Adversarial Time-to-Event Modeling

Paidamoyo Chapfuwa ¹  Chenyang Tao ¹  Chunyuan Li ¹  Courtney Page ¹  Benjamin Goldstein ¹  Lawrence Carin ¹  Ricardo Henao ¹

Abstract

Modern health data science applications leverage abundant molecular and electronic health data, providing opportunities for machine learning to build statistical models to support clinical practice. Time-to-event analysis, also called survival analysis, stands as one of the most representative examples of such statistical models. We present a deep-network-based approach that leverages adversarial learning to address a key challenge in modern time-to-event modeling: nonparametric estimation of event-time distributions. We also introduce a principled cost function to exploit information from censored events (events that occur subsequent to the observation window). Unlike most time-to-event models, we focus on the estimation of time-to-event distributions, rather than time ordering. We validate our model on both benchmark and real datasets, demonstrating that the proposed formulation yields significant performance gains relative to a parametric alternative, which we also propose.

1. Introduction

Time-to-event modeling is one of the most widely used statistical analysis tools in biostatistics and, more broadly, health data science applications. For a given subject, these models estimate either a risk score or the time-to-event distribution, from a pre-specified point in time at which a set of covariates (predictors) are observed. In practice, the model is parameterized as a weighted, often linear, combination of covariates. Time is estimated parametrically or nonparametrically, the former by assuming an underlying time distribution and the latter as proportional to observed event times. These models have been widely used in risk profiling (Hippisley-Cox & Coupland, 2013; Cheng et al., 2013), treatment planning, and drug development (Fischl et al., 1987). Time-to-event modeling, and in a larger context, point processes, constitute the fundamental analytical tools in applications for which the future behavior of a system or individual is to be characterized statistically.

The principal time-to-event modeling tool is the Cox Proportional Hazards (Cox-PH) model (Cox, 1992). Cox-PH is a semi-parametric model that assumes the effect of covariates is a fixed, time-independent, multiplicative factor on the hazard rate, which characterizes the instantaneous death rate of the surviving population. By optimizing a partial likelihood formulation, Cox-PH circumvents the difficulty of specifying the unknown, time-dependent, baseline hazard function. Consequently, Cox-PH results in point-estimates proportional to the event times. Further, estimation of Cox-PH models depends heavily on event ordering and not the time-to-event itself, which is known to compromise the scalability of the estimation procedure to large datasets. This poor scaling behavior is manifested because the formulation is not amenable to stochastic training with minibatches.

It is well accepted that the fixed-covariate-effects assumption made in Cox-PH is strong, and unlikely to hold in reality (Aalen, 1994). For instance, individual heterogeneity and other sources of variation, often likely to be dependent on time, are rarely measured or totally unobservable. This unobservable variation has been gradually recognized as a major concern in survival analysis and cannot be safely ignored (Collett, 2015; Aalen et al., 2001). When these sources of variation are independent of time, they can be modeled via fixed or random effects (Aalen, 1994; Hougaard, 1995). However, in cases for which they render the hazard rate time-dependent, such variation is difficult to control, diagnose, or model parametrically. Cox-PH is known to be sensitive to such assumption violations (Aalen et al., 2001; Kleinbaum & Klein, 2010). Moreover, Cox-PH focuses on the estimation of the covariate effects rather than the survival time distribution, i.e., time-to-event prediction. The motivation behind Cox-PH and its shortcomings make it less appealing in applications where prediction is of highest importance.

An alternative to the Cox-PH model is the Accelerated Failure Time (AFT) model (Wei, 1992). AFT makes the simplifying assumption that the effect of covariates either accelerates or delays the event progression, relative to a parametric...
baseline time-to-event distribution. However, by not making
the baseline hazard a constant, as in standard Cox-PH,
AFT is often a more reasonable assumption in clinical set-
tings when predictions are important (Wei, 1992). AFT also
encompasses a wide range of popular parametric propor-
tional hazards models and proportional odds models, when
the event baseline time distribution is specified properly
(Klein & Moeschberger, 2005). Learning in AFT models
falls into the category of maximum likelihood estimation,
and therefore it scales well to large datasets, when trained
via stochastic gradient descent. Further, AFT is also more
robust to unobserved variation effects, relative to Cox-PH
(Keiding et al., 1997).

