases), PAR consistently achieves slightly higher scores in utility metrics. These results suggest that PAR is particularly effective at capturing and preserving the marginal and pairwise joint distributions of the data, which is likely due to its parametric nature and reliance on explicit probabilistic assumptions optimized for global distributional. In contrast, NSSDG-LD employs locally computed likelihoods using BDe scoring combined with stochastic change point sampling, which can lead to variability in how well marginal distributions are preserved. That said, the performance gap is acceptable without domain-specific parameter tuning. As structural complexity increases, PAR yields more consistent utility scores across runs, whereas NSSDG-LD's performance varies due to its dependence on MCMC convergence and segmentation accuracy.

A key strength of NSSDG-LD lies in its ability to preserve segment-level temporal dependencies and structural interpretability, particularly in complex datasets. Unlike PARSynthesizer, which excels at matching global marginal distributions, NSSDG-LD captures local dynamics through change point inference and segmentwise CPD extraction. These CPDs are then clustered and synthetic sequences are generated by sampling from the resulting clusters and a learned transition matrix. This design enables the model to reflect distinct temporal regimes and variable dependencies, as evidenced by its better classification AUROC, especially on the MIMIC dataset.

From a privacy perspective, the stochasticity introduced at multiple stages—inference, clustering, and CPD sampling—acts as a natural defense against memorization, reducing the likelihood of membership inference attacks without explicit differential privacy mechanisms. However, the generative process also presents limitations. Clustering flattened CPDs may disregard structural nuances in multi-dimensional conditional tables, and sampling from clusters can lead to underrepresented transitions. In general, while NSSDG-LD trades off some utility loss, its modular design offers a solid base between utility, interpretability and privacy.

8 Conclusion

This study shows that the proposed NSSDG-LD method, which is a combination of MCMC-based change point detection and structure learning within a DBN framework, is

effective ground framework for generating synthetic data that captures both temporal and structural dependencies. Across synthetic datasets, change point detection performance improved as the data points increased but saturated at a certain point, suggesting that less data is sufficient to recover low-complexity segmentation. In contrast, complex segmentation scenarios showed reduced accuracy due to combinatorial growth in segmentation space and limited data per segment. Structural recovery results further highlighted the challenge of sparsity and exponential growth in parent configurations, particularly in categorical models, which caused performance degradation in higher-dimensional networks. However, both change point detection and structure learning proved reliable given adequate data, with continuous models offering better performance due to smoother scoring landscapes.

The results for generated synthetic data, NSSDG-LD performed competitively with the PARSynthesizer, particularly in preserving segment wise dependencies and structural patterns. While PAR yielded slightly better results in marginal distribution alignment NSSDG-LD achieved higher AUROC, especially on the real-world MIMIC dataset, emphasizing its strength in maintaining interaction patterns. Moreover, NSSDG-LD's sampling based generation process offers potential privacy advantages due to its stochastic nature and dynamic structure modeling. While the method shows strong potential for generating realistic and privacy-aware longitudinal data, further refinements are needed to improve utility and privacy.

9 Future Work

9.1 Differential Privacy

Privacy is a major concern when working with real clinical data, especially when that data is shared or reused for secondary analysis. While our framework includes some privacy trough randomness, additional steps can be taken to strengthen privacy even further. One promising direction is to integrate differential privacy (DP) directly into the data generation process.

There are several stages in the pipeline where DP could be introduced. For example, when creating individual Bayesian network structures for each patient, we could apply differential privacy to ensure that no single patient's data has too much influence on the resulting model. Later, during the process of aggregating these individual networks into a global structure, we could again apply DP to protect against any single patient's contribution being identifiable.

Of course, using DP often comes with a trade-off. The more strongly we enforce privacy, the more noise we need to add which can reduce the accuracy or utility of the generated data. That said, this balance between privacy and utility is something that can be studied and optimized. In future work, it can be explored for tuning this trade-off more effectively, so that the synthetic data remains both safe to share and valuable for real-world analysis.

