Adaptive Discretization for Event PredicTion (ADEPT)

Jimmy Hickey[1]   **Ricardo Henao**[2,5]   **Daniel Wojdyla**[3]
Michael Pencina[4,5]   **Matthew Engelhard**[4,5]
[1]North Carolina State University   [2]King Abdullah University of Science and Technology
[3]Duke Clinical Research Institute   [4]Duke AI Health   [5]Duke University School of Medicine

Abstract

Recently developed survival analysis methods improve upon existing approaches by predicting the probability of event occurrence in each of a number pre-specified (discrete) time intervals. By avoiding placing strong parametric assumptions on the event density, this approach tends to improve prediction performance, particularly when data are plentiful. However, in clinical settings with limited available data, it is often preferable to judiciously partition the event time space into a limited number of intervals well suited to the prediction task at hand. In this work, we develop Adaptive Discretization for Event PredicTion (ADEPT) to learn from data a set of cut points defining such a partition. We show that in two simulated datasets, we are able to recover intervals that match the underlying generative model. We then demonstrate improved prediction performance on three real-world observational datasets, including a large, newly harmonized stroke risk prediction dataset. Finally, we argue that our approach facilitates clinical decision-making by suggesting time intervals that are most appropriate for each task, in the sense that they facilitate more accurate risk prediction.

1 INTRODUCTION

Time to event modeling, also called survival analysis, is ubiquitous throughout clinical medicine as well as in many other fields concerned with predicting risk of events of interest (*e.g.*, clinical outcomes) based on

available features (*e.g.*, patient characteristics). Traditional approaches include the well-known Cox proportional hazards (Cox-PH) model [Cox, 1972], in which features modulate a baseline hazard rate; and the accelerated failure time (AFT) model [Wei, 1992] model, in which features accelerate or decelerate a learned, parametric event time density.

Recently developed methods have focused on *a)* allowing effects of features on the hazard rate or event time density to be non-linear and flexible [Katzman et al., 2018, Ranganath et al., 2016, Kvamme et al., 2019, Miscouridou et al., 2018]; and *b)* also allowing greater flexibility in the form of the event time density itself via approaches that *discretize* time, then predict the probability of event occurrence in each resulting *time interval* [Yu et al., 2011, Lee et al., 2018, Ren et al., 2019, Tjandra et al., 2021, Engelhard and Henao, 2022].

The prognostic information provided by these models often has direct and significant impact on stakeholder decision-making. In a clinical setting, for example, information about risk within a particular time interval might influence providers' or patients' decisions about whether to pursue treatment, or which specific treatment to pursue. It is therefore critical not only that predictions are accurate, but also that they are easily interpretable by stakeholders who wish to integrate them in decision-making. The predictions of a Cox-PH model might be presented to stakeholders as relative hazards, for instance, whereas it is natural to present the predictions of more recent models as the probability of event occurrence in a time interval of interest.

Importantly, however, decisions about these intervals made during model development – in other words, choices about the number and placement of *cut points* used to discretize the event time space – can have substantial impact on interpretability as well as performance. Equipped with unlimited data, we might use a large number of cut points to divide the timeline into tiny intervals; this would then allow us to summarize risk over an arbitrary time period of interest

by combining predictions across all the intervals that comprise that period. However, the amount of data required to accurately estimate risk in each interval increases as the number of intervals increases, making this approach impractical even for large observational datasets. Equipped with unlimited time, on the other hand, we might present risk in a format most relevant to a particular patient, or to the decision at hand. Again, however, practical considerations typically require us to instead summarize risk over a consistent, limited number of time-frames (*e.g.*, 10-year risk, 5-year risk). In some cases a particular discretization is most actionable given the clinical context, but in others the choice is arbitrary, and it would be preferable to identify a discretization that facilitates more accurate prediction.

To illustrate the problem more concretely, consider the following example from the maternal health setting, which partly motivated this work. Patients with preeclampsia and gestational hypertension have substantially increased risk of postpartum cardiovascular events [Meng et al., 2022], but this risk can be mitigated by regular monitoring (*e.g.,* increased visits) of high-risk patients in the months after delivery. When developing a monitoring strategy, it is important to determine not only (a) which patients are at highest risk, but also (b) how long monitoring should take place; yet we have limited data available for learning because the outcome rates are low.

Our goal, therefore, is to develop a principled, data-driven approach to answer both of these questions. Specifically, we wish to develop a method that providers can use to identify time intervals that are optimal when *understanding* risk, for example to design an intervention or monitoring strategy, as well as when *reporting* risk to patients. At the same time, we wish to retain the substantial advantages and flexibility of other recently developed approaches, including their lack of strong parametric assumptions about the form of the event density. To solve this problem, we develop Adaptive Discretization for Event PredicTion (ADEPT).

We begin by recasting learning from *discrete* survival times as learning from *continuous* survival times under the assumption that the density is piecewise constant; and then formulate a smooth relaxation of this piecewise constant density that allows cut points (*i.e.*, interval boundaries) to be learned by gradient-based optimization methods. We then present our learning procedure and results of experiments with two simulated and three real datasets – including a newly harmonized stroke risk prediction dataset that pools data across three large cohorts – that illustrate the effectiveness and potential clinical relevance of ADEPT.
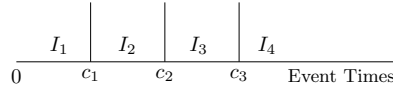

Figure 1: The event time space partitioned by three cut points into four intervals.