From a machine learning perspective, recent advances in
deep learning are starting to transform clinical practice.
Equipped with modern learning techniques and abundant
data, machine-learning-driven diagnostic applications have
surpassed human-expert performance in a wide array of
health care applications (Cheng et al., 2013; 2016; Havaei
et al., 2017; Djuric et al., 2017; Gulshan et al., 2016). How-
ever, applications involving time-to-event modeling have
been largely under-explored. From the existing approaches,
most focus on extending Cox-PH with nonlinear neural-
network-based covariate mappings (Katzman et al., 2016;
Faraggi & Simon, 1995; Zhu et al., 2016), casting the time-
to-event modeling as a discretized-time classification prob-
lem (Yu et al., 2011; Fotso, 2018), or introducing a nonlinear
map between covariates and time via Gaussian processes
(Fernández et al., 2016; Alaa & van der Schaar, 2017).

Generative Adversarial Networks (GANs) (Goodfellow
et al., 2014) have recently demonstrated unprecedented poten-
tial for generative modeling, in settings where the goal is
to estimate complex data distributions via implicit sampling.
This is done by specifying a flexible generator function,
usually a deep neural network, whose samples are adversar-
ially optimized to match in distribution to those from real
data. Successful examples of GAN include generation of
images (Radford et al., 2015; Salimans et al., 2016), text
(Yu et al., 2017; Zhang et al., 2017) and data conditioned
on covariates (Isola et al., 2016; Reed et al., 2016). However,
ideas from adversarial learning are yet to be exploited for
the challenging task that is time-to-event modeling.

Previous work often represents time-to-event distributions
using a limited family of parametric forms, e.g., log-normal,
Weibull, Gamma, Exponential, etc. It is well understood
that parametric assumptions are often violated in practice,
largely because of the model is unable to capture unobserved
(nuisance) variation. This fundamental shortcoming is one
of the main reasons why non-parametric methods, e.g., Cox
proportional hazards, are so popular. Adversarial learning
leverages a representation that implicitly specifies a time-
to-event distribution via sampling, rather than learning the
parameters of a pre-specified distribution. Further, GAN-
learning penalizes unrealistic samples, which is a known
issue in likelihood-based models (Karras et al., 2018).

The work presented here seeks to improve the quality of
the predictions in nonparametric time-to-event models. We
propose a deep-network-based nonparametric time-to-event
model called a Deep Adversarial Time-to-Event (DATE)
model. Unlike existing approaches, DATE focuses on the
estimation of time-to-event distributions, rather than event
ordering, thus emphasizing predictive ability. Further, this is
done while accounting for missing values, high-dimensional
data and censored events. The key contributions associated
with the DATE model are: (i) The first application of GANs
to nonlinear and nonparametric time-to-event modeling, con-
ditioned on covariates. (ii) A principled censored-event-
aware cost function that is distribution-free and independent
of time ordering. (iii) Improved uncertainty estimation via
deep neural networks with stochastic layers. (iv) An alterna-
tive, parametric, non-adversarial time-to-event AFT model
to be used as baseline in our experiments. (v) Results on
benchmark and real data demonstrate that DATE outper-
forms its parametric counterpart by a substantial margin.

2. Background

Time-to-event datasets are usually composed of three
variables \{x_i, t_i, l_i\}_{i=1}^N = D, the covariates \(x_i = [x_{i1},
..., x_{ip}] \in \mathbb{R}^p\) and event pairs \((t_i, l_i)\), where \(t_i\) is
the time-to-event of interest and \(l_i \in \{0, 1\}\) is a binary cen-
soring indicator. Typically, \(l_i = 1\) indicates the event is
observed, alternatively, \(l_i = 0\) indicates censoring at \(t_i\)
when \(l_i = 0\), an event, if it occurs, transpires after the
observation window that ends at time \(t_i\). \(N\) denotes the
size of the dataset.

Time-to-event models characterize the survival function:
\[
S(t) = P(T > t) = 1 - F(t) = \exp \left(-\int_0^t h(s)ds\right),
\]
defined as the fraction of the population that survives up to
time \(t\), or the conditional hazards rate function \(h(t|x)\) (see
below), defined as the instantaneous rate of occurrence of
an event at time \(t\) given covariates \(x\). We derive the rela-
tionship between \(h(t|x)\) and \(S(t)\) from standard definitions
(Kleinbaum & Klein, 2010):
\[
h(t|x) = \lim_{dt \to 0} \frac{P(t < T < t + dt|x)}{P(T > t|x)dt} = \frac{f(t|x)}{S(t|x)},
\]
where \(f(t|x)\) is the conditional survival density function and
\(S(t|x)\) is formally the complement of the cumulative con-
ditional density function \(F(t|x)\). To solve the hazard-rate
where \( L \) in Cox-PH, then we can write \( f \) satisfies coefficients. If the conditional survival density function satisfies of \( \psi \) as standard Cox-PH. See Bender et al. (2005) for specific time-to-event, however, not nearly as widely adopted for 

\[
S(t|x) = \exp(-H(t|x)), \quad f(t|x) = h(t|x)S(t|x).
\]

See Supplementary Material for examples of parametric time-to-event characterizations. Time-to-event models, e.g., Cox-PH and AFT, leverage the results in (2) and (3), to characterize the relationship between covariates \( x \) and time-to-event \( t \), when estimating the conditional hazard function \( h(t|x) \). Two popular frameworks, Cox-PH and AFT, approach the estimation of \( h(t|x) \) using nonparametric and parametric techniques, respectively.

