Combining Ranking and Point-wise Losses for Training Deep Survival Analysis Models

Lu Wang*,Yan Li* and Mark Chignell[§]
Dept. of Mechanical and Industrial Engineering
University of Toronto, Toronto, ON M5S 3G8, Canada
{wanglu.wang, yanrock.li}@mail.utoronto.ca, chignel@mie.utoronto.ca

2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00080

***Abstract*—Being able to accurately predict the time to event of interest, commonly known as survival analysis, is extremely beneficial in many real-world applications. Traditional commonly used statistical survival analysis methods, e.g., Cox proportional hazards model and parametric censored regressions, are based on strong and sometimes impractical assumptions and can only handle linearity relationship between features and target. Recently, deep learning based formulations have been proposed for survival analysis to handle non-linearity. However, these existing deep learning methods either inherit strong assumptions from their corresponding base models or tailor discrete-time survival analysis. To overcome the limitations within these existing models in the literature, we propose an objective function to guide the training of a deep learning model for continuous-time survival analysis. The objective function combines both ranking based and point-wise regression based losses. The ranking based loss measures the goodness of the orders of the predicted survival time for all instances. The point-wise based loss measures the difference between the predicted survival time and the true survival time for the right censored time-to-event data. More specifically, we derive two versions of the ranking based loss from the smoothed concordance index, and two versions of point-wise based loss based on the normalized mean squared error (MSE) and mean absolute error (MAE). Thus, the proposed formulation is capable of dealing with the continuous-time survival analysis from both global and local perspectives. We conduct experimental analysis over several large-scale real-world time-to-event datasets, and the results demonstrate that our model outperforms the state-of-the-art survival analysis methods. The codes and data used in the experiments are available in the link[1].**

***Index Terms*—Survival Analysis, Censored Regression, Concordance Index, Right Censoring, Deep Learning.**

I. Introduction

Survival analysis is an important sub-field of statistics, which aims at accurately predicting the time to an event of interest [1]. Because of time constraints or losing track of an instance, the event of interest may not always be observed during the observation window. Consider the example of a participant who drops out of a drug study after experiencing unpleasant side effects. This phenomenon is known as censoring, and the instance that does not have an event of interest during the observation window is known as censored instance. In general, there are three types of censoring, i.e., right censoring, left censoring, and interval censoring. In practice, right censoring is the most common scenario, where the actual survival time of a censored instance is unknown but should be greater than or equal to its last observed time (censored time). In this paper, we only consider right censoring.

Standard statistical and machine learning predictive methods fail to handle censored instances, and hence cannot be readily applied to analyze survival data (time-to-event data). To deal with this problem, various survival analysis methods have been designed, which are able to consider both uncensored instances (the instances' target events are observed) and censored instances. Consequently, survival analysis models leverage more information than standard prediction models and hence result in a more robust prediction. Many real word applications have benefited from survival analysis, such as patients death/recovery time prediction [2], fatigue life estimation of machine elements [3], and recruiting time estimation for hiring into position [4].

In traditional survival analysis, the Cox proportional hazard model along with its extensions [5]–[8] and parametric censored regression methods [9]–[12] are the most commonly used prediction methods for personalized survival time prediction. However, they both make impractical and overly strong assumptions: 1). The Cox based models assume that the hazard ratio between two instances is time-invariant and the survival curves of all instances share a similar shape with different amplitudes [5]; 2). The parametric censored regression methods require prior knowledge to choose an appropriate theoretical distribution to approximate the survival probability [9]. Recently, some deep leaning techniques have been introduced to enhance the prediction performance of the aforementioned two types of models. For instances, the Cox model is augmented with deep learning techniques [8], [13], and a deep generative model has been proposed under the framework of parametric censored regression [14]. However, these aforementioned deep survival analysis models inherit the weakness of the strong assumptions from their corresponding base models that are often unrealistic.

To overcome the weakness of above mentioned deep survival analysis models (where they are built on some unrealistic assumptions), several discrete-time survival analysis models have recently been proposed. This type of model segments the observation window into a series of time intervals, and then conducts survival time prediction via estimating whether

*These authors contributed equally to this work.
[§]Corresponding Author
[1]https://github.com/yanlirock/local_global_survival

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00080

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

the event-of-interest has happened or not at each time interval. For example, in [15] a deep neural network (DNN) is proposed that incorporates both ranking loss and binary loss to predict the probability density values at each time point for discrete-time survival analysis. In addition, different techniques such as multi-task learning algorithms [16], [17] and recurrent neural networks (RNN) [18], [19] have been used to encode the dependencies at adjacent time intervals. This type of methods is assumption free w.r.t. the underlying distribution of survival time and survival function. However, applying these methods for continuous-time survival analysis can involve an irreversible loss of precision of predicted survival time.