Our performance evaluation focuses on comparing our method to its state of the art alternative, namely, discrete-time, neural network-based risk prediction over fixed-length intervals.

In summary, our contributions are as follows:

- Present ADEPT, a novel model and associated learning procedure to learn an optimal event time partition from data rather than fixing it *a priori*.

- Present simulation results illustrating effective learning of cut points that are consistent with the true, underlying generative model.

- Demonstrate improved prediction performance across three real datasets, including two clinical datasets.

- Identify clinically meaningful risk cut points illustrating the potential of the approach to provide improved prognostic information.

2 METHODS

2.1 Setup and Notation

Consider a time-to-event outcome where each observation is represented by the triplet $\{\boldsymbol{X}, Y, S\}$, where $\boldsymbol{X} \in \mathcal{X} \in \mathbb{R}^p$ is a p-dimensional feature vector, $Y \in (0, T_{\max}]$ is an observed event time over a finite time horizon, and $S \in \{0, 1\}$ indicates whether Y is a right-censoring time ($S = 0$) or an event time ($S = 1$). The observed time Y is the minimum of the event time T and the right-censoring time U, *i.e.*, $Y = \min(T, U)$, and $S = \mathbb{I}(T < U)$, where the indicator function $\mathbb{I}(\cdot)$ is 1 when the argument is true and 0 otherwise.

We consider possible sequences of M *cut points* $C = \{c_j\}_{j=1}^M$, where $0 = c_0 < c_1 < \cdots < c_M < c_{M+1} = T_{\max}$, that partition the event time space, $(0, T_{\max}]$, into the intervals $I_1, \ldots, I_{M+1}$, where $I_j = (c_{j-1}, c_j]$. Figure 1 provides an example of the event time space partitioned into four intervals: $I_1 = (0, c_1]$, $I_2 = (c_1, c_2]$, $I_3 = (c_2, c_3]$, and $I_4 = (c_3, T_{\max}]$. Given such a partition, we introduce an auxiliary random variable $Z \in \{1, \ldots, M+1\}$ that indicates which interval contains T, *i.e.*, $Z = j \iff t \in I_j$.

2.2 Piecewise Constant Density

We begin by considering learning with fixed cut points, which is currently the predominant approach. For example, Lee et al. [2018] and other recently-developed methods [Ren et al., 2019, Tjandra et al., 2021, Engelhard and Henao, 2022] use fixed cut points to discretize time in order to avoid placing restrictive, parametric assumptions on the form of the event time density. Instead, the density is restricted to be piecewise constant according to the intervals defined by the cut points. The cut points themselves might be evenly spaced in time, or alternatively they might be evenly spaced across the observed or estimated event time distribution, $e.g.$, via empirical quantiles. The goal of learning is then to estimate $P(Z|\boldsymbol{X})$, the conditional probability that T will fall in each of the pre-defined intervals, rather than $p(T|\boldsymbol{X})$, the conditional density of T. Typically T is discretized to Z a $priori$.

However, it is not possible to $learn$ the cut points C with this approach, because Z depends on C in addition to T. To see this, consider the value of Z associated with an observed time $t \in (0, T_{\max}]$ under the binary partition defined by the single cut point c_1. If we choose $c_1 \geq t$, then $t \in (0, c_1]$, therefore $Z = 1$; but for $c_1 < t$, we have $t \in (c_1, T_{\max}]$, therefore $Z = 2$.

To circumvent this limitation, we note that estimating $P(Z|\boldsymbol{X})$ is equivalent to estimating $p(T|\boldsymbol{X})$ with the following piecewise constant model, which supposes $p(T|\boldsymbol{X})$ has uniform density over each interval I_j:

$$\hat{p}(t|\boldsymbol{x}) = \sum_{j=1}^{M+1} p_\phi(z_j|\boldsymbol{x}) \frac{\mathbb{I}_{I_j}(t)}{|I_j|}, \qquad (1)$$

where $\mathbb{I}_{I_j}(\cdot)$ is the indicator function associated with the interval I_j, and ϕ parameterizes our model of $P(Z|\boldsymbol{X})$. Importantly, we must normalize by $|I_j|$, the length of I_j, to ensure $\int_{(0,T_{\max}]} \hat{p}(t|\boldsymbol{x}) = 1$ and $\int_{I_j} \hat{p}(t|\boldsymbol{x}) = p_\phi(z_j|\boldsymbol{x})$. As a potential drawback of this approach, we note that whereas standard discrete-time approaches are well suited to handle outlying event times, in principle this normalization term could cause the loss to become numerically unstable in the case of extreme event time outliers.

2.3 Smooth Relaxation of Piecewise Density

The parameters ϕ of our model for Z can be learned directly from equation (1). However, our goal is to learn not only ϕ but also C, the specific partition that allows our model to best approximate $p(T|\boldsymbol{X})$ across a given dataset. Unfortunately, (1) cannot be optimized with respect to C via gradient-based methods. This is because the indicator function $\mathbb{I}_{I_j}(\cdot)$ implicitly depends on C, and is discontinuous whenever a cut point is equal to an observed event time.