**Cox Proportional Hazard** The Cox-PH (Cox, 1992) model is a semi-parametric, linear model where the conditional hazard function \( h(t|x) \) depends on time through the baseline hazard \( h_0(t) \), and independent of covariates \( x \) as

\[
h(t|x) = h_0(t) \exp(x^T \beta). \tag{4}
\]

Provided with the \( N \) observation tuples in \( \mathcal{D} \), Cox-PH estimates the regression coefficients, \( \beta \in \mathbb{R}^p \), that maximize the partial likelihood (Cox, 1992):

\[
\mathcal{L}(\beta) = \prod_{i, l_i = 1} \frac{\exp(x_{i, l_i}^T \beta)}{\sum_{j: t_j \geq t_i} \exp(x_{j}^T \beta)}, \tag{5}
\]

where \( \mathcal{L}(\beta) \) is independent of the baseline hazard in (4). Note also that (5) only depends on the ordering of \( t_i \), for \( i, \ldots, N \), and not their actual values. Cox-PH is nonparametric in that it estimates the ordering of the events, not their times, thus avoiding the need to specify a distribution for \( h_0(t) \). Several techniques have been developed that assume a parametric distribution for \( h_0(t) \), in order to estimate the actual time-to-event, however, not nearly as widely adopted as standard Cox-PH. See Bender et al. (2005) for specifications of \( h_0(t) \) that result in an exponential, Weibull or Gompertz survival density functions. For example, when

\[
f(t) = \lambda \exp(-\lambda t), \quad i.e., \exp(-x^T \beta), \tag{6}
\]

where \( t \sim p_0(t) \) is the unmoderated time, thus \( \xi \) characterizes the baseline survival density distribution. Note the similarity between (4) and (6) despite the differences in their motivation. Different choices of baseline distribution yield a variety of AFT distributions, including Weibull, log-normal, gamma and inverse Gaussian (Klein & Moeschberger, 2005).

Intuitively, AFT assumes the effect of the covariates, \( \psi(x) \), accelerates or delays the life course, which is often meaningful in a clinical or pharmaceutical setting, and sometimes easier to interpret compared with Cox-PH (Wei, 1992). Empirical evidence has shown that AFT is more robust to missing values and misspecification of the survival function than Cox-PH (Keiding et al., 1997). Gaussian-process-based AFT models (Fine & Gray, 1999; Alaa & van der Schaar, 2017) have been used to model competing risk applications with success.

### 3. Adversarial Time-to-Event

We develop a nonparametric model for \( p(t|x) \), where \( t \) is the (non-censored) time-to-event from the time at which covariates \( x \) were observed. More precisely, we learn the ability to sample from \( p(t|x) \) via approximation \( q(t|x) \). Further, we do so without specifying a distribution for the marginal (baseline survival distribution), \( p_0(t) \), which in AFT is usually assumed log-normal. Like in Cox-PH and AFT, we assume that \( p_0(t) \) is independent of covariates \( x \).

For censored events, \( l_i = 0 \), we wish the model to have a high likelihood for \( p(t > t_i|x_i) \), while for non-censored events, \( l_i = 1 \), we wish that the pairs \( \{x_i, t_i\} \) be consistent with data generated from \( p(t|x)p_0(x) \), where \( p_0(x) \) is the (empirical) marginal distribution for covariates, from which we can sample but whose explicit form is unknown.

We consider a conditional generative adversarial network (GAN) (Goodfellow et al., 2014), in which we draw approximate samples from \( p(t|x) \), for \( l_i = 1 \), as

\[
t = G_\theta(x; \epsilon; l = 1), \quad \epsilon \sim p_\epsilon(\epsilon), \tag{7}
\]

where \( p_\epsilon(\epsilon) \) is a simple distribution, e.g., isotropic Gaussian or uniform (discussed below). The generator, \( G_\theta(x; \epsilon; l = 1) \) is a deterministic function of \( x \) and \( \epsilon \), specified as a deep neural network with model parameters \( \theta \), that implicitly defines \( q_\theta(t|x; l = 1) \) in a nonparametric manner. We explicitly note that \( l = 1 \), to emphasize that all \( t \) drawn from this model are event times (non-censored times). Ideally the pairs \( \{x, t\} \) manifested from the model in (7) are indistinguishable from the observed data \( \{x, t, l = 1\} \in \mathcal{D}, i.e., \) the non-censored samples.

