

DAGSurv: Directed Acyclic Graph Based Survival Analysis Using Deep Neural Networks

Ansh Kumar Sharma*

ANSH18130@IITD.AC.IN

Rahul Kukreja*

RAHUL18254@IITD.AC.IN *

Ranjitha Prasad

RANJITHA@IITD.AC.IN

ECE dept., IIT Delhi

Shilpa Rao

SHILPA@IITG.AC.IN

ECE dept., IIT Guwahati

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Causal structures for observational survival data provide crucial information regarding the relationships between covariates and time-to-event. We derive motivation from the information theoretic source coding argument, and show that incorporating the knowledge of the directed acyclic graph (DAG) can be beneficial if suitable source encoders are employed. As a possible source encoder in this context, we derive a variational inference based conditional variational autoencoder for causal structured survival prediction, which we refer to as **DAGSurv**. We illustrate the performance of **DAGSurv** on low and high-dimensional synthetic datasets, and real-world datasets such as METABRIC and GBSG. We demonstrate that the proposed method outperforms other survival analysis baselines such as **Cox** Proportional Hazards, **DeepSurv** and **Deephit**, which are oblivious to the underlying causal relationship between data entities.

1. Introduction

Modern data analysis and processing involve vast amounts of data, where the structure carries critical information about the interrelationships between the entities. This structure is often captured via a *graph*, where an unweighted/weighted edge provides a flexible way of representing the relationship between the nodes. Several signal processing and machine learning algorithms in the past decade have analyzed graphical data (Marques et al., 2020). In the context of machine learning, ignoring these relationships between covariates in the data may lead to biased and erroneous predictions. Hence, it is crucial to incorporate the knowledge of graph topology into learning algorithms.

Directed acyclic graphs (DAG) allows statistical modeling of covariates by enforcing a topological ordering of these entities. DAGs are useful in answering what-if questions such as “*What* would be the system behavior *if* a variable is set to a value A instead of B?”, with a focus on taking actions that induce a controlled change in systems. For instance, while placing an advertisement on online platforms, the relevant *what-if* question is associated with the platform used for ad-placement, and the outcome is time-to-buy. Obtaining the cause-effect relationship between the platform and the outcome only would lead to erroneous

* * indicates equal contribution

predictions since user covariates such as age, geography, online purchase behavior, economic strata etc., also impact a purchase, albeit indirectly (Kumar et al., 2020), as depicted in Fig. 1. Modeling such data as a graphical model allows us to encode the graph structure using conditional independence relationship among random variables that are represented by the vertices, as depicted in Fig. 1. In this work, we assume that the joint distribution of the covariates factorizes as dictated by the adjacency matrix of a DAG whose vertices are features of the dataset.

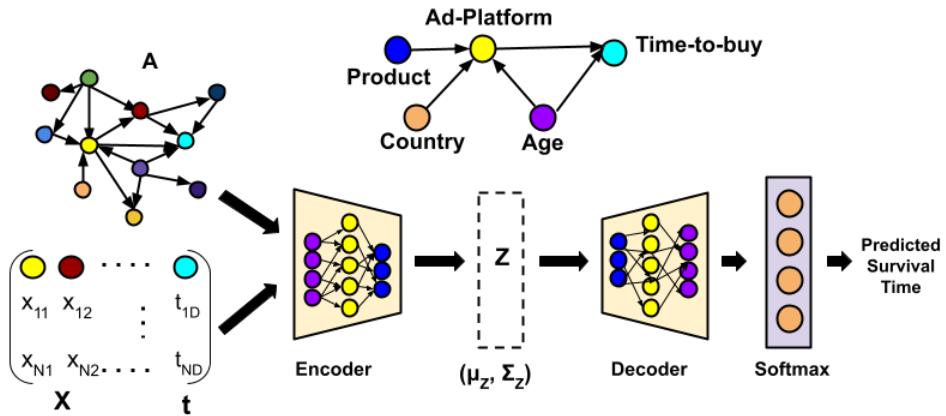


Figure 1: DAGSurv framework: The input of conditional VAE consists of the dataset \mathcal{D} (defined in the sequel) and the adjacency matrix \mathbf{A} . The latent variable that encodes \mathcal{D} and \mathbf{A} is given by \mathbf{Z} . Unlike conventional VAE, the output of CVAE based decoder is the conditional distribution $p(\mathbf{t}|\mathbf{X}, \mathbf{Z})$, and we apply a softmax layer to obtain the predicted survival time. We also illustrate the example graph from the advertising use-case.

Survival analysis (SA) is a well-known statistical technique for the study of temporal events, where time-to-an-event data is modeled using a probabilistic function of fully or partially observed covariates. An impediment in modeling time-to-event data is the presence of *censored* observations, i.e., instances whose event of interest is not observed (and hence, time-to-event information is missing). Neglecting censored data introduces bias in the inference process, and hence, analyzing such data necessitates significantly different statistical and machine learning techniques (Katzman et al., 2018; Lee et al., 2018). Moreover, such maximum likelihood techniques for survival analysis do not enforce any relationship between the features, and the model learns the interactions between the features and the time-to-event outcomes, i.e., any feature interaction is outcome based. In our work, we provide the DAG as an input, with the features as the nodes of the DAGs and their interactions is represented by the edges of the DAG.

Contributions: In this work, we integrate the cause-effect relationship between covariates and the time-to-event outcome by encoding the causal DAG structure into the analysis of temporal data. The contributions are as follows:

- Using information-theoretic source coding arguments, we show that by utilizing the knowledge of the adjacency matrix along with the input covariates leads to optimal encoding of the source distribution as compared to the case where covariates are assumed to be statistically independent.
- Motivated by the source coding argument, we propose a conditional variational autoencoder (CVAE) based novel deep-learning architecture to incorporate the knowledge of the causal DAG for structured survival prediction, which we refer to as **DAGSurv**.
- We demonstrate the performance of the proposed **DAGSurv** framework using the time-dependent concordance index (CI) as a metric, on synthetic and real-world datasets such as Metabric and GBSG.