To illustrate, consider learning a single cut point c_1 while holding the parameters ϕ fixed. For small ε such that $0 < \varepsilon < t$, where t is an observed event time associated with covariates $\boldsymbol{x}$, suppose the cut point $c_1 = t + \varepsilon$ is just after the observed event time. In this case, we have $t \in I_1$, therefore $\mathbb{I}_{I_1}(t) = 1$ and $\mathbb{I}_{I_2}(t) = 0$, and consequently $\hat{p}(t|\boldsymbol{x}) = p_\phi(z_1|\boldsymbol{x})/|I_1|$. On the other hand, suppose the cut point $c_1 = t - \varepsilon$ is just before the observed event time. In this case, we have $t \in I_2$, therefore $\mathbb{I}_{I_1}(t) = 0$ and $\mathbb{I}_{I_2}(t) = 1$, and consequently, $\hat{p}(t|\boldsymbol{x}) = p_\phi(z_2|\boldsymbol{x})/|I_2|$. Thus, for any non-trivial model p_ϕ for which $p_\phi(z_1|\boldsymbol{x}) \neq p_\phi(z_2|\boldsymbol{x})$, equation (1) is discontinuous at $c_1 = t$. This argument readily generalizes to all cut points.

To smooth this discontinuity and allow gradient-based optimization, we replace the indicator function $\mathbb{I}_{I_j}(t)$ in (1) with the smooth approximation $\sigma((t-c_{j-1})/\tau) * \sigma((c_j - t)/\tau)$, where $\sigma(z) = (1 + e^{-z})^{-1}$ is the sigmoid function. The temperature τ is a hyperparameter of the model that should be tuned based on the scale of the observed event times.

This results in the following relaxed model:

$$\hat{p}(t|\boldsymbol{x}) = \sum_{j=1}^{M+1} p_\phi(z_j|\boldsymbol{x}) \frac{\sigma(\frac{t-c_{j-1}}{\tau})\sigma(\frac{c_j-t}{\tau})}{|I_j|}, \qquad (2)$$

which is approximately piecewise constant for $\tau \ll T_{\max}$, yet differentiable everywhere with respect to C and thus suitable for gradient-based optimization.

2.4 Learning Procedure

Under the common assumption of non-informative right-censoring, we may ignore the censoring density and optimize $\hat{p}(y, s|\boldsymbol{x}; \theta)$, where $\theta = \{\phi, C\}$, over the observed data $\mathcal{D} = \{\boldsymbol{x}_i, y_i, s_i\}_{i=1}^N$ as follows:

$$\theta = \arg\max_\theta \sum_i^N \big[s_i \log \hat{p}(t_i|\boldsymbol{x}; \theta) \qquad (3)$$
$$+ (1 - s_i) \log \hat{P}(t_i > y_i|\boldsymbol{x}_i; \theta) \big],$$

where $\hat{P}(t_i > y_i|\boldsymbol{x}_i; \theta) = 1 - \int_0^T \hat{p}(\tau|\boldsymbol{x}_i; \theta)$ is the survival function associated with $\hat{p}(t_i|\boldsymbol{x}_i; \theta)$ for observation i.

However, optimizing equation (2) alone can result in degenerate solutions in which cut points become arbitrarily close together or even coincide. In the extreme case, it is possible to have $I_j = (0, T_{\max}]$ for a particular $j \in \{1, M + 1\}$, resulting in the trivial model in which $p_\phi(z|\boldsymbol{x})$ places all mass on z_j.

It is therefore critical to balance optimizing equation (2) versus ensuring that $p_\phi(z|\boldsymbol{x})$ is non-trivial. We accomplish this by incorporating a regularization term, $H(p_\phi(z|\boldsymbol{x}))$, with associated hyperparameter λ_1 in our optimization procedure. We use a scaled Beta$(1.5, 1.5)$ distribution on each cut point. For example, suppose there are three cut points $c_1 < c_2^\star < c_3$ where $c_2^\star$ is the newly proposed value for the middle cut point c_2. We scale the value of the cut point to find its location relative to the cut points near it: $c_{2,\text{scaled}}^\star = (c_2^\star - c_1)/(c_3 - c_1)$. The final regularization value is the PDF value of $c_{2,\text{scaled}}^\star$ evaluated over a Beta$(1.5, 1.5)$ distribution. This regularization term encourages cut points to be near the center of their two surrounding cut points.

We may then optimize θ over $\mathcal{D}$ by choosing $\theta = \arg\min_\theta \sum_\mathcal{D} \mathcal{L}(\theta)$, where $\mathcal{L}(\theta)$ is defined as follows:

$$\mathcal{L}(\theta) = -\log \hat{p}(y, s \mid \boldsymbol{x}; \theta) - \lambda_1 H(p_\phi). \quad (4)$$

Here the first term is the negative log likelihood in (3) and the second is our beta distribution regularizer. Our learning procedure then becomes:

$$\theta = \arg\min_\theta \sum_\mathcal{D} \mathcal{L}(\theta) + \lambda_2 R(\theta), \quad (5)$$

where we have included an additional regularization term $R(\cdot)$ (*e.g.*, L_2-regularization) along with an associated hyperparameter λ_2 to control for overfitting.

3 IMPLEMENTATION DETAILS

3.1 Baseline Model: Discrete-Time Neural Network

We compare ADEPT to a discrete-time neural network baseline that is identical to the proposed model, except the cut points (and corresponding intervals) are initialized based on the observed outcomes and remain fixed when learning the classifier. This approach, hereafter called the *DTNN*, was popularized by DeepHit [Lee et al., 2018] and is currently the predominant approach. However, to isolate the effect of learning the partition we do not include the ranking loss used in DeepHit . Our method and the DTNN have similar computational complexity, which is dominated by the computation of gradients with respect to model parameters θ rather than the cut points C. In this work, we instantiate p_θ as neural network, but our approach is flexible to the model choice; thus it can be changed based on application if, for example, interpretability is more important that predictive performance.