Let \( \mathcal{D}_{nc} \subset \mathcal{D} \) and \( \mathcal{D}_c \subset \mathcal{D} \) be the disjoint subsets of non-censored and censored data, respectively. Given a discriminator function \( D_\phi(x, t) \) specified as a deep neural network with model parameters \( \phi \). The cost function based on the
We also leverage the censored data
where
\[ \max(0, \theta) p \]
The complete cost function is
\[ \ell \]
The time domains for events we set \( d \) that penalizes the costs in (8) and (9) underrepresent the desire that time-
For non-censored data, the formulation in (8) is the standard conditional GAN.

We also leverage the censored data \( D_c \) to inform the parameters \( \theta \) of generative model \( G \theta(x, \epsilon; l = 0) \). We therefore consider the additional cost function
\[ \ell_2(\theta; D_c) = \]
where \( \max(0, \cdot) \) encodes that \( G \theta(x, \epsilon; l = 0) \) incurs no cost as long as the sampled time is larger than the censoring point. Further, \( p_c(t, x) \) is the empirical joint distribution responsible for \( D_c \), from which samples \( \{t, x\} \) are drawn to approximate the expectation in (9). Note that \( \max(0, \cdot) \) is one of many choices; smoothed or margin-based alternatives may be considered, but are not addressed here, for simplicity.

For cases in which the proportion of observed events is low, the costs in (8) and (9) underrepresent the desire that time-
events must be as close as possible to the ground truth, \( t \). For this purpose, we also impose a distortion loss \( d(\cdot, \cdot) \)
\[ \ell_3(\theta; D_{nc}) = \]
that penalizes \( G \theta(x, \epsilon; l = 1) \) for not being close to the event time \( t \) for non-censored events only. In the experiments, we set \( d(a, b) = \|a - b\|_1 \).

The complete cost function is
\[ \ell(\theta, \phi; D) = \]
where \( \{\lambda_2, \lambda_3\} > 0 \) are tuning parameters controlling the trade-off between non-censored and censored cost functions relative to the discriminator objective in (8). In our experiments we set \( \lambda_2 = \lambda_3 = 1 \), provided that (9) and (10) are written in terms of expectations, thus already account for the proportion differences in \( D_c \) and \( D_{nc} \). However, this may not be sufficient in heavily imbalanced cases or when the time domains for \( D_c \) and \( D_{nc} \) are very different.

The cost function in (11) is optimized using stochastic gradient descent on minibatches from \( D \). We maximize \( \ell(\theta, \phi; D) \) wrt \( \phi \) and minimize it wrt \( \theta \). The terms

\[ \ell_1(\theta, \phi; D_{nc}) = \]

\[ + \]
Here we take a simple approach consisting on adding sources of stochasticity to every layer of the generator as
\[ h_j = g(W_{j0}h_{j-1} + W_{j1}e_j), \]
where \( j = 1, \ldots, L \) and \( L \) is the number of layers. By doing this, we encourage increased coverage on the event times produced by the generator, without substantially changing the model or the learning procedure. In the experiments, we use a multivariate uniform distribution, \( e_j \sim \text{Uniform}(0, 1) \), for \( j = 1, \ldots, L \), over Gaussian to reduce cancelling effects. As we show empirically, this approach produces substantially better coverage compared to having noise only on the input layer and without the convergence issues associated with the additional stochasticity.

In Figure 1 we illustrate the contribution of the noise on each layer to the distribution of event times. In this example, we show 10 test-set estimated time-to-event distributions using a 2-layer model with noise sources in all layers, including the input. We see that ground-truth times are nicely covered by the estimated distributions. Also, that the combination of noise sources, rather than any individual source, jointly contribute to the desired distribution coverage. Additional examples, including censored times, are found in the Supplementary Material.

4. Baseline AFT Model

In the experiments, we will demonstrate the capabilities of the DATE model. However, since there are not many scalable nonlinear time-to-event models focused on event time estimation, below we present a parametric (non-adversarial) model to be used in the experiments as baseline. We do so with the goal of providing a fair comparison and, at the same time, to highlight the distinguishing factors of the DATE model. Starting from (6), we consider the following MLP-based log-normal AFT:

\[
\log t = \mu_\beta(x) + \xi, \quad \xi \sim \mathcal{N}(0, \sigma_\beta^2(x)),
\]

where \( \mu_\beta(x) \) and \( \sigma_\beta^2(x) \) are MLPs parameterized by \( \beta \), representing the mean and variance of the log-transformed time-to-event as a function of covariates \( x \). For convenience, we adopt the log-normal distribution for event time \( t \), mainly because we found that it is considerably more stable during optimization, compared to other popular survival distributions, e.g., Weibull or Gamma.