9.2 Rare Disease Aware Modeling and Synthetic Data Generation

Another area for future work could be handling rare diseases in both structure learning and synthetic data generation. In healthcare data, it can be observed that a small subset of patients may exhibit unusual patterns because of rare conditions or unique treatment responses. These unusual cases are often underrepresented or smoothed out when building generalized models, yet they carry important clinical relevance especially in scenarios such as adverse event prediction, rare disease modeling, or stress testing of decision systems.

Our current model focuses on learning population wide patterns, which may lead to the ignore or dilution of these rare behaviors. Future extensions could incorporate robust modeling techniques that identify and preserve rare structures without letting them distort the overall model. For example, separate structure learning could be performed on isolated clusters of atypical patients, or mixture models could be used to learn both dominant and rare patterns simultaneously. Additionally, synthetic data generation could be enhanced by ensuring that the simulator includes low-frequency but clinically critical trajectories, which are often missed in average case models.

Incorporating rare patterns methods would improve the diversity, realism, and clinical utility of the synthetic data. It would also support more comprehensive evaluations of downstream models by exposing them to a broader range of patient profiles including those most vulnerable to algorithmic bias or misclassifications in real-world applications.

9.3 Incorporating Time-Varying Graph Structures

In the current framework, the dependency structure among variables is held constant across the entire time series. Once a consensus graph is learned based on edges that frequently appear across patients it is fixed, and only the conditional probabilities are allowed to vary between segments identified by change points. While this simplifies model design and supports efficient estimation, it imposes a strong assumption: that the same set of variable relationships holds true throughout a patient's clinical journey. However, in real-world healthcare data, the relationships between variables are often dynamic and context-dependent.

For example, consider a patient undergoing cancer treatment. In the early stages, the clinical data includes variables like tumor size, prescribed medication, blood pressure, and stress levels, which are used to monitor progress over several visits. However, at a later point, it is revealed that the patient is a smoker, something that had not been recorded earlier. This new information is clinically significant, as smoking can influence both treatment response and long term outcomes. The model's existing structure does not include a

node for smoking status, so this discovery introduces a new variable that may need to connect to several others, such as tumor progression or drug effectiveness. Adding this node and possibly new edges to reflect its influence would alter the graph structure and improve the model's ability to reflect the patient's actual condition.

To better capture such transitions, future work could explore models that allow the graph structure itself to evolve over time. Instead of using a single global graph, the model could learn separate structures for different segments, each tailored to a specific phase of the clinical timeline. This would make the model more expressive, allowing it to detect not only when variable distributions change, but also when the nature of their relationships shifts. For example, it could learn that a treatment variable only becomes connected to a lab result after a certain threshold is crossed or after a diagnosis is made.

References

[1] George J. Annas et al. 2003. HIPAA Regulations–A New Era of Medical-Record Privacy? *New England Journal of Medicine* 348, 15 (2003), 1486–1490.

[2] Alaa Badawi, Giancarlo Di Giuseppe, Alind Gupta, Abbey Poirier, and Paul Arora. 2020. Bayesian network modelling study to identify factors influencing the risk of cardiovascular disease in Canadian adults with hepatitis C virus infection. *BMJ open* 10, 5 (2020), e035867.

[3] Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. John Wiley & Sons.

[4] Javier Cózar, José M Puerta, and José A Gámez. 2017. An application of dynamic Bayesian networks to condition monitoring and fault prediction in a sensored system: A case study. *International Journal of Computational Intelligence Systems* 10, 1 (2017), 176–195.

[5] Fida K Dankar and Mahmoud Ibrahim. 2021. Fake it till you make it: Guidelines for effective synthetic data generation. *Applied Sciences* 11, 5 (2021), 2158.

[6] Adnan Darwiche. 2010. Bayesian networks. *Commun. ACM* 53, 12 (2010), 80–90.

[7] Pedro Domingos. 2000. A Unifeid Bias-Variance Decomposition and its Applications. 231–238.