Using experimental results, we demonstrate that incorporating the causal DAG in survival prediction is beneficial, not only for improving outcomes but also for validating the assumed causal dynamics of a system. In the case of real-world datasets, DAG is not readily available and hence, we use a pre-processing step where we estimate the graph from the given dataset, and use the estimated graph as an input to the proposed model. Simulation results confirm our hypothesis that incorporating the DAG into the machine learning model indeed leads to better representation of data which further leads to improved values of time-dependent CI, as compared to conventional SA techniques.

In the sequel, we describe the mathematical preliminaries of SA followed by the source coding argument for optimal source compression if the adjacency matrix is known. Subsequently, we define the proposed **DAGSurv** framework, and conclude with experimental results and discussions.

2. Related Works

It has been proven time and again that incorporating the knowledge of the graph structure into machine learning models reaps immense benefits. Graph convolutional networks (GCNs) are powerful tools that are used with undirected graphs for semi-supervised classification per instance in the dataset (Kipf and Welling, 2017). In this work, we focus on exploiting the relationship between the covariates in a dataset, and hence, the GCN is not directly applicable. Knowledge graphs bring in the ability to establish relationships between entities in an efficient manner that is explainable and re-usable. However, these relationships are often semantic (Nickel et al., 2015), and may not be of statistical relevance.

In cases where graphs provide statistical information, probabilistic graphical models framework play an important role (Koller and Friedman, 2009). In probabilistic graphical models, nodes of a graph are considered as random variables, and the covariate and target information are considered as the realizations of these random variables. Evidently, the edge between the random variables translates the statistical relationships between random variables, and hence, the graph forms a joint distribution over the dataset. In scenarios where the underlying graph is known, deep neural networks have been used along with graphical models for prediction (Yoon et al., 2019). In this work, we utilize the probabilistic graphical models based framework for graph-based survival prediction.

In the context of survival analysis, Kaplan-Meier (KM) technique is a popular but naive, covariate-ignorant non-parametric technique for obtaining the empirical estimate of the survival function (Kaplan and Meier, 1958). An improvement to the KM technique is the Cox proportional hazards model (Cox, 2018) (CPH) which incorporates the user covariates for inference. Several parametric methods that propose Weibull or log-normal distributions Wang et al. (2019) and non-parametric methods using Gaussian processes have been proposed for survival analysis (Fernández et al., 2016). Modern techniques based on deep neural networks (DNNs) have been used for time-to-event analysis in (Faraggi and Simon, 1995) and (Katzman et al., 2018), where non-linear representations replace linear models for modeling the relationship between covariates and the risk. However, the limitation of these methods is the inherent assumption of constant hazard rate and the linearity of the log-hazard rate. In (Lee et al., 2018), authors propose a cumulative index curve (CIC) approach, which uses the marginal probabilities of an event, in the presence of multiple competing events. This technique does not assume constant hazard rate or any other assumptions about the model.

Probabilistic graphical models have been used in the context of survival analysis (Bandyopadhyay, 2015) where graph based inference algorithms are proposed for survival prediction assuming constant hazard rate. In contrast, we propose a conditional VAE (CVAE) based graphical model approach for structured survival prediction, where we do not assume constant hazard rate. Our work is closely related to DAG-GNN (Yu et al., 2019). Note that the proposed CVAE is inspired by certain design aspects of DAG-GNN, but it is substantially different in functionality, as compared to DAG-GNN (Yu et al., 2019). In DAG-GNN, the VAE (and not CVAE) is designed to learn the weighted adjacency matrix of the DAG and it is not specific to a machine learning task. In our work, we incorporate the adjacency matrix as a *known* parameter, and subsequently obtain an assumption-free machine learning model for survival prediction. Although, survival analysis is the theme of this work, it will be evident from the analysis that our model can be adapted for classification and regression tasks as well.

Several methods that incorporate graph-represented relations of features into predictions approaches using GCNs have been proposed in literature. However, these methods incorporate separate modules for graph embedding and regression, classification or survival analysis. For instance, in (Di et al., 2020), a graph is considered between patches of pathological images and the feature representation generated by GCN is considered for survival analysis. On the other hand, we embed the knowledge of the graph into the network, and specifically address the problem of survival analysis. Another closely related work is (Chen, 2019), where authors propose an undirected graph based survival analysis by using a probabilistic graphical model with the exponential family distribution to describe the relationship between the covariates. In comparison, we specifically consider DAGs to model causal relationships, and do not assume specific probabilistic models among covariates.

3. DAG Based Survival Analysis: Preliminaries and Loss Function

In this section, we describe the problem of DAG-based SA. First, we provide mathematical preliminaries of survival prediction and subsequently formulate the problem based on the source coding argument. We propose the CVAE framework as a possible source encoder

that incorporates the knowledge of DAG for survival prediction. We develop the variational loss function, which is dual-purpose in the sense that it incorporates the causal DAG along with learning system parameters for survival prediction.

3.1. Mathematical Preliminaries

Time-to-event datasets such that the dataset $\mathcal{D} = \{(\mathbf{x}^{(n)}, t^{(n)}, \delta^{(n)})\}_{n=1}^N$ are usually characterized by three variables for the n -th instance where, $\mathbf{x}^{(n)} \in \mathbb{R}^L$, i.e, for n instances, $\mathbf{X} \in \mathbb{R}^{N \times L}$. Here, L represents the number of covariates. We consider survival time $t^{(n)}$ as discrete, and the time horizon as finite so the $t \in \mathcal{T}$ where $\mathcal{T} = \{0, \dots, M\}$ for a predefined maximum time horizon M . Further, $\mathbf{t} \in \mathbb{R}^{N \times 1}$ represents the time at which the event has occurred and $\delta^{(n)} \in \{0, 1\}$ is an indicator variable which specifies if the n -th instance is censored or not. Time-to-event models are characterized by the survival function given by