Note that (12) is very different from (7), where the generator implicitly defines the distribution \( q_\theta(t|x) \). The likelihood function of the log-normal AFT model in (12) for all events (censored and non-censored) is then

\[
\prod_{i}^{N} p(t_i|x_i) = \prod_{i:t_i=1} f_\beta(t_i|x_i) \prod_{i:t_i=0} S_\beta(t_i|x_i) \prod_{i:t_i=1} \phi(\nu(t_i,x_i)) \prod_{i:t_i=0} (1 - \Phi(\nu(t_i,x_i))),
\]

where \( \phi(\cdot) \) and \( \Phi(\cdot) \) are the Gaussian density and cumulative density functions, respectively, and \( \nu(t_i,x_i) = (\log t_i - \mu_\beta(x_i))/\sigma_\beta(x_i) \). The likelihood in (13) is convenient, because it allows estimation of time-to-event, while seemingly accounting for censored events. The latter comes as benefit of having a parametric model with closed-form cumulative density function.

The cost function \( \mathcal{L}(\beta; D) \) for the Deep Regularized AFT (DRAFT) model is written as:

\[
\mathcal{L}(\beta; D) = \log p(t|D) + \eta R(\beta; D),
\]

where the first term is the negative log-likelihood loss from (13), \( \eta > 0 \) is a tuning parameters, and \( R(\beta; D) \) is a regularization loss that encourages event times to be properly ordered. Specifically, we use the following cost function adapted from Steck et al. (2008):

\[
R(\beta; D) = \frac{1}{|E|} \sum_{i,j=1} |t_j > t_i| \frac{1}{\log 2} \left( |D| - 1 - \log \Phi(\beta - D) \right),
\]

where \( E \) is the set of all pairs \( \{i, j\} \) in \( \{1, \ldots, N\} \) for which the second argument is observed, i.e., \( t_i = 1 \), and \( \Phi(\cdot) \) is the sigmoid function. The cost function above is a lower bound on the Concordance Index (CI) (Harrell et al., 1984), which constitutes a difficult-to-optimize discrete objective, that is widely used as performance metric for survival analysis, precisely because it captures time-to-event order. Further, it is reminiscent of the partial-likelihood of Cox-PH in (5), but is more amenable to stochastic training.

5. Related Work

Deep learning models, specifically MLPs, have been successfully integrated with Cox-PH-based objectives to improve risk estimation in time-to-event models. Faraggi & Simon (1995) proposed an neural-network-based model optimized using the standard partial-likelihood cost function from Cox-PH. Katzman et al. (2016) is similar to Faraggi & Simon (1995), but leverages modern deep learning techniques such as weight decay, batch normalization and dropout. Luck et al. (2017) replaced the partial-likelihood formulation in (5) with Efron’s approximation (Efron, 1977) and an isotonic regression cost function adapted from Menon et al. (2012) to handle censored events. Zhu et al. (2016) proposed a time-to-event model for image covariates based on convolutional networks.

From the Gaussian process literature, Fernández et al. (2016) proposed a time-to-event model inspired by a Poisson process, where the nonlinear map between covariates and time is modeled as a Gaussian process on the Poisson rate. More recently, Alaa & van der Schaar (2017) proposed a deep multi-task Gaussian process model for survival analysis with competing risks, and learned via variational inference.
Following a different path, other approaches recast the time-to-event problem as a classification task. Yu et al. (2011) proposed a linear model where (discretized) time is estimated using a sequence of dependent regressors. More recently, Fotso (2018) extended their approach to a nonlinear mapping of covariates using deep neural networks. Generative approaches have also been proposed to infer survival-time distributions with variational inference. Deep Survival Analysis (DSA) (Ranganath et al., 2016) specifies a latent model that leverages deep exponential family distributions by DATE and DRAFT on \( \text{EHR} \) and \( \text{DRAFT} \), respectively, and use medians for quantitative results requiring point estimates, i.e., \( \hat{t} = \text{median}(\{t_s\}_{s=1}^{200}) \), where \( t_s \) is a sample from the trained model. Detailed network architectures, optimization parameters and initialization settings are in the Supplementary Material. TensorFlow code to replicate experiments can be found at https://github.com/paidamoyo/adversarial_time_to_event.