In light of the limitations of existing methods above, in this paper we propose an objective function to guide the training of deep learning models to achieve accurate continuous-time survival time estimation, taking into account both global and local perspectives. The proposed objective function combines a ranking based loss, i.e., smoothed concordance index (C-index), and a point-wise regression based losses, i,e., smoothed and normalized mean squared error (MSE) and mean absolute error (MAE) for right censored data.

The C-index [20], [21] is a general performance measure of rankings, which is similar to the Wilcoxon-Mann-Whitney statistic [22] used in bipartite ranking problems. Commonly, it is calculated as the proportion of concordant pairs divided by the total number of possible evaluation pairs. Therefore, it measures the goodness of the orders of the predicted survival time for all instances, and hence provides a global perspective performance quantification. However, this evaluation metric fails to consider the difference between the predicted survival time and the true survival time of each instance, and hence fails to guide the prediction model to achieve accurate survival time prediction. To deal with this problem, we have modified two standard point-wise regression losses, i.e., MSE and MAE, to enable them to measure the difference between the predicted survival time and the true survival time for each instance, in the presence of the right censoring. To the best of our knowledge, this is the first work that integrates both ranking based and point-wise regression based losses in continuous-time survival analysis. Since our proposed solution solves the problem from both global and local perspectives, it achieves more robust prediction than previous methods. Our comprehensive experiments over several large-scale real-world time-to-event datasets demonstrate that our model achieves significant improvements against state-of-the-art models under various metrics.

The rest of the paper is organized as follows. In Section 2, we list a comprehensive set of related works. The details of our proposed work are presented in section 3. Section 4 presents our experimental results on several real-world large-scale time-to-event datasets and Section 5 concludes our work with some possible further research directions.

II. Related work

A. Statistical Survival Analysis Methods

Statistical survival analysis methods are categorized into three types: i). Non-parametric methods, ii). Parametric methods, and iii). Semi-parametric methods.

The non-parametric methods, e.g., Kaplan-Meier curve [23] and Nelson-Aalen estimator [24], do not require any underlying distribution of survival time but cannot provide estimation for individual instance. Parametric censored regression methods [9], including linear censored regressions [25] and accelerated failure time (AFT) models [10], assume that the survival time or their logarithm transformation follows a particular theoretical distribution. However, choosing an appropriate distribution to approximate survival time is challenging [9]. The semi-parametric methods such as the Cox model [5] and its extensions [6], [7], [26], are built based on unrealistic and strong assumption, i.e., the proportional hazards hypothesis, where the hazard ratio between two instances is constant in time and the survival curves of all instances share a similar shape with different amplitudes. Consequently, this type of method sacrifices some flexibility and fails to model time-to-event data that has more complex structure. Another issue is that these methods cannot predict the survival time directly. In order to predict the survival time, these models have to first conduct parameter estimation to estimate the hazard ratio, and then estimate the baseline hazard function based on the Kaplan-Meier curve to approximate the shape of underling survival curves.

B. Machine Learning Methods for Survival Analysis

Over the past two decades, researchers in the machine learning and data mining communities have developed various algorithms for survival analysis, and these algorithms can be categorized into two types: i). Using machine learning techniques to improve traditional statistical methods, and ii). Modifying standard machine learning models to handle censored instances.

The first type of methods roots from the previously referenced statistical survival analysis methods refered to above, and various machine learning techniques are employed to augment model performance under different scenarios. For example some sparse learning techniques, e.g., Lasso [27] and adaptive Lasso [28], have been used to regularize Cox proportional hazard model [6], [7], [26], [29] and parametric censored regression [11], [12] in order to conduct robust survival analysis with limited numbers of instances. In recent years, some deep leaning techniques have been introduced to enable traditional statistical survival analysis methods to handle nonlinearity in large-scale time-to-event data. For example, the Cox model is augmented by deep neural networks (DNN) [30], [31] and convolutional neural networks (CNN) [13]. The parametric censored regression, with Weibull distribution, is augmented by DNN in [14].