[8] Khaled El Emam, David Buckeridge, Robyn Tamblyn, Angelica Neisa, Elizabeth Jonker, and Aman Verma. 2011. The re-identification risk of Canadians from longitudinal demographics. *BMC medical informatics and decision making* 11 (2011), 1–12.

[9] Markus Endres, Asha Mannarapotta Venugopal, and Tung Son Tran. 2022. Synthetic data generation: A comparative study. In *Proceedings of the 26th international database engineered applications symposium*. 94–102.

[10] European Parliament and Council. 2016. General Data Protection Regulation (GDPR) (EU) 2016/679. https://eur-lex.europa.eu/eli/reg/2016/679/oj. Accessed: 2024-05-20.

[11] Dan Geiger and David Heckerman. 1994. Learning Gaussian networks. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (1994), 235–243.

[12] Z. Ghahramani. 1998. Learning Dynamic Bayesian Networks. In *Adaptive Processing of Sequences and Data Structures*. Springer, 168–197.

[13] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. 2020. Dynamical variational autoencoders: A comprehensive review. *arXiv preprint arXiv:2008.12595* (2020).

[14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[15] Peter J Green. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 4 (1995), 711–732.

[16] Marco Grzegorczyk and Dirk Husmeier. 2009. Non-stationary continuous dynamic Bayesian networks. *Advances in neural information processing systems* 22 (2009).

[17] A. J. Hartemink. 2001. *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*. PhD thesis. Massachusetts Institute of Technology.

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*.

[19] David Heckerman, Dan Geiger, and David Maxwell Chickering. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning* 20, 3 (1995), 197–243.

[20] Alistair E. W. Johnson, Lucas Bulgarelli, Tom J. Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo A. Celi, and Roger G. Mark. 2024. MIMIC-IV (version 3.1). https://doi.org/10.13026/kpb9-mt58. PhysioNet.

[21] Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li wei H. Lehman, Leo A. Celi, and Roger G. Mark. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data* 10 (2023). https://doi.org/10.1038/s41597-022-01899-x

[22] Mladen Kolar, Le Song, Amr Ahmed, and Eric P Xing. 2010. Estimating time-varying networks. *The Annals of Applied Statistics* (2010), 94–123.

[23] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E* 69, 6 (2004), 066138.

[24] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86. http://www.jstor.org/stable/2236703

[25] Anton D. Lautrup, Tobias Hyrup, Arthur Zimek, and Peter Schneider-Kamp. 2024. Syntheval: a framework for detailed utility and privacy evaluation of tabular synthetic data. *Data Mining and Knowledge Discovery* 39, 1 (Dec. 2024). https://doi.org/10.1007/s10618-024-01081-4

[26] Sophie Lèbre. 2009. Inferring Dynamic Genetic Networks with Low Order Independencies. *Statistical Applications in Genetics and Molecular Biology* 8, 1 (2009). https://doi.org/doi:10.2202/1544-6115.1294

[27] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. 2021. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *CoRR* abs/2108.04978 (2021). https://arxiv.org/abs/2108.04978

[28] K.P. Murphy. 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph. D. Dissertation. University of California, Berkeley.

[29] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.

[30] Hajra Murtaza, Musharif Ahmed, Naurin Farooq Khan, Ghulam Murtaza, Saad Zafar, and Ambreen Bano. 2023. Synthetic data generation: State of the art in health care domain. *Computer Science Review* 48 (2023), 100546. https://doi.org/10.1016/j.cosrev.2023.100546

[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[32] Katariina Perkonoja, Kari Auranen, and Joni Virta. 2024. Methods for generating and evaluating synthetic longitudinal patient data: a

systematic review. arXiv:2309.12380 [stat.ME] https://arxiv.org/abs/2309.12380

[33] Trivellore E Raghunathan, Jerome P Reiter, and Donald B Rubin. 2003. Multiple imputation for statistical disclosure limitation. *Journal of official statistics* 19, 1 (2003), 1.