### 6. Experiments

The loss functions in (11) and (14) for DATE and DRAFT, respectively, are minimized via stochastic gradient descent. At test time, we draw 200 samples from (7) and (12) for DATE and DRAFT, respectively, and use medians for quantitative results requiring point estimates, i.e., \( t = \text{median}(\{t_s\}_{s=1}^{200}) \), where \( t_s \) is a sample from the trained model. Detailed network architectures, optimization parameters and initialization settings are in the Supplementary Material. TensorFlow code to replicate experiments can be found at https://github.com/paidamoyo/adversarial_time_to_event.

#### Comparison Methods

For non-deep learning based models, we considered arguably the two most popular approaches to time-to-event modeling, namely, (regularized) Cox-Efron and RSF. For deep learning models, we considered DRAFT, which generalizes existing neural-network-based methods by using both a parametric log-normal AFT objective and a non-parametric ordering cost function. Extending DRAFT to a mixture of log-normal distributions with different variances but shared mean, did not result in improved performance. We did not consider (variational) models for non-censored events only because learning from censored events is one of the main defining characteristics of time-to-event modeling (otherwise, the model essentially becomes non-negative regression). Further, in most practical situations the proportion of non-censored events is low, e.g., 24% in the EHR.

### Datasets

Our model is evaluated on 4 diverse datasets: i) FLCHAIN: a public dataset introduced in a study to determine whether non-clonal serum immunoglobulin free light chains are predictive of survival time (Dispensieri et al., 2012). ii) SUPPORT: a public dataset introduced in a survival time study of seriously-ill hospitalized adults (Knau et al., 1995). iii) SEER: a public dataset provided by the Surveillance, Epidemiology, and End Results Program. See (Ries et al., 2007) for details concerning the definition of the 10-year follow-up breast cancer subcohort used in our experiments. iv) EHR: a large study from Duke University Health System centered around inpatient visits due to comorbidities in patients with Type-2 diabetes.

The datasets are summarized in Table 1, where \( p \) denotes the number of covariates to be analyzed, after one-hot-encoding for categorical (cat) variables. Events indicates the proportion of the observed events, i.e., those for which \( l_i = 1 \). NaN indicates the proportion of missing entries in the \( N \times p \) covariate matrix and \( t_{\text{max}} \) is the time range for both censored and non-censored events.

#### Details about the public datasets: FLCHAIN, SUPPORT and SEER, including preprocessing procedures, can be found in the references provided above. EHR is a study designed to track primary care encounters of 19,064 Type-2 diabetes patients over a period of 10 years (2007-2017). The purpose of the analysis is to predict diabetes-related causes of hospitalization within 1 year of an EHR-recorded primary care encounter. Data is processed and analyzed at the patient encounter-level. The total number of encounters is \( N = 394,823 \). To avoid bias due to multiple encounters per patient, we split the training, validation and test sets so that a given patient can only be in one of the sets. The covariates, collected over a period of a year before the primary care encounter of interest, consist of a mixture of continuous and categorical summaries extracted from electronic health records: vitals and labs (minimum, maximum, count and mean values); comorbidities, medications and procedures ICD-9/10 codes (binary indicators and counts); and demographics (age, gender, race, language, smoking indicator, type of insurance coverage, as either continuous or categorical variables).

#### 6.1. Qualitative results

First, we visually compare the test-set time-to-event distributions by DATE and DRAFT on FLCHAIN data. In Figure 2 we show the top best (left) and worst (middle-left)
predictions on censored events, and the top best (middle-right) and worst (right) predictions on non-censored events. Circles denote ground-truth events or censoring points, while box-plots represent distributions over 200 samples for both DATE and DRAFT models. We see that: (i) in nearly every case, DATE is more accurate than DRAFT, (ii) DRAFT tends to make predictions outside the event range ($t_{\text{max}} = 5, 215$ days), denoted as a horizontal dashed line. (iii) DRAFT tends to overestimate the variance of its predictions, approximately by one order of magnitude relative to DATE. This is not very surprising as DRAFT has an MLP dedicated to estimate, conditioned on the covariates, the variance of the time-to-event distribution. However, note that variances estimated well over the domain of the events ($t_{\text{max}}$) are not necessarily meaningful or desirable. Figures with similar findings for the other 3 datasets can be found in the Supplementary Material.