$$S(t|\mathbf{x}) = P(T > t|\mathbf{x}) = 1 - F(t|\mathbf{x}),$$

which is defined as the fraction of the population that survives up to time t ¹, where $F(t|\mathbf{x})$ represents the cumulative distribution function of time-to-event, given user covariates \mathbf{x} . Another important statistic is the conditional hazard rate function $h(t|\mathbf{x})$ which is defined as the instantaneous rate of occurrence of an event at time t given covariates \mathbf{x} . It is known that the relationship between $h(t|\mathbf{x})$ and $S(t|\mathbf{x})$ from standard definitions is given by:

$$h(t|\mathbf{x}) = \lim_{dt \rightarrow 0} \frac{P(t < T < t + dt|\mathbf{x})}{P(T > t|\mathbf{x})dt} = \frac{f(t|\mathbf{x})}{S(t|\mathbf{x})}, \quad (1)$$

where $f(t|\mathbf{x})$ is the conditional survival density function and $S(t|\mathbf{x})$ is as described earlier. The Cox-PH model Cox (1972) is a semi-parametric, linear model where the conditional hazard function $h(t|\mathbf{x})$ depends on time through the baseline hazard $h_0(t)$, and independent covariates \mathbf{x} such that

$$h(t|\mathbf{x}) = h_0(t) \exp(\mathbf{x}^T \boldsymbol{\gamma}). \quad (2)$$

For a given dataset with N observations as described earlier, Cox-PH estimates the regression coefficients, $\boldsymbol{\gamma} \in \mathbb{R}^L$, such that the partial likelihood is maximized (Cox, 1972). In DeepSurv, the authors propose a CPH based DNN, as the basis for a treatment recommender system. Further, DeepHit directly learns the joint distribution of survival times and events, effectively avoiding the PH assumptions or those inherent to a particular form of the model. In these methods, the covariates are assumed to be independent, and there is no formal mechanism using which any dependence between covariates can be included. In Chen (2019), an undirected graph is assumed between the covariates and an exponential distribution based probabilistic graphical model is incorporated into analysis. However, in contrast, we design a CVAE based framework for incorporating a DAG between the covariates for survival prediction. Note that the proposed technique does not require any modeling assumptions such as those in Chen (2019), and hence, it is suitable for all datasets.

1. For better readability, we drop the superscript n while discussing about generic concepts.

3.2. Problem Formulation

In this work, we employ the the DAG, denoted as $\mathcal{G}(V, E)$, to describe the causal relationship between the features in the dataset \mathcal{D} . Each vertex in $\mathcal{G}(V, E)$ represents a random variable with $V = \{1, \dots, L+1\}$ consisting of the indices of these random variables, i.e., X_l is a vertex if $l \in V$. Further, let $V \times V$ consist of all pairs of indices in V . A pair of random variables $\{X_l, X_m\}$ is called an edge of the graph \mathcal{G} if $(l, m) \in E \subset V \times V$. The $L+1$ vertices includes the L covariates in \mathbf{X} , and the $L+1$ -th vertex is the target variable given by the survival time \mathbf{t} . Let $\mathbf{A} \in \mathbb{R}^{(L+1) \times (L+1)}$ denote the weighted adjacency matrix of this DAG.

3.2.1. MOTIVATION

In this work, the covariate matrix \mathbf{X} and the adjacency matrix \mathbf{A} are encoded into an efficient representation for structured survival prediction. We view the problem of encoding \mathbf{X} and \mathbf{A} jointly as a problem of *source encoding*. We invoke the basic principles of information theory which establishes the fundamental limit for the compression of information. For optimal source compression, the expected length of the source code must be greater than or equal to the entropy of the source (Cover, 1999). First we note that the adjacency matrix governs the probabilistic relationship between the features, as given by the following proposition.

Proposition 1. *The adjacency matrix \mathbf{A} of the directed acyclic graph (DAG) $\mathcal{G}(V, E)$ characterizes the joint distribution $p(\mathbf{t}, \mathbf{X})$.*

Proof. See the supplementary material. □

In the next two propositions, we establish that the entropy of the source that emits symbols governed by $p(\mathbf{t}, \mathbf{X}|K_{\mathbf{A}})$ with $\mathbf{A} \neq 0$, is upper bounded by the entropy of a source that emits statistically independent source symbols. Here, we use a binary random variable $K_{\mathbf{A}}$, such that $K_{\mathbf{A}} = 1$, if the graph is known apriori and 0 otherwise. Let $\mathcal{X}_{pa(i)}$ denote the set of parents of X_i .

Proposition 2. *The adjacency matrix \mathbf{A} is a non-zero matrix if and only if the i -th term in the factorization of $p(\mathbf{X}|K_{\mathbf{A}})$ given by $p(X_i|\mathcal{X}_{pa(i)})$ is not equal to $p(X_i)$, for any i .*

Proof. See the supplementary material. □

In other words, the above proposition also implies that if $\mathbf{A} = 0$, then the set of parents of X_i given by $\mathcal{X}_{pa(i)} = \{\}$, and hence, $p(\mathbf{X}|K_{\mathbf{A}}) = \prod_{i=1}^L p(X_i)$.

Proposition 3. *If the i -th term in the factorization of $p(\mathbf{t}, \mathbf{X}|K_{\mathbf{A}})$ given by $p(X_i|\mathcal{X}_{pa(i)})$ is not equal to $p(X_i)$ for any i , then $H(\mathbf{X}) < \sum_{i=1}^L H(X_i)$, where $H(\cdot)$ is the entropy function.*

Proof. See the supplementary material. □

From the propositions stated above, we observe that if $A(i, j) \neq 0$ for all i, j , then the entropy of the source is strictly smaller than entropy of the source that emits statistically independent symbols. Furthermore, if the knowledge of \mathbf{A} is not provided for data representation, the optimal encoder may need to consider $A(i, j) = 0$ for all i, j , and as a result