We initialize the DTNN model's cut points to be evenly spaced on the percentiles of the empirical Kaplan-Meier curve of the observed outcomes; for example, if there are three cut points then they would be placed at the time points associated with the 25th, 50th, and 75th percentiles on the estimated Kaplan-Meier curve. With these cut points fixed, we then build a model predicting the probability that the patient will experience the outcome in each interval. Note that this differs from ADEPT in which we also consider the cut points themselves as parameters. The DTNN classification model learns the probability of each observation being in each of the pre-defined intervals. In the notation of Section 2.2, the DTNN approach learns only the model parameters, ϕ_{baseline}, whereas ADEPT learns both model parameters ϕ and the cut points C. Importantly, we search the same grid of hyperparameters for the DTNN model as for ADEPT.

Due to its popularity, we also include a comparison to DeepSurv [Katzman et al., 2018]. However, understanding differences between DeepSurv and other approaches is challenging due to their stark differences, notably the assumption of proportional hazards, which can either improve or worsen performance depending on the degree to which this assumption holds. Moreover, because DeepSurv does not incorporate discretization, it will have fewer comparative performance metrics.

3.2 Performance Quantification

With simulated data we were able to judge the correctness of the estimated cut points by their proximity to the true cut points used in the data generation mechanism. For the real data we do not know the true cut point values and thus need other metrics to judge performance. While we focus on several metrics quantifying predictive performance, clinical collaboration is necessary to determine which metrics are most relevant in a particular clinical context, including when implementing treatment decisions.

Time-Dependent Concordance Index (CI) Since we consider a time-to-event outcome with censored observations rather than a regression or classification outcome, standard metrics such as root mean square error and area under the receiver operating characteristic are insufficient to capture prediction performance. Initially developed by Harrell Jr et al. [1984], the concordance index (CI) measures how well predicted event times match the order of the true event times. However, both ADEPT and the DTNN predict discrete interval membership rather than continuous event times, and the ordering of predicted risk can change over time. To properly account for these characteristics, we use a discrete-time implementation of the time-dependent concordance index developed by

Antolini et al. [2005]. This metric compares model-predicted risk at observed failure times to the model-predicted risks *at that time* for other individuals known to have later failure times. Pairs of individuals are only considered if (*a*) both failure times are known (neither are censored), or (*b*) one failure time is known to have occurred before the censoring time of the other.

AUC at last cut point The Area Under the Receiver Operating Characteristic Curve (AUC) is a common metric to evaluate predictive performance for a binary outcome. To adapt this to our method, we focus on the AUC at the last cut point. That is, we are interested in determining if the methods are able to predict whether an event happens before or after the final cut point. This is especially relevant for data sets with high amounts of censoring at the end of the study. The cases are all observations that experienced an event prior to the final cut point and the controls are all observations with an observed time (either an event or censored) after the final cut point. Notice that observations that are censored prior to the last cut point are omitted from this metric.

Integrated Brier Score (IBS) The Brier Score evaluated at time a chosen time t is the mean squared difference between the model-predicted cumulative event probability at time t and the true, binary, observation of whether the even occurred by t. The Integrated Brier Score improves upon this by integrating over all times $t \in \{t_{\min}, t_{\max}\}$ [Graf et al., 1999]. An IBS of 0 indicates that the model was able to perfectly predict outcomes.

Calibration slope and intercept We also consider the calibration slope and intercept as described by Crowson et al. [2016], which quantify the degree to which model-predicted probabilities accurately estimate true event probabilities, as determined based on observed event rates. A well calibrated model will have a calibration slope near 1 and a calibration intercept near 0.

It is important to note that DeepSurv does not discretize the event time space; because of this, only the AUC and IBS metrics are included. We cannot compare the discrete-time concordance index to the standard concordance index because there are inevitably more ties with the discrete-time approach.

3.3 Hyperparameter Tuning

ADEPT is flexible, allowing for any number of cut points. In our simulation examples we will know exactly how many cut points were used to generate the data; however, this is not the case for the real data experiments. Thus, we use 3, 5, and 10 cut points. We use a two layer neural network as our predictive model.
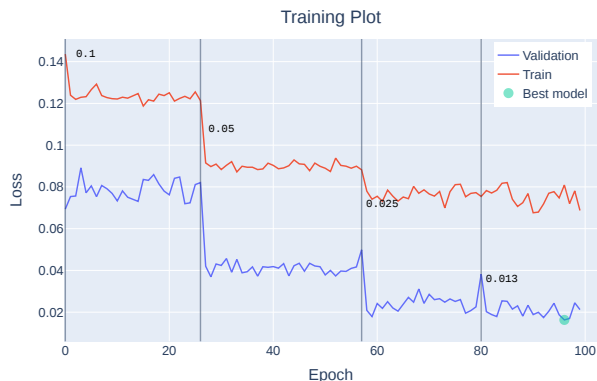

Figure 2: A training plot of the training and validation loss at each epoch for the two interval simulation example. The vertical lines represent drops in sigmoid temperature τ and the accompanying new value of τ.

The first layer has input dimension p based on the feature dimension of the data and output dimension h, for which we explore values of 32, 128, and 512. This is then connected by a Rectified Linear Unit activation function to another layer with input dimension h and output dimension. These networks are optimized using Adam [Kingma and Ba, 2014] with a learning rate of 0.01 and weight decay values between 0.0001 and 0.1. We vary the strength of the regularization on the cut points λ_1 from values in the range of 0.1 to 20 and use a mini-batch size of 64 for the training data.