[34] Faisal Ramzan, Claudio Sartori, Sergio Consoli, and Diego Reforgiato Recupero. 2024. Generative adversarial networks for synthetic data generation in finance: Evaluating statistical similarities and quality assessment. *AI* 5, 2 (2024), 667–685.

[35] Joshua Robinson and Alexander Hartemink. 2008. Non-stationary dynamic Bayesian networks. *Advances in neural information processing systems* 21 (2008).

[36] Joshua W. Robinson and Alexander J. Hartemink. 2010. Learning Non-Stationary Dynamic Bayesian Networks. *J. Mach. Learn. Res.* 11 (Dec. 2010), 3647–3680.

[37] Joshua W Robinson, Alexander J Hartemink, and Zoubin Ghahramani. 2010. Learning non-stationary dynamic Bayesian networks. *Journal of Machine Learning Research* 11, 12 (2010).

[38] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence* 1, 5 (2019), 206–215.

[39] N. Ruiz, K. Muralidhar, and J. Domingo-Ferrer. 2018. On the Privacy Guarantees of Synthetic Data: A Reassessment from the Maximum-Knowledge Attacker Perspective. In *Privacy in Statistical Databases*. Springer International Publishing, Cham, 59–74.

[40] scikit-learn developers. 2024. KMeans clustering — scikit-learn documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. Accessed: 2025-05-30.

[41] Marco Scutari, Catharina Elisabeth Graafland, and José Manuel Gutiérrez. 2019. Who Learns Better Bayesian Network Structures: Accuracy and Speed of Structure Learning Algorithms. arXiv:1805.11908 [stat.ME] https://arxiv.org/abs/1805.11908

[42] Judith D Singer and John B Willett. 2003. *Applied longitudinal data analysis: Modeling change and event occurrence*. Oxford university press.

[43] U.S. States. 1996. The Health Insurance Portability and Accountability Act (HIPAA). https://aspe.hhs.gov/reports/health-insurance-portability-accountability-act-1996 Accessed: 2023-10-25.

[44] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing* 167 (Feb. 2020), 107299. https://doi.org/10.1016/j.sigpro.2019.107299

[45] Ioannis Tsamardinos, Laura Brown, and Constantin Aliferis. 2006. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning* 65 (10 2006), 31–78. https://doi.org/10.1007/s10994-006-6889-7

[46] Yaxin Wang, Zhen Zhang, Heng Qian, Yongchao Gao, and Qiuyue Wang. 2024. A High-Dimensional Temporal Data Publishing Method Based on Dynamic Bayesian Networks and Differential Privacy. In *2024 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN60899.2024.10650346

[47] Wikipedia contributors. 2024. Dirichlet-multinomial distribution — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Dirichlet-multinomial_distribution

[48] Jinhong Wu, Konstantinos Plataniotis, Lucy Liu, Ehsan Amjadian, and Yuri Lawryshyn. 2023. Interpretation for variational autoencoder used to generate financial synthetic tabular data. *Algorithms* 16, 2 (2023), 121.

[49] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Trans. Database Syst.* (2017). https://doi.org/10.1145/3134428

[50] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)* 42, 4 (2017), 1–41.

[51] Kevin Zhang, Kalyan Veeramachaneni, and Neha Patki. 2022. Sequential Models in the Synthetic Data Vault. (2022).

[52] Geoffrey Zweig and Stuart Russell. 1998. Speech recognition with dynamic Bayesian networks. (1998).

10 Appendix

10.1 Implementation phase

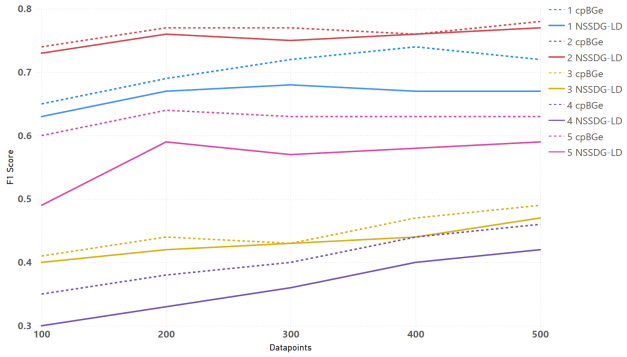

Figure 13. F1 Score on Y-axis and the amount of Data points on the X-axis. 8-Node dataset