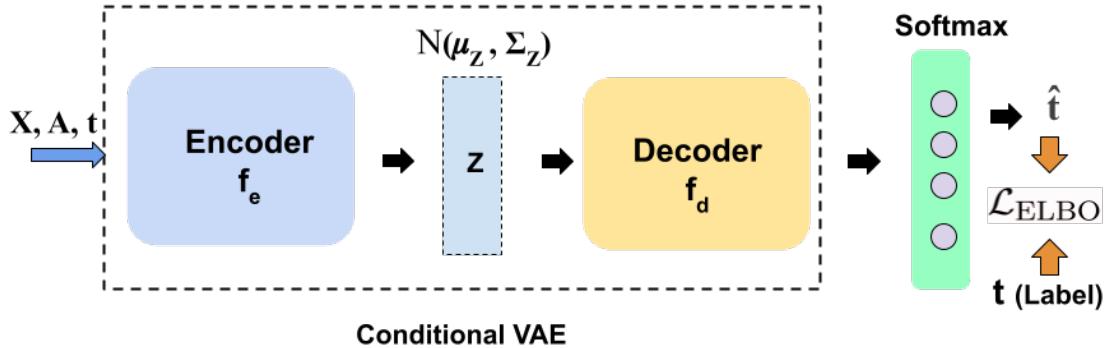


Figure 2: DAGSurv framework: \mathbf{X} , \mathbf{A} and \mathbf{t} are provided as inputs to the CVAE during training. The decoder is followed by the softmax layer, such that the output $\hat{\mathbf{t}}$ represents the probability that an individual will experience an event at a given time. During test time, only the decoder(f_d) is used where \mathbf{X} , \mathbf{Z} (Input samples to decoder are from $N(0, \mathbf{I})$. The reparameterization trick ensures that \mathbf{Z} is sampled from $N(\mu_{\mathbf{Z}}, \Sigma_{\mathbf{Z}})$ and this distribution is embedded into decoder during training.) and \mathbf{A} are provided as inputs, and $\hat{\mathbf{t}}$ is obtained at the output.

the number of bits used to represent the source is as large as $\sum_{i=1}^L H(X_i)$. Therefore, it is evident that the knowledge of \mathbf{A} must be appropriately used for data representation of the source so that the number of bits required to encode such a source is strictly less than $\sum_{i=1}^L H(X_i)$. Here, we state and prove this fundamental information theoretic source encoding argument, since it provides us a strong motivation to design efficient encoders. Towards that direction, we incorporate the knowledge of \mathbf{A} in the context of structured survival prediction.

3.2.2. CVAE AND THE COST FUNCTION

A possible approach towards utilizing the knowledge of the adjacency matrix for source encoding is by using the variational autoencoder (VAE) (Kingma and Welling, 2019). Several authors have successfully employed VAEs for joint source-channel coding (Choi et al., 2019). Motivated by this, we derive a conditional variational autoencoder (CVAE) framework for DAG based survival prediction, while incorporating the knowledge of \mathbf{A} .

We use the standard CVAE (Sohn et al., 2015) for incorporating DAG into survival prediction. The conditional refers to the conditional probability density function (pdf) used in CVAE, instead of the joint pdf as used in VAE. Although VAE and CVAE use latent variable based formulation, conditioning on \mathbf{x} is unique to CVAE. The novelty in the proposed method is in combining the knowledge of DAG and individual features for SA by encoding the DAG structure into the graph as additional information. The generative aspect of CVAE allows for the ELBO framework for encoding the graph into the neural network, and predictive capability of DAGSurv is a result of prediction capability of CVAE. In order to design DAGSurv, we employ the sample generation process according to the

generalized SEM given by

$$\mathbf{t} = f_d \left((\mathbf{I} - \mathbf{A}^T)^{-1} g([\mathbf{X}^T, \mathbf{Z}^T]) \right), \quad (3)$$

where \mathbf{A}^T is the transpose of matrix \mathbf{A} , $g : \mathbb{R}^{(2L+1) \times N} \rightarrow \mathbb{R}^{(L+1) \times N}$, and $f_d : \mathbb{R}^{(L+1) \times N} \rightarrow \mathbb{R}^{M \times 1}$. Hence, the input to the decoder is \mathbf{A} , and a concatenated matrix consisting of \mathbf{X} and \mathbf{Z} . Here $\mathbf{Z} \in \mathbb{R}^{N \times (L+1)}$ is a latent variable with a zero-mean Gaussian prior distribution $N(0, \mathbf{I})$, and \mathbf{I} is the identity matrix. Often, (3) is referred to as the *decoder* model, and the corresponding *encoder* model is given by

$$\mathbf{Z}^T = (\mathbf{I} - \mathbf{A}^T) f_e(\tilde{\mathbf{X}}^T), \quad (4)$$

where $f_e : \mathbb{R}^{(L+1) \times N} \rightarrow \mathbb{R}^{(L+1) \times N}$ is a parameterized function of the encoder and $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times (L+1)}$ denotes the augmented matrix consisting of the features in \mathbf{X} and time-to-event vector \mathbf{t} , i.e., $\tilde{\mathbf{X}} = [\mathbf{X}, \mathbf{t}]$. Note that if $\mathbf{A} = 0$ above, the encoder is given as $\mathbf{Z}^T = f_e(\tilde{\mathbf{X}}^T)$ and the decoder is given by $\mathbf{t} = f_d(g[\mathbf{X}^T, \mathbf{Z}^T])$, which is similar to the encoder and decoder correspond to the conventional CVAE, where the input covariates \mathbf{X} are considered statistically independent.