During training, we initially set the sigmoid temperature used in our smooth approximation (see Equation (2)) to a value $\tau = 0.1$, then lower it when the loss stops changing significantly between epochs. Lowering the temperature reduces the degree of smoothing and sharpens the boundaries between intervals defined by each cut point. Figure 2 shows an example training plot where the temperature drops after multiple epochs with no improvement in the validation loss. It is clear that this drop then leads to an improvement in both training and validation loss.

We perform a grid search over the hyper parameters, testing every combination of output dimension, weight decay, and regularization strength. The evaluation process to compare hyperparameters is described in Section 3.2. We train each network for 250 epochs.

To evaluate performance we perform five-fold cross validation. We randomly partition the data into training (75%), validation (15%), and test (10%) sets. For each set of hyperparameters we perform this partition five times, using the training sets for learning the model parameters. We then calculate average performance metrics metrics on the out of sample validation sets.
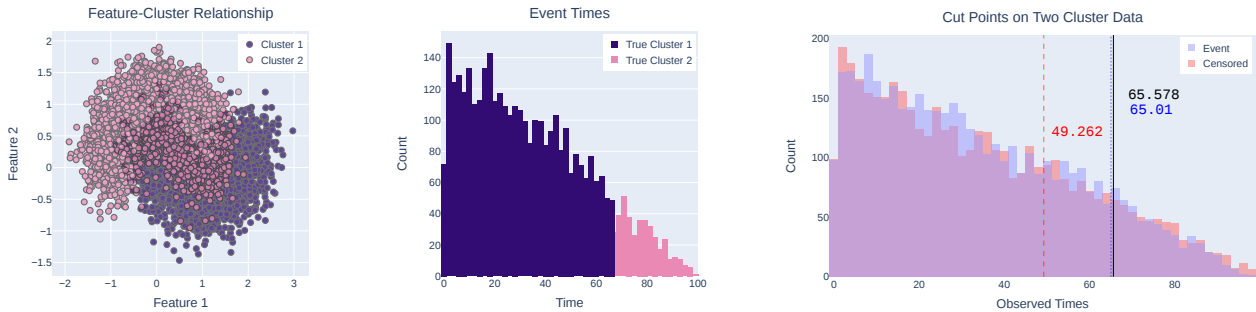
(a) The noisy, nonlinear feature-cluster relationship.

(b) The event times colored by true cluster.

(c) The cut points for ADEPT with $n = 7,500$ (black, solid), ADEPT with $n = 300$ (blue, dotted), and DTNN (red, dashed).

Figure 3: The event times and observed times of the two interval data. The true cut point is at time 67.

Table 1: Performance metrics for synthetic data. We report average metrics across 5-fold cross validation with standard errors in parentheses. In bold are the highest CI values for each setting.

Intervals	ADEPT CI	DTNN CI
Two ($n = 7,500$)	**0.947 (0.001)**	0.797 (0.002)
Two ($n = 300$)	**0.813 (0.039)**	0.756 (0.016)
Four ($n = 7,500$)	**0.980 (0.012)**	0.937 (0.007)
Four ($n = 300$)	**0.964 (0.013)**	0.931 (0.013)

Only the model with the best average validation set performance is then applied to the corresponding, yet unseen, test sets. We report the average and standard deviations of the performance metrics calculated across the folds on the test sets. Through this general cross-validation strategy, we are able to find the hyperparameter setting that performs the best on out of sample data from the hyperparameters tested. We report the mean and standard deviation of each metric across the folds.

We perform the same parameter search and evaluation to find the best DTNN model as described in Section 3.1. We compare the metrics of the best ADEPT model to that of the best baseline models. We report the performance metrics all both methods calculated on the unseen test set.

4 SIMULATION EXAMPLES

4.1 Learning Two Intervals

We start with the simple case of data generated from two clusters with uniform censoring. Cluster membership is generated using the `make_moons` function `sklearn Python` package to get a noisy, nonlinear relationship between $p = 2$ features [Pedregosa et al.,

2011]. Figure 3a shows the feature-cluster relationship; each cluster has $5,000$ observations for a total of $n = 7,500$ observations. These clusters are used to generate the event times.

Event times in Cluster 1 are generated uniformly on the interval $(0, 67]$ and event times in Cluster 2 are uniformly on the interval $(67, 100)$. Censoring times are then generated uniformly throughout the entirety of $(0, 100)$. Note that while the censoring and event times are both uniformly distributed, the observed times are the minimum of the two and therefore not uniformly distributed. These observed times are shown in Figure 3b. Because these intervals are determined by the relationship between the covariates, this simulates data generated with a true cut point at time 67.

Figure 3c shows out of sample test set along with the DTNN cut point in red at $t = 49.3$ and the ADEPT learned cut point in black at $t = 65.6$. Knowing that the true cut point is at time 67 demonstrates the efficacy of our method. Even with a starting point far from the true cut points, we are able to recover the true cut point. Table 1 reports the performance metrics, showing a large gain in CI.

In this simple example, many combinations of hyperparameters were able to recover the true cut point; reported are the results from using a small neural network with $h = 32$ with Adam weight decay of 0 and a regularization strength of $\lambda = 1$.