10.2 Visuals of the test datasets

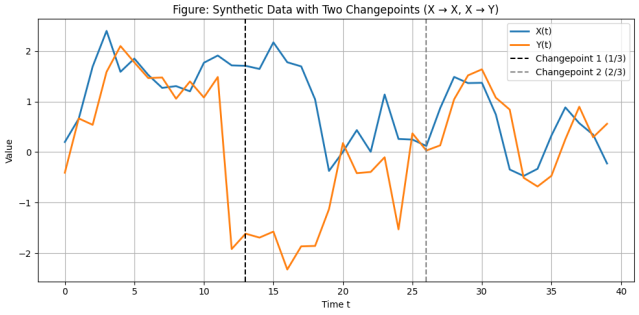

Figure 14. 2-Node dataset

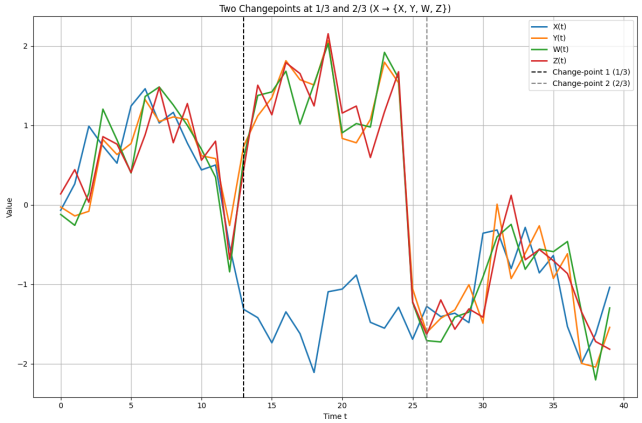

Figure 15. 4-Node dataset

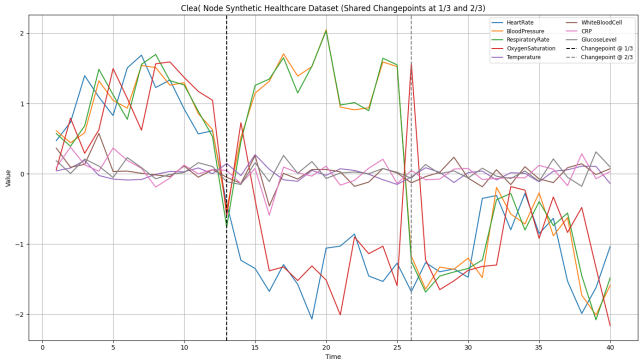

Figure 16. 8-Node dataset

10.3 Description of Selected Variables from the MIMIC-IV Dataset

Table 3. MIMIC-IV dataset

Variable	Description
ab_name	Name of the antibiotic tested against the detected organism.
admission_location	Location from where the patient was admitted (e.g., ER, clinic).
admission_type	Type of hospital admission (e.g., emergency, elective).
admittime	Timestamp when the patient was admitted to the hospital.
anchor_age	Patient's age at time of admission.
discharge_location	Location to which the patient was discharged.
dischtime	Timestamp when the patient was discharged from the hospital.
gender	Biological sex of the patient (Male or Female).
hospital_expire_flag	Indicates if the patient died during the hospital stay (1 = Yes, 0 = No).
label	Indicates if the case is serious (1 if length of stay > 72 hours and not discharged home or deceased).
length_of_stay	Duration of hospital stay in days.
medication	Name of the medication administered during the stay.
org_name	Name of bacterial organism found, if any.
spec_type_desc	Type of specimen collected (e.g., blood, urine).
test_name	Name of the microbiological test performed.
interpretation	Result of antibiotic susceptibility test (e.g., Sensitive, Resistant).