For purposes of data-driven survival time prediction, we learn the parameters that constitute encoder and decoder by maximizing the log-evidence $\frac{1}{N} \sum_{n=1}^N \ln(p(t^n | \mathbf{x}_n))$, where \mathbf{x}_n denotes the covariates of the n -th sample in \mathbf{X} . Maximizing the log-evidence is often intractable, and hence, we resort to variational inference theory which allows us to maximize the lower bound on evidence, referred to as ELBO (Bishop, 2006). The relationship between log-evidence and ELBO is given as

$$\frac{1}{N} \sum_{n=1}^N \ln(p(t^{(n)} | \mathbf{x}_n)) \geq \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{q(\mathbf{z}_n | \mathbf{x}_n, t^{(n)})} \left[\ln \left(\frac{p(t^{(n)}, \mathbf{z}_n | \mathbf{x}_n)}{q(\mathbf{z}_n | \mathbf{x}_n, t^{(n)})} \right) \right] \equiv \mathcal{L}_{\text{ELBO}}. \quad (5)$$

Here, $q(\mathbf{z}_n | \mathbf{x}_n, t^{(n)})$, $1 \leq n \leq N$, denotes the variational posterior distribution, which *encodes* the samples into the latent variable \mathbf{z}_n . Here, \mathbf{z}_n denotes the n -th row of \mathbf{Z} . Unlike the conventional VAE, the decoder in CVAE is trained to *predict* the target, and in this context, time-to-event \mathbf{t} for previously unseen samples. In particular, we obtain the mean and covariance of the conditional distribution $p(\mathbf{t} | \mathbf{X}, \mathbf{Z})$, and the predictions are obtained by sampling the conditional distribution. Further, we simplify $\mathcal{L}_{\text{ELBO}}$ as (Bishop, 2006)

$$\mathcal{L}_{\text{ELBO}} = \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{q(\mathbf{z}_n | \mathbf{x}_n, t^{(n)})} \left[\ln(p(t^{(n)} | \mathbf{z}_n, \mathbf{x}_n)) \right] - D_{\text{KL}}(q(\mathbf{z}_n | \mathbf{x}_n, t^{(n)}) || p(\mathbf{z}_n)), \quad (6)$$

where $D_{\text{KL}}(\cdot || \cdot)$ is the KL divergence function and $p(\mathbf{z}_n)$ is the prior distribution on \mathbf{z}_n . Hence, ELBO leads to an expected likelihood based objective function, constrained by KL-divergence. Since time-to-event data is censored, the

$$\ln p(t^{(j)} | \mathbf{x}_j, \mathbf{z}_j) = \delta_j \ln f(t^{(j)} | \mathbf{x}_j, \mathbf{z}_j) + (1 - \delta_j) \ln S(t^{(j)} | \mathbf{x}_j, \mathbf{z}_j), \quad (7)$$

where δ_j is an indicator variable as defined earlier, $f(t | \mathbf{x}, \mathbf{z})$ is the failure density, and $S(t | \mathbf{x}, \mathbf{z})$ is the survival function. Here, $\hat{\mathbf{t}}$ is a probability distribution $\hat{\mathbf{t}} = [\hat{t}_1, \dots, \hat{t}_M]$, i.e., given the covariates \mathbf{X} , \hat{t}_k is the probability that the individual will experience the event at k -th time-epoch, as depicted in Fig. 2. Similar to (Lee et al., 2018), the cost function in (7)

drives the network to learn non-linear, non-proportional relationships between covariates and risks, for a given event. Hence, the overall cost function of the survival based CVAE integrates the above cost function into $\mathcal{L}_{\text{ELBO}}$.

In order to accomplish the proposed design, we use the encoder model which is a multi-layered perceptron (MLP) with weights \mathbf{W}_e , represented as f_e . Accordingly, at the decoder, f_d is an MLP with weights \mathbf{W}_d , followed by a softmax layer. The decoder of the CVAE generates the samples from the conditional distribution $p(\mathbf{t}|\mathbf{Z}, \mathbf{X})$, which are given by

$$\hat{\mathbf{t}} \leftarrow \text{Softmax}((I - \mathbf{A}^T)^{-1} \mathbf{Z}, \mathbf{W}_d, \mathbf{X}), \quad (8)$$

where \mathbf{Z} is generated at encoder. The weights \mathbf{W}_e and \mathbf{W}_d , and thereby the functions f_e and f_d are learnt by maximizing $\mathcal{L}_{\text{ELBO}}$, as given in (6). Since we integrate the SA based cost function given in (7) into $\mathcal{L}_{\text{ELBO}}$, it is possible to train the CVAE for efficient, graph-based, time-to-event prediction. For prediction on previously unseen samples, only the decoder is used, as shown in Fig. 2.

In summary, our model leads to a predictive distribution for the survival time of a user based upon the user’s covariates and the underlying structure that exists among those covariates.

4. Simulation Results

In this section, we demonstrate the efficacy of **DAGSurv** on synthetic and publicly-available real-world clinical datasets such as METABRIC (Curtis et al., 2012), Rotterdam (Foekens et al., 2000) & GBSG (Schumacher et al., 1994). These real-world datasets are a widely-accepted standard, and have been used for bench-marking several methods such as **DeepSurv** (Katzman et al., 2018) and **DeepHit** Lee et al. (2018). We provide the description of the datasets along with the processing steps, followed by the evaluation metric, baseline approaches and implementation specifics of **DAGSurv**. For reproducibility purposes, we have made the code public at <https://github.com/rahulk207/DAGSurv>.

4.1. Datasets & Data processing

4.1.1. SYNTHETIC DATASETS

We sample a random DAG, $\mathcal{G}(V, E)$ using Erdos-Renyi model (Erdos and Renyi, 1959), where, $|V| = L + 1$, L refers to the number of covariates and 1 refers to the target variable which is the time-to-event outcome. For simulations, we set the expected node degree as 3. We initialise the edge weights uniformly but randomly, i.e., as $\forall e \in E$, we have the DAG edge weight $W(e) \sim U(0.5, 2)$. We embed the DAG-based relationship among covariates using the following equations (Yu et al., 2019):

$$\mathbf{X}^T = \mathbf{A}^T(\cos(\tilde{\mathbf{X}} + \mathbf{1})) + \mathbf{Z}_{\mathbf{X}}^T, \quad \text{and} \quad \mathbf{t} = \max(\mathbf{0}, c \exp \{ \mathbf{A}^T(\cos(\tilde{\mathbf{X}}^T + \mathbf{1})) \} + \mathbf{Z}_{\mathbf{t}}^T), \quad (9)$$

where entries of $\mathbf{Z}_{\mathbf{X}}$ and $\mathbf{Z}_{\mathbf{t}}$ are sampled independently from $N(0, 1)$ and $N(30, 70)$, respectively. Further, $\mathbf{1}$ is an all 1 matrix, $\mathbf{0}$ is an all zero matrix, and c is a constant chosen such that we obtain \mathbf{t} in a certain range; we set $c = 90$. Using this data generating process, we obtain 10,000 data points. Although harsh and conservative, we censored 50% of the data,