To demonstrate that our method works in limited data settings, we repeat this experiment randomly selecting only $n = 300$ training data points. Despite this data limitation, the blue lines in Figure 3c shows that ADEPT was still able to recover the true cut point. Table 1 shows that ADEPT still outperforms the DTNN model in prediction.
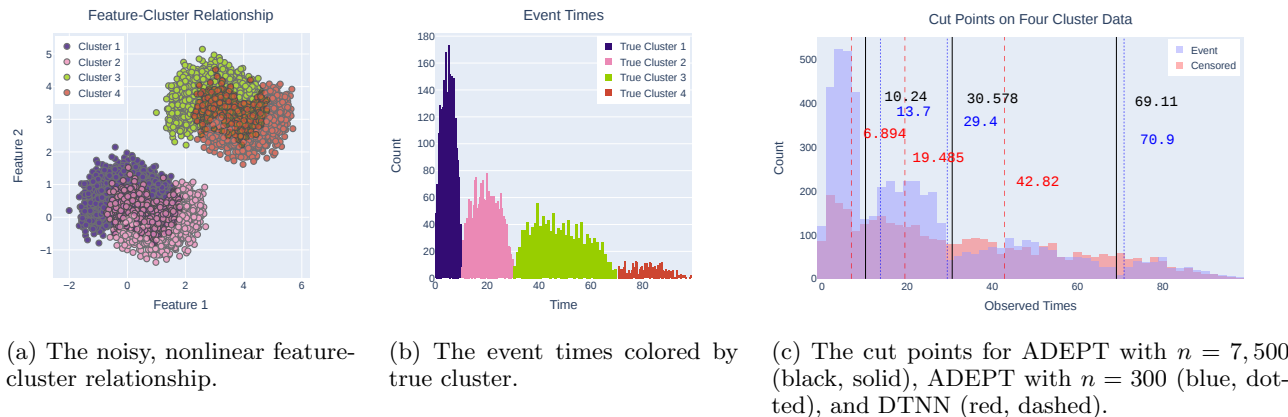
(a) The noisy, nonlinear feature-cluster relationship.

(b) The event times colored by true cluster.

(c) The cut points for ADEPT with $n = 7,500$ (black, solid), ADEPT with $n = 300$ (blue, dotted), and DTNN (red, dashed).

Figure 4: Event times and observed times of the four interval data. The true cut points are at 10, 30, and 70.

4.2 Learning Four Intervals

With confidence in ADEPT's ability to learn a single cut point when it is present in the data generation, we expand to learning three true cut points. Again we use the `make_moons` function to generate noisy, nonlinear relationships between $p = 2$ features, however now for four separate clusters as shown in Figure 4a; each cluster has 2,500 observations for a total of $n = 7,500$ observations. Figure 4b shows the how these clusters are used to generate event times. Event times are generated using a Beta$(1.5, 1.5)$ distribution which are then scaled to be in the appropriate interval based on the observation's cluster. The first cluster has event times on the interval $(0, 10]$, the second on the interval $(10, 30]$, the third on the interval $(30, 70]$, and the fourth on the interval $(70, 100)$. This corresponds to the true but points being at $t = 10, 30, 70$. We again apply uniform censoring times to all observations. Note that with uniform censoring, there are particularly few uncensored observations for events in the last interval. This makes learning the final cut point more difficult.

The black lines in Figure 4c shows that ADEPT was able to successfully recover all three cut points despite the challenges due to censoring. Table 1 shows that the ADEPT's learned intervals provide an increase in CI over the DTNN. Since we know the data generating mechanism, it is intuitive for this simulation example that including more than 3 cut points leads to worse performance as introducing more would overparametrize the model. The results in the next section suggest that it is beneficial to consider models with fewer cut points even when the generating mechanism is unknown.

We repeat this simulation with only $n = 300$ data points. With the uniform censoring, this results in

even fewer observed events in the final cluster. As in the two interval case, ADEPT is still able to recover the true cut points, shown in Figure 4c, and outperform the DTNN baseline model in predictive performance, shown in Table 1.

5 DATA ANALYSIS

5.1 Real-World Data Sources

We apply our method to three real-world data sources of varying sizes.

German Breast Cancer Study Group (GBSG) The GBSG data set is a publicly available data set introduced by Schumacher et al. [1994]. It is a multicenter clinical trial which includes $n = 686$ patients with $p = 8$ features. The endpoint of recurrence free survival occurred for 299 (43.6%) patients.

Assay of Serum Free Light Chain (FL Chain) The FL Chain data set is a publicly available data set introduced by Dispenzieri et al. [2012] studies the relationship between nonclonal serum immunoglobulin free light chains and mortality. We examine the data for the $n = 6,524$ patients that had no missing data with $p = 8$ features. The endpoint of death occurred for 1,962 (30.1%) of these patients.

Pooled Stroke Risk Cohorts This is a combined dataset consisting of the Framingham Offspring Study [Feinleib et al., 1975] ($n_1 = 8,348$), The Atherosclerosis Risk in Communities Study [Investigators, 1989] ($n_2 = 23,158$), and the Multi-Ethnic Study of Atherosclerosis ($n_3 = 6,390$) [Bild et al., 2002]. Data harmonization procedures and characteristics of the dataset have previously been described by
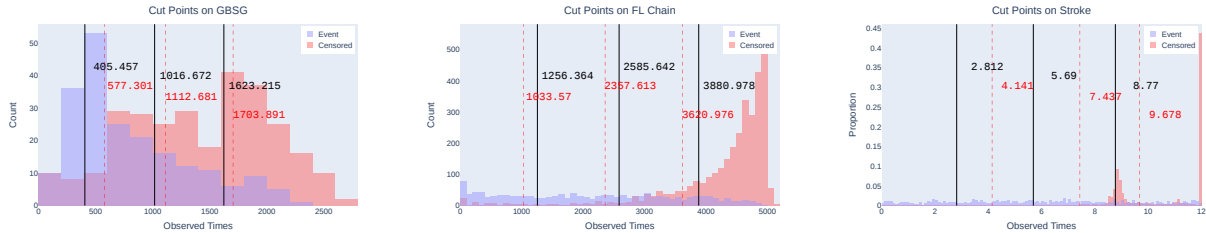
(a) Cut points on the GBSG data.   (b) Cut points on the FL Chain data.   (c) Cut points on the Stroke data.