Some standard machine learning methods are modified to handle the censored instances. For instances, random survival

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

forests [32] are built based on random forests, and standard support vector regression (SVR) are modified to handle censored instances [33], [34]. The gradient boosting algorithm has been applied to optimize pair-wise ranking based loss [35], and multi-task learning algorithms has been used to decompose censored regression as a series of dependent binary classifications [16], [17]. Apart from the techniques based on traditional machine learning models, some novel deep learning formulations have recently been proposed for discrete-time survival analysis recently. For example, in [18], [19] the RNNs have been used for discrete-time survival analysis.

Similar to our approach, a gradient boosting algorithm has been proposed to optimize the smoothed C-index [35] and in [15] the authors have proposed a feed forward DNN that incorporated both ranking loss and binary loss (log-likelihood) at each discrete time point for discrete-time survival analysis. However, in contrast to our work, in [35] the authors only consider the ranking based loss and hence the trained model can only estimate a risk score rather than survival time. Also, the model proposed in [15] is for discrete-time survival analysis. In this paper, we formulate the survival analysis as a machine learning problem from two aspects, i.e., ranking and regression. The innovative feature of our approach is that the proposed objective function combines both ranking based and point-wise regression based losses; therefore, it deals with the continuous-time survival analysis from both global and local perspectives.

III. Methodology

In this section, we first introduce the problem statement for survival analysis along with some accompanying notations. We then propose a carefully designed loss function, which combines both ranking and point-wise losses, to conduct model training for survival analysis.

A. Problem Statement

In survival analysis, an instance in the time-to-event data is usually denoted as a triplet (X_i, y_i, δ_i), where $X_i \in \mathbb{R}^{1\times p}$ is the feature vector, $y_i \in \mathbb{R}_{\geq 0}$ is the observed time, and $\delta_i \in \{0, 1\}$ is the event indicator. $\delta_i = 1$ indicates the i-th instance is an uncensored instance, i.e., the instance has an event of interest during the observation window; while $\delta_i = 0$ indicates the i-th instance is a censored instance, i.e., the instance does not have an event of interest during the observation window. For a certain instance, we can only observe either a survival time (T_i) or a censored time (C_i), but not both. And the observed time (y_i) is equal to the survival time T_i for an uncensored instance and C_i for a censored instance, i.e.,

$$y_i = \begin{cases} T_i & \text{if} \quad \delta_i = 1 \\ C_i & \text{if} \quad \delta_i = 0 \end{cases}. \tag{1}$$

For a censored instance, the exact value of T_i is unobserved; however, in the scenario of right censoring, it should be greater than or equal to the corresponding observed C_i. The goal of survival analysis is to build a prediction model, which can be used to conduct survival time estimation of a new instance with an input feature vector X_j, based on all observed triplets (X_i, y_i, δ_i) of both censored and uncensored instances.

B. The Proposed Objective

In this paper, we propose a loss function, which combines both ranking and point-wise losses, to conduct model training for survival analysis models. In our proposed framework, a prediction model such as a deep neural network is employed to conduct survival time estimation for each instance, i.e., takes X_i as input to predict the estimated survival time $\hat{y}_i$. In the training phase, the model parameters are trained via minimizing:

$$\mathcal{L}(\mathbf{y},\boldsymbol{\delta},\hat{\mathbf{y}}) = \alpha \cdot \mathcal{L}_{\text{Pointwise}}(\mathbf{y},\boldsymbol{\delta},\hat{\mathbf{y}}) + (1-\alpha)\cdot \mathcal{L}_{\text{Ranking}}(\mathbf{y},\boldsymbol{\delta},\hat{\mathbf{y}}), \tag{2}$$

where $\mathbf{y} = [y_1, \cdots, y_N]$, $\boldsymbol{\delta} = [\delta_1, \cdots, \delta_N]$, and $\hat{\mathbf{y}} = [\hat{y}_1, \cdots, \hat{y}_N]$ represent vectors of observed times, event indicators and predicted survival times, respectively. $\alpha \in [0, 1]$ is a hyper-parameter, which is used to balance the ranking based loss, $\mathcal{L}_{\text{Ranking}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}})$, and point-wise loss, $\mathcal{L}_{\text{Pointwise}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}})$. Here we propose negative a smoothed concordance index and normalized smoothed MSE/MAE as our ranking based and point-wise losses, respectively.