Dataset	# Censored	# Features	T_{max}	C_{max}
Synthetic-small	50.06%	9	377	91
Synthetic-large	51.58%	49	395	235
METABRIC	42.06%	9	355	337
GBSG	43.23%	7	83	87

Table 1: Description of Synthetic and Real-world Datasets (C_{max} is the maximum Censoring Time).

Dataset	n_l, n_h (Encoder)	n_l, n_h (Decoder)	Activation	lr
Synthetic-small	5,128	3,64	ReLU	1e-4
Synthetic-large	5,64	4,32	ReLU	1e-5
METABRIC	3,256	3,64	SELU	1e-5
GBSG	3,128	3,32	ReLU	1e-5

Table 2: Hyperparameters used in different datasets: n_l and n_h represent the number of layers and number of hidden nodes per layer, respectively and lr is the learning rate.

and we sample censoring time uniformly but randomly as $U(0, \max(t))$. Using the above described settings, we created the following two datasets -

1. **Synthetic-small:** Here, we set $L = 9$ covariates (hence, $|V| = 10$).
2. **Synthetic-large:** In order to test our model's scalability and performance on a bigger dataset, we set $L = 49$.

4.1.2. REAL-WORLD DATASETS

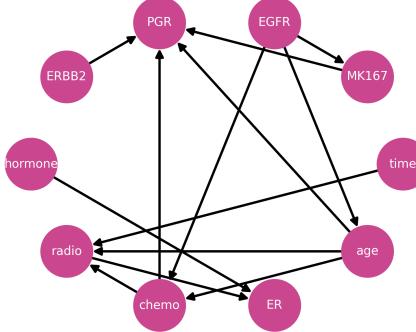


Figure 3: DAG: METABRIC

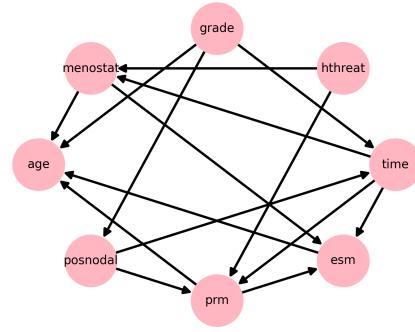


Figure 4: DAG: GBSG

In the context of real-world datasets, the input DAG is not known. Given the covariates in a dataset, manually constructing a DAG may be infeasible since it requires domain-specific expertise, and hence, it can be an expensive process. Instead, we used two well-known algorithms for pre-computing our adjacency matrix \mathbf{A} . They are as follows:

1. **bnlearn, R-package** ([Scutari, 2009](#)) - Within the R package, we used the Hill Climbing (HC) algorithm to learn the structure of Bayesian network, which leads to an unweighted directed graph.
2. **DAG-GNN** ([Yu et al., 2019](#)) - DAG-GNN is a recent deep-learning model for generating a weighted DAG, establishing structure among the features of a given dataset.

We use these algorithms on the real-world datasets as follows:

- **METABRIC:** The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) is a clinical dataset which consists of gene expressions used to determine different subgroups of breast cancer. We consider the data for 1,904 patients with each patient having 9 covariates - 4 gene indicators (MKI67, EGFR, PGR, and ERBB2) and 5 clinical features (hormone treatment indicator, radiotherapy indicator, chemotherapy indicator, ER-positive indicator, age at diagnosis). Furthermore, out of the total 1,904 patients, 801 (42.06%) are right-censored, and the rest are deceased (event). We obtained the DAG as depicted in Fig. 3 using a modified DAG-GNN algorithm.
- **GBSG:** Rotterdam and German Breast Cancer Study Group (GBSG) contains breast-cancer data from Rotterdam Tumor bank. The dataset consists of 2,232 patients out of which 965 (43.23%) are right-censored, remaining are deceased (event), and there were no missing values. In total, there were 7 features per patient that include hormonal therapy (hthreat), age of patient, menopausal status, tumor grade, number of positive nodes, progesterone receptor(in fmol) and estrogen receptor(in fmol). The graph for this dataset is obtained using bnlearn and it is as depicted in Fig. 4.

4.2. Implementation and Evaluation

In this section, we provide the details of the experimental evaluation, which includes the evaluation metric, baseline models, implementation specifics and the experimental results. We randomly split the data into training set (80%) and test set (20%), and further reserved 20% of the training set for validation.

As depicted in Fig. 2, DAGSurv is a CVAE consisting of MLPs as encoder and decoder. The model has a DNN architecture, and we used grid-search to perform an extensive hyperparameter search on the number of layers, number of hidden units, activation function and learning rate. The hyperparameter values that were used to obtain the results reported in this paper are as given in Table 2. Adaptive Moment Estimation (Adam) was chosen as the gradient descent optimization algorithm, and the entire module was coded using pyTorch. Following the implementation in DAG-GNN ([Yu et al., 2019](#)), we set the variance of the latent variable Σ_Z as \mathbf{I}_{L+1} which is the identity matrix in $L+1$ dimensions. We have considered only μ_Z as trainable, since it was observed that the value of Σ_Z explodes due to the exponent term, particularly in datasets with larger time-to-event values. Note that the results remain unaffected in spite of this modification.