Figure 5: The DTNN (red, dashed) and ADEPT learned (black, solid) cut points.

Hong et al. [2023]. We consider a total of $n = 35,450$ data points of which $1,221$ (3.44%) experience a stroke. There are $p = 69$ features that include cardiovascular medical history, demographic indicators, and diet information.

5.2 Results

Figure 5 shows the best ADEPT learned cut points for each data set compared to the DTNN. Note Figure 5c is a histogram of the proportion of observations rather than raw counts because of the high amount of censoring. Table 2 shows the performance metrics for all real-world data sets. The reported metrics are calculated on the held-out test sets not used for training or model validation.

Two interesting trends strongly support the benefits of ADEPT. First, for all data sets, the CI was the highest for the models that used only 3 cut points and tended to decrease as more cut points were added. Additionally, for all numbers of cut points, the predictive performance in both CI and AUC for the learned cut point model was higher than the DTNN model.

Notice that the greatest improvement in CI was observed for the GBSG data set, which has the fewest observations among all data sets. This underscores the importance of ADEPT. For small data sets with a limited number of outcomes, it is necessary to limit the number of cut points, but performance can be improved by optimizing their locations. Notice that for the FL Chain data set, the DTNN model achieved its highest CI with 10 cut points, but this was still lower than the performance of ADEPT using only 3 learned cut points. Interestingly, the Stroke data set, which had the most data points and the highest outcome imbalance, also had a higher CI and AUC with 10 cut points than with 5. Similar to the other data sets, it achieved its highest CI and AUC using 3 cut points.

A model that is well calibrated has a calibration slope near 1 and a calibration intercept near 0. While the DTNN model had slightly better calibration slopes for

the GBSG and FL Chain data sets for 3 cut points, ADEPT was better calibrated in nearly every setting with more cut points demonstrating model robustness.

While ADEPT was able to outperform DTNN in IBS for any given number of cut points, the continuous prediction of DeepSurv had the lowest overall IBS for each data set.

6 CONCLUSION

Herein we develop ADEPT, a flexible method to learn an optimal partitioning of the event time space that does not place strong assumptions on the form of the event density. Our approach is designed for clinical applications in which it is advantageous to learn, from data, a time discretization that facilitates more accurate prediction. The simulated examples demonstrated the ability of our method to recover cut points when they are truly present in the data generation mechanism. Moreover, results on real data show that the approach improves prediction performance over otherwise equivalent, state of the art models that use a fixed discretization scheme.

Our approach can be extended in several ways. In future work, we will consider a similar approach to learn separating hyperplanes in higher dimensional output spaces. The method can also be extended to sequential or time series data by using an appropriate encoder (*e.g.*, a recurrent neural network). Another interesting extension motivated by the real-world data analysis would be to learn the number of cut points from the data instead of fixing it *a priori*.

Acknowledgements

Table 2: Test-set performance metrics for real-world data. Reported are average metrics across 5-fold cross validation with corresponding standard errors in parentheses. In bold are the highest CI, highest AUC, and lowest IBS models for each data set.

	3 Cut Points	5 Cut Points	10 Cut Points
GBSG			
ADEPT CI	**0.744 (0.015)**	0.68 (0.018)	0.671 (0.024)
DTNN CI	0.681 (0.027)	0.651 (0.059)	0.619 (0.065)
ADEPT AUC	0.804 (0.021)	0.801 (0.02)	**0.822 (0.024)**
DTNN AUC	0.800 (0.03)	0.750 (0.034)	0.807 (0.016)
DeepSurv AUC	0.795 (0.054)	—	—
ADEPT IBS	0.180 (0.005)	0.189 (0.012)	0.173 (0.006)
DTNN IBS	0.187 (0.006)	0.198 (0.008)	0.194 (0.005)
DeepSurv IBS	**0.165 (0.007)**	—	—
ADEPT Calibration Slope	0.813 (0.089)	0.995 (0.112)	0.793 (0.165)
DTNN Calibration Slope	1.00 (0.091)	1.855 (0.154)	1.397 (0.254)
ADEPT Calibration Intercept	0.178 (0.048)	0.130 (0.042)	0.142 (0.06)
DTNN Calibration Intercept	0.129 (0.057)	-0.285 (0.084)	-0.215 (0.129)
FL Chain			
ADEPT CI	**0.798 (0.003)**	0.793 (0.003)	0.787 (0.004)
DTNN CI	0.763 (0.007)	0.772 (0.004)	0.774 (0.012)
ADEPT AUC	0.806 (0.004)	0.81 (0.004)	**0.834 (0.002)**
DTNN AUC	0.788 (0.008)	0.809 (0.004)	0.828 (0.004)
DeepSurv AUC	0.831 (0.002)	—	—
ADEPT IBS	0.194 (0.005)	0.163 (0.006)	0.146 (0.005)
DTNN IBS	0.191 (0.008)	0.153 (0.004)	0.13 (0.002)
DeepSurv IBS	**0.099 (0.001)**		
ADEPT Calibration Slope	1.199 (0.103)	1.182 (0.075)	1.057 (0.098)
DTNN Calibration Slope	1.005 (0.04)	0.999 (0.039)	0.898 (0.076)
ADEPT Calibration Intercept	0.056 (0.007)	0.050 (0.01)	0.085 (0.018)
DTNN Calibration Intercept	0.102 (0.008)	0.100 (0.009)	0.116 (0.026)
Stroke			
ADEPT CI	**0.789 (0.014)**	0.747 (0.02)	0.765 (0.006)
DTNN CI	0.778 (0.01)	0.739 (0.017)	0.758 (0.022)
ADEPT AUC	**0.766 (0.011)**	0.701 (0.03)	0.723 (0.013)
DTNN AUC	0.743 (0.01)	0.681 (0.031)	0.713 (0.021)
DeepSurv AUC	0.705 (0.015)		
ADEPT IBS	0.027 (0.002)	0.029 (0.002)	0.03 (0.002)
DTNN IBS	0.032 (0.003)	0.033 (0.002)	0.032 (0.002)
DeepSurv IBS	**0.019 (0.001)**		
ADEPT Calibration Slope	0.783 (0.013)	1.117 (0.195)	1.270 (0.321)
DTNN Calibration Slope	1.098 (0.321)	1.370 (0.248)	1.295 (0.376)
ADEPT Calibration Intercept	0.019 (0.002)	0.022 (0.003)	0.025 (0.002)
DTNN Calibration Intercept	0.022 (0.003)	0.019 (0.002)	0.025 (0.002)