To provide additional insight into the performance of DATE compared to DRAFT, we report the Normalized Relative Error (NRE) defined as $(\hat{t} - t)/t_{\text{max}}$ and $\min(0, t - \hat{t})/t_{\text{max}}$ for non-censored and censored events, respectively, where $t$, $\hat{t}$ and $t_{\text{max}}$ denote the ground-truth time-to-event, median time estimated (from samples) and event range, as indicated in Table 1. The NRE distribution provides a visual representation of the extent of test-set errors, while revealing whether the models are biased toward either overestimating ($\hat{t} > t^*$) or underestimating ($t^* > \hat{t}$) the event times. Although models with unbiased NREs are naturally preferred, in most clinical applications where being conservative is important, overestimated time-to-events must be avoided as much as possible. Figure 3 shows NRE distributions for test-set non-censored events on SUPPORT and EHR data. We see that DRAFT results in a considerable amount of errors beyond the event range ($|\text{NRE}| > 1$), $t_{\text{max}} = 120$ months or $t_{\text{max}} = 365$ days for SUPPORT and EHR, respectively. Further, we see that the NRE distribution for DRAFT is heavily skewed toward $\text{NRE} > 1$, thus tending to overestimate event times. On the other hand, DATE produces errors substantially more concentrated around 0 and within $|\text{NRE}| < 1$, relative to DRAFT. This demonstrates the advantage of the adversarial method DATE over the likelihood-based method DRAFT in generating realistic samples. Similar results were observed on the other datasets for both censored and non-censored events. See Supplementary Material for additional figures.

6.2. Quantitative results

Relative absolute error The performance of DATE is evaluated in terms of absolute error relative to the event range, i.e., $|\hat{t} - t|/t_{\text{max}}$. For censored events, the relative error is defined as $\max(0, t - \hat{t})/t_{\text{max}}$ to account for the fact that no error is made as long as $t \leq \hat{t}$. Table 2 shows median and 50% empirical intervals for relative absolute errors on non-censored events, on all test-data. Results on censored data are small and comparable across approaches, and are thus presented in the Supplementary Material. Specifically, we see that DATE outperforms DRAFT in 3 out of 4 cases by a substantial margin, and is comparable on the SUPPORT data. For instance, on EHR data 75% of all DATE test-set predictions have a relative absolute error less than 43% (approx. 156 days) which is substantially better than the 81% (approx. 295 days) by DRAFT.

Missing data Since missing data are common in clinical data, e.g., SEER data contains 23.4% missing values, we also consider a modified version of DATE, where the generator in (7) takes the form $t = G_{\theta}(z, \epsilon, l = 1)$, where $z$ is modeled as an adversarial autoencoder (Dumoulin et al., 2016; Li et al., 2017; Pu et al., 2017) with an encoder/decoder.
Table 2. Median relative absolute errors (as percentages of $t_{max}$), on non-censored data. Ranges in parentheses are 50% empirical ranges over (median) test-set predictions.

<table>
<thead>
<tr>
<th></th>
<th>DATE</th>
<th>DATE-AE</th>
<th>DRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EHR</td>
<td>23.6 (11.1, 43.0)</td>
<td>24.5 (12.4, 44.0)</td>
<td>36.7 (16.1, 81.3)</td>
</tr>
<tr>
<td>FLCHAIN</td>
<td>19.5 (9.5, 31.1)</td>
<td>19.3 (8.9, 32.4)</td>
<td>26.2 (9.0, 53.5)</td>
</tr>
<tr>
<td>SUPPORT</td>
<td>2.7 (0.4, 16.1)</td>
<td>1.5 (0.4, 19.2)</td>
<td>2.0 (0.2, 35.3)</td>
</tr>
<tr>
<td>SEER</td>
<td>18.6 (8.3, 34.1)</td>
<td>20.2 (10.3, 35.8)</td>
<td>23.7 (9.5, 51.2)</td>
</tr>
</tbody>
</table>

We also tried to quantify statistically the match between FLCHAIN and DATE. See Supplementary Material for additional details. This model, denoted in Table 2 as DATE-AE, does not require missing covariates to be imputed before hand, which is the case of DATE and DRAFT as specified originally. The results show no substantial performance improvement by DATE-AE, relative to DATE: results indicate that all of these approaches (DRAFT, DATE and DATE-AE) are robust to missing data. As a benchmark, we took FLCHAIN and SUPPORT, to then artificially introduced missing values ranging in proportion from 20% to 50%. Results in Supplementary Material support the idea of robustness, since all three approaches resulted in median relative absolute errors within 1% of those in Table 2.

We also tried to quantify statistically the match between time-to-event samples generated from DATE and those from the empirical distribution of the data, using the distribution-free two-sample test based on Maximum Mean Discrepancy (MMD) proposed by Sutherland et al. (2017). Due to sample size limitations (number of non-censored events in the test-set) and high-variances on the $p$-value estimates, we could not reliably reject the hypothesis that real and DATE samples are drawn from the same distribution. We did confirm it for DRAFT, which is not surprising considering both qualitative and quantitative results discussed above.