4.2.1. EVALUATION METRIC

We employ the time-dependent concordance index (CI) as our evaluation metric since it is robust to changes in the survival risk over time. Mathematically it is given as

$$C_{td} = P\left(F(t^{(i)}|x^{(i)}) > F(t^{(i)}|x^{(j)}) | t^{(i)} < t^{(j)}\right) \approx \frac{\sum_{i \neq j} R_{i,j} \mathbf{1}(F(t^{(i)}|x^{(i)}) > F(t^{(i)}|x^{(j)}))}{\sum_{i \neq j} R_{i,j}}, \quad (10)$$

where $\mathbf{1}(\cdot)$ is the indicator function and $R_{i,j} \triangleq \mathbf{1}(t^{(i)} < t^{(j)})$, i.e., we use an empirical estimate of the time-dependent CI as our metric (Lee et al., 2018). To test the robustness of trained models on unseen data, we performed bootstrapping on the test set. Using the bootstrap C_{td} values obtained on the test set, we plot notched box-plots and compared them for the proposed and baseline methods. The notch here represents 95% confidence interval (C_I) around the median which can be calculated as $\text{median} \pm 1.57 \times \frac{IQR}{\sqrt{b}}$, where IQR is the interquartile range which includes 50% of the data and b denotes the number of bootstrap samples.

4.2.2. BASELINE MODELS

In this section, we discuss the following baseline approaches for survival prediction against which we compare the proposed **DAGSurv**:

- **CoxTime:** Cox-PH is a classical, and one of the most fundamental baselines to compare against. While the PH assumption is essential for these models, they allow easy interpretation and ranking of risk factors. We used **CoxTime** (Kvamme et al., 2019) which is a relative risk neural network model that extends Cox regression beyond linear PH.
- **DeepSurv:** **DeepSurv** is a DNN extension of the classical Cox-PH model. It generally performs better than Cox-PH model since it captures some non-linearity which may be important in the context of real-world datasets.
- **DeepHit:** **DeepHit** predicts the time-to-event directly, unlike survival risk prediction algorithms such as **DeepSurv/Cox**. Furthermore, **DeepHit** is not inherently based on the PH assumption, and hence, an important baseline to compare against.

4.3. Experimental Results

In this section, we illustrate the time-dependent CI values (C_{td}), along with the confidence intervals (95%) using tables and box-plots.

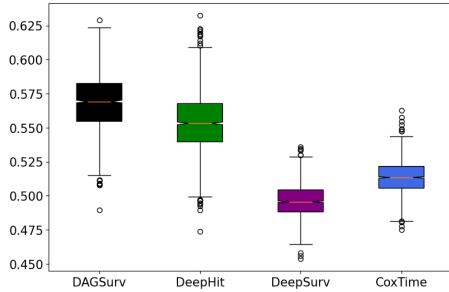
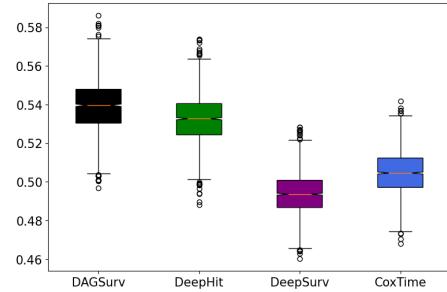
4.3.1. SYNTHETIC DATASET

In this section, we present the results obtained using the proposed and baseline methods on a small and large synthetic datasets which we defined in the previous section. It is observed that most of the models find it hard to learn the underlying model, and the C_{td} values as measured on the test-set are low. It can be observed from Table 3 that **DeepSurv** and **CoxTime** fail to learn a meaningful model and their C_{td} values are close to 0.5. With fewer model-based assumptions, **DeepHit** and **DAGSurv** are able to learn the model with

Synthetic-small		Synthetic-large	
Algorithms	C_{td} (95% C_I)	Algorithms	C_{td} (95% C_I)
DAGSurv	0.5692 \pm 0.0009	DAGSurv	0.5396 \pm 0.0006
DeepHit	0.5532 \pm 0.0009	DeepHit	0.5326 \pm 0.0005
DeepSurv	0.4956 \pm 0.0005	DeepSurv	0.4936 \pm 0.0004
CoxTime	0.5134 \pm 0.0005	CoxTime	0.5045 \pm 0.0005

Table 3: C_{td} for Synthetic-small and Synthetic-large datasets

acceptable C_{td} . Note that we do not employ the constraint on concordance index as in Deephit. Generally this constraint is hard to compute for large datasets, since it requires pairwise computations. The knowledge of the input DAG helps **DAGSurv** to perform better than **DeepHit**, in the absence of the concordance constraint. As expected, the box-plot shows smaller variation in values of C_{td} over the test set since in the case of synthetic data, the testing and training samples come from the same data generating process.

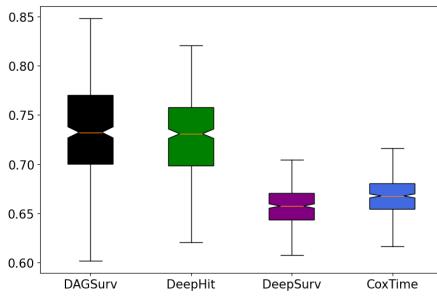
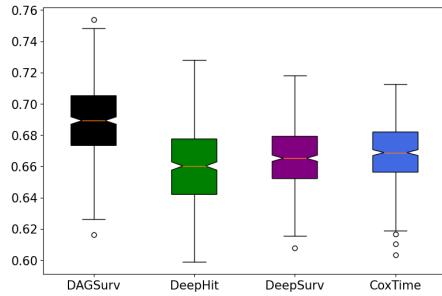
Figure 5: Box-plot: C_{td} for Synthetic-smallFigure 6: Box-plot: C_{td} for Synthetic-large

4.3.2. REAL-WORLD DATASETS

In this section, we illustrate the performance of the proposed approach and the baseline schemes on real-world datasets which we described earlier. We observe that **DAGSurv** consistently performs better or is as competitive as compared to the baseline schemes.