References

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in medicine*, 24(24):3927–3944, 2005.

Diane E Bild, David A Bluemke, Gregory L Burke, Robert Detrano, Ana V Diez Roux, Aaron R Folsom, Philip Greenland, David R Jacobs Jr, Richard Kronmal, Kiang Liu, et al. Multi-ethnic study of atherosclerosis: objectives and design. *American Journal of Epidemiology*, 156(9):871–881, 2002.

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

Cynthia S Crowson, Elizabeth J Atkinson, and Terry M Therneau. Assessing calibration of prognostic risk scores. *Statistical methods in medical research*, 25(4):1692–1706, 2016.

Angela Dispenzieri, Jerry A Katzmann, Robert A Kyle, Dirk R Larson, Terry M Therneau, Colin L Colby, Raynell J Clark, Graham P Mead, Shaji Kumar, L Joseph Melton III, et al. Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. In *Mayo Clinic Proceedings*, volume 87, pages 517–523. Elsevier, 2012.

Matthew Engelhard and Ricardo Henao. Disentangling whether from when in a neural mixture cure model for failure time data. In *International Conference on Artificial Intelligence and Statistics*, pages 9571–9581. PMLR, 2022.

Manning Feinleib, William B Kannel, Robert J Garrison, Patricia M McNamara, and William P Castelli. The framingham offspring study. design and preliminary data. *Preventive medicine*, 4(4):518–525, 1975.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

Frank E Harrell Jr, Kerry L Lee, Robert M Califf, David B Pryor, and Robert A Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in medicine*, 3(2):143–152, 1984.

Chuan Hong, Michael J Pencina, Daniel M Wojdyla, Jennifer L Hall, Suzanne E Judd, Michael Cary, Matthew M Engelhard, Samuel Berchuck, Ying Xian, Ralph D'Agostino, et al. Predictive accuracy of stroke risk prediction models across black and white race, sex, and age groups. *Jama*, 329(4):306–317, 2023.

Aric Investigators. The atherosclerosis risk in communit (aric) study: design and objectives. *American Journal of Epidemiology*, 129(4):687–702, 1989.

Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger.

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 Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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 *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Marie-Louise Meng, Zachary Frere, Matthew Fuller, Yi-Ju Li, Ashraf S Habib, Jerome J Federspiel, Sarahn M Wheeler, Jennifer B Gilner, Svati H Shah, Tetsu Ohnuma, et al. Maternal cardiovascular morbidity events following preeclampsia: A retrospective cohort study. *Anesthesia & Analgesia*, pages 10–1213, 2022.

Xenia Miscouridou, Adler Perotte, Noemie Elhadad, and Rajesh Ranganath. Deep survival analysis: Non-parametrics and missingness. In *Machine Learning for Healthcare Conference*, page 244–256. PMLR, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. *arXiv preprint arXiv:1608.02158*, 2016.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4798–4805, 2019.

M Schumacher, G Bastert, H Bojar, K Hübner, M Olschewski, W Sauerbrei, C Schmoor, C Beyerle, R L Neumann, and H F Rauschecker. Randomized 2 x 2 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.

doi: 10.1200/JCO.1994.12.10.2086. URL `https://doi.org/10.1200/JCO.1994.12.10.2086`. PMID: 7931478.

Donna Tjandra, Yifei He, and Jenna Wiens. A hierarchical approach to multi-event survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 591–599, 2021.

Lee-Jen Wei. The accelerated failure time model: a useful alternative to the cox regression model in survival analysis. *Statistics in medicine*, 11(14-15):1871–1879, 1992.

Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

CHECKLIST

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable] **See Section 2.**

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/**Not Applicable**]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable] **We have provided source code with the Supplementary Material.**

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/**Not Applicable**]

   (b) Complete proofs of all theoretical results. [Yes/No/**Not Applicable**]

   (c) Clear explanations of any assumptions. [Yes/No/**Not Applicable**]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable] **The code is available in this GitHub repository**

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable] **See Section 3.3.**

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable] **See Section 3.2.**

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/**No**/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable] **See Sections 3.1 and 5.1.**

(b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not Applicable**]

(d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]