Concordance Index The concordance Index (CI) (Harrell et al., 1984), which quantifies the degree to which the order of the predicted times is consistent with the ground truth, is the most well-known performance metric in survival analysis. Although not the focus of our approach, we compared DATE to DATE-AE, DRAFT, Random Survival Forests (Ishwaran et al., 2008), and Cox-PH (with Efron’s approximation (Efron, 1977)). We found all of these models to be largely comparable. The results, presented in the Supplementary Material, show DATE-AE and DATE-AE and DRAFT being the best-performing models on EHR and SUPPORT, respectively. On SEER, DATE-AE and DATE perform best, Cox-PH and RSF. Finally, on FLCHAIN, the smallest dataset, all methods perform about the same. Note that unlike Cox-PH and DRAFT in (5) and (14), DATE-AE does not explicitly encourage proper time-ordering on the objective function; it is consequently deemed a strength of the proposed GAN-based DATE model that it properly learns ordering, without needing to explicitly impose this condition when training. DATE does not have a clear advantage in terms of learning the correct order; however, we found empirically that adding an ordering cost function, $R(\beta; D)$ in (14), to DATE does not improve the results.

Distribution coverage We now demonstrate that the DATE model, with noise sources on all layers, has time-to-event distributions with larger variances than versions of DATE with noise only on the input of the neural network. Table 3 shows the median of the 95% intervals for all test-set time-to-event distributions on SUPPORT data. DATE with Uniform(0,1) has larger variance and coverage compared to the other alternatives, while keeping relative absolute errors and CIs largely unchanged (see Supplementary Material for details). We did not run models with Uniform(-1,1) and Gaussian(0,1) only on the output layer, because from the other results presented above it is clear that these two options are not nearly as good as having Uniform(0,1) noise on all layers. Note also that we did not include DRAFT in these comparisons. DRAFT has naturally good coverage due to the variance of the time-to-event distributions being modeled independent for each observation as a function of the covariates (see for instance Figure 2). However, DRAFT has difficulties keeping good coverage while maintaining good performance, i.e., small absolute relative error.

7. Conclusions

We have presented an adversarially-learned time-to-event model that leverages a distribution-free cost function for censored events. The proposed approach extends GAN models to time-to-event modeling with censored data, and it is based on deep neural networks with stochastic layers. The model yields improved uncertainty estimation relative to alternative approaches. As a baseline model for our experiments, we also proposed a parametric AFT-based with a parametric log-normal distribution on the time of event. To the best of our knowledge, this work is the first to leverage adversarial learning to improve estimation of time-to-event distributions, conditioned on covariates. Experimental results on challenging time-to-event datasets showed that DATE, our adversarial solution, consistently outperforms DRAFT, its parametric (log-normal) counterpart. As future work, we will extend DATE to models with competing risks and longitudinally measured covariates.

### Table 3. Median of 95% intervals for all test-set time-to-event distributions on SUPPORT data. Ranges in parentheses are 50% empirical quantiles.

<table>
<thead>
<tr>
<th></th>
<th>Uniform(-1,1)</th>
<th>Uniform(0,1)</th>
<th>Gaussian(0,1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-censored</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>60.0 (3.9, 176.5)</td>
<td>149.9 (8.5, 526.8)</td>
<td>37.9 (3.5, 237.4)</td>
</tr>
<tr>
<td>Input</td>
<td>28.9 (1.8, 114.8)</td>
<td>22.4 (1.5, 91.2)</td>
<td>33.7 (1.6, 127.6)</td>
</tr>
<tr>
<td>Output</td>
<td>——</td>
<td>168.8 (16.6, 844.3)</td>
<td>——</td>
</tr>
<tr>
<td>Censored</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>231.3 (177.2, 332.1)</td>
<td>1391.7 (900.0, 2000.1)</td>
<td>350.5 (254.4, 539.3)</td>
</tr>
<tr>
<td>Input</td>
<td>137.3 (99.4, 200.0)</td>
<td>86.9 (64.4, 135.0)</td>
<td>155.8 (106.7, 220.3)</td>
</tr>
<tr>
<td>Output</td>
<td>——</td>
<td>1158.5 (773.8, 1670.4)</td>
<td>——</td>
</tr>
</tbody>
</table>
Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments. This research was supported in part by DARPA, DOE, NIH, ONR, NSF and Sanoit. The authors would also like to thank Shuyang Dai, Liqun Chen, Rachel Ballantyne Draelos, Xilin Cecilia Shi, Yan Zhao and Dr. Neha Pagidipati, for fruitful discussions.

References


Bender, R., Augustin, T., and Blettner, M. Generating survival times to simulate Cox proportional hazards models. *Statistics in medicine*, 2005.


Adversarial Time-to-Event Modeling