In addition to improved performance, **DAGSurv** lends itself to better interpretation as well. First of all, the concordance score acts as validation for the input graph, i.e., if C_{td} improves when we set $\mathbf{A} = 0$ in **DAGSurv**, it implies that graph is not aiding to obtain better ML models for survival analysis. Further, it also helps to establish relationship between covariates and the outcome. For instance, we observe from the graph pertaining to the GBSG dataset in Fig. 4 that the grade of tumor affects both, the number of positive lymph nodes as well as time-to-event (death). Hence, the relationship between number of positive lymph nodes and survival time, would have to account for the grade of tumor. Such interpretable results are powerful tools for practitioners and clinicians, and we plan to explore the aspects of explainable AI in our future work.

METABRIC		GBSG	
Algorithms	C_{td} (95% C_I)	Algorithms	C_{td} (95% C_I)
DAGSurv	0.7323 \pm 0.0056	DAGSurv	0.6892 \pm 0.0023
DeepHit	0.7309 \pm 0.0047	DeepHit	0.6602 \pm 0.0026
DeepSurv	0.6575 \pm 0.0021	DeepSurv	0.6651 \pm 0.0020
CoxTime	0.6679 \pm 0.0020	CoxTime	0.6687 \pm 0.0019

Table 4: C_{td} for METABRIC and GBSG datasetsFigure 7: Box-plot: C_{td} for METABRICFigure 8: Box-plot: C_{td} for GBSG

4.4. Discussions and Conclusions

In this work, we propose **DAGSurv**, in which we incorporate the knowledge of the causal DAG and design a novel CVAE framework for SA. Using the source coding argument we prove that the knowledge of the DAG leads to reduced entropy as compared to a source that emits statistically independent symbols, a default choice in DAG-agnostic ML models. We employed the CVAE as a possible source encoder for achieving efficient data representation. However, CVAE is not an optimal choice, and we reserve the the design of optimal source encoder to future work.

Using synthetic and real-world datasets, we demonstrated that **DAGSurv** has an improved performance (in terms of concordance index) while it being more interpretable. Using this method requires the knowledge of the DAG, which is generally not known. In the absence of experts' knowledge, we demonstrated that one may opt to use one of the several algorithms available to obtain a DAG from a given dataset. Unlike **CoxTime** and **DeepSurv**, **DAGSurv** can be used in the presence of time-varying hazard. Further, note that **DAGSurv** does not require the expensive concordance index based constraint which requires pairwise comparisons across all the points in a dataset as in (Lee et al., 2018). In spite of not using the constraint, **DAGSurv** is able to perform better than **DeepHit**. Furthermore, **DAGSurv** can be used to validate the causal relations in any graphical model.

Further, extending our analysis to the multiple risk case is straightforward. Some interesting extensions include analysis in the context of recurring events (Gupta et al., 2019) and for counterfactual inference.

References

Sunayan et. al Bandyopadhyay. Data mining for censored time-to-event data: a bayesian network model for predicting cardiovascular risk from electronic health record data. *Data Mining and Knowledge Discovery*, 29(4):1033–1069, 2015.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Li-Pang Chen. Survival analysis of complex featured data with measurement error. 2019.

Kristy Choi, Kedar Tatwawadi, Aditya Grover, Tsachy Weissman, and Stefano Ermon. Neural joint source-channel coding. In *ICML*, pages 1182–1192. PMLR, 2019.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

David Roxbee Cox. *Analysis of survival data*. Routledge, 2018.

Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

Donglin Di, Shengrui Li, Jun Zhang, and Yue Gao. Ranking-based survival prediction on histopathological whole-slide images. In *MICCAI*, pages 428–438. Springer, 2020.

P Erdos and A Renyi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

David Faraggi and Richard Simon. A neural network model for survival data. *Statistics in medicine*, 14(1):73–82, 1995.

Tamara Fernández, Nicolás Rivera, and Yee Whye Teh. Gaussian processes for survival analysis. In *NeurIPS*, pages 5021–5029, 2016.

John A Foekens, Harry A Peters, Maxime P Look, Henk Portengen, Manfred Schmitt, Michael D Kramer, Nils Brünner, Fritz Jänicke, Marion E Meijer-van Gelder, Sonja C Henzen-Logmans, et al. The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients. *Cancer research*, 60(3):636–643, 2000.

Garima Gupta, Vishal Sunder, Ranjitha Prasad, and Gautam Shroff. Cresa: A deep learning approach to competing risks, recurrent event survival analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 108–122. Springer, 2019.

Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

Jared L Katzman, Uri Shaham, Alexander Cloninger, et al. DeepSurv: Personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):24, 2018.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Sachin Kumar, Garima Gupta, Ranjitha Prasad, Arnab Chatterjee, Lovekesh Vig, and Gautam Shroff. Camta: Casual attention model for multi-touch attribution. *DMS Workshop, ICDM*, 2020.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of machine learning research*, 20(129):1–30, 2019.

Changhee Lee, William R Zame, Jinsung Yoon, and Mihaela van der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *Proc. AAAI*, 2018.

Antonio G Marques, Negar Kiyavash, José MF Moura, Dimitri Van De Ville, and Rebecca Willett. Graph signal processing: Foundations and emerging directions [from the guest editors]. *IEEE Signal Processing Magazine*, 37(6):11–13, 2020.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

M Schumacher, G Bastert, H Bojar, K Huebner, M Olschewski, W Sauerbrei, C Schmoor, C Beyerle, RL Neumann, and HF Rauschecker. Randomized 2x2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. german breast cancer study group. *Journal of Clinical Oncology*, 12(10):2086–2093, 1994.

Marco Scutari. Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*, 2009.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *NIPS*, 28:3483–3491, 2015.

Ping Wang, Yan Li, and Chandan K Reddy. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

Jung Yoon, Renjie Liao, Yuwen Xiong, Lisa Zhang, Ethan Fetaya, Raquel Urtasun, Richard Zemel, and Xaq Pitkow. Inference in probabilistic graphical models by graph neural networks. In *Asilomar Conference*, pages 868–875, 2019.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proc. ICML*, volume 97, pages 7154–7163. PMLR, 2019.