C. Ranking Based Loss: Smoothed Negative Concordance Index

The concordance index (C-index), is a general performance measure of prediction models that generates continuous, ordinal and dichotomous outcomes [36], which quantify the quality of predicted rankings. Let us consider a pair of 2-tuples $(y_1, \hat{y}_1)$ and $(y_2, \hat{y}_2)$, where y_i is the true target value, and $\hat{y}_i$ is the predicted outcome. The concordance probability is defined as:

$$c = Pr(\hat{y}_1 > \hat{y}_2 | y_1 > y_2). \tag{3}$$

By definition, the C-index has the same scale as the area under the ROC Curve (AUC) in binary classification, and if y_i is binary, then the C-index is same as the AUC. Therefore, similar to the AUC, $c = 1$ indicates perfect prediction and $c = 0.5$ indicates the prediction is no better than a random guess.

As censored data can be easily taken into account, the C-index is the most commonly used evaluation metric in survival analysis [20]. In 1982, Harrell et al. proposed the first definition and computational formulation of C-index for time-to-event data [20], i.e., the proportion of concordant pairs divided by the total number of possible evaluation pairs. Based on the types of learning targets, the existing survival prediction methods can be divided into two categories: risk score orientated and survival time orientated. The risk score orientated methods, e.g. the Cox proportional hazard model and its extensions, aim at learning a risk score for each instance. Note that, the instance with a low risk score should

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

survive longer. Therefore, for this type of methods the C-index is calculated as:

$$\hat{c}_{\text{Harrell-risk}} = \frac{\sum_{i,j} \mathbf{1}_{y_i<y_j} \cdot \mathbf{1}_{\hat{\eta}_i>\hat{\eta}_j} \cdot \delta_i}{\sum_{i,j} \mathbf{1}_{y_i<y_j} \cdot \delta_i} \tag{4}$$

where $\hat{\eta}_i$ and $\hat{\eta}_j$ are the predicted risk scores; $\mathbf{1}_{a<b}$ is the indicator function, which equals 1 if $a < b$ and 0 otherwise. The survival time orientated methods, e.g. parametric censored regression models, aim at directly learning the survival time of each instance, and the C-index should be calculated as:

$$\hat{c}_{\text{Harrell}} = \frac{\sum_{i,j} \mathbf{1}_{y_i<y_j} \cdot \mathbf{1}_{\hat{y}_i<\hat{y}_j} \cdot \delta_i}{\sum_{i,j} \mathbf{1}_{y_i<y_j} \cdot \delta_i}, \tag{5}$$

where $\hat{y}_i$ and $\hat{y}_j$ are the predicted survival time. In [37] Uno et al. claim that the $\hat{c}_{\text{Harrell}}$ shows an upward bias in the presence of censoring, as $\hat{c}_{\text{Harrell}}$ ignores all comparable pairs in which the smaller observed survival time is censored. To alleviate this bias, Uno et al. introduced inverse probability of censoring as a weight in the computation of the C-index. For the survival time orientated survival prediction method the C-index is calculated as:

$$\hat{c}_{\text{Uno}} = \frac{\sum_{i,j} \hat{G}(y_i)^{-2} \cdot \mathbf{1}_{y_i<y_j} \cdot \mathbf{1}_{\hat{y}_i<\hat{y}_j} \cdot \delta_i}{\sum_{i,j} \hat{G}(y_i)^{-2} \cdot \mathbf{1}_{y_i<y_j} \cdot \delta_i}, \tag{6}$$

where $\hat{G}(t)$ represents the Kaplan-Meier estimator [23] of the survival probability at time t.

In this paper, we use deep neural networks (DNN) to conduct survival time estimation for each instance; therefore, the $\hat{c}_{\text{Harrell}}$ and $\hat{c}_{\text{Uno}}$ should be calculated based on Eq.(5) and Eq.(6). However, the indicator function, Eq.(5) and Eq.(6) is not differentiable w.r.t. $\hat{y}_i$ and hence can not be used in a back propagation algorithm for training a DNN. To overcome this limitation, we employ the sigmoid function $S(\mu, \sigma) = \frac{1}{1+\exp(-\mu/\sigma)}$ to approximate the indicator function in Eq.(5) and Eq.(6), and hence the two derived versions of ranking based losses are defined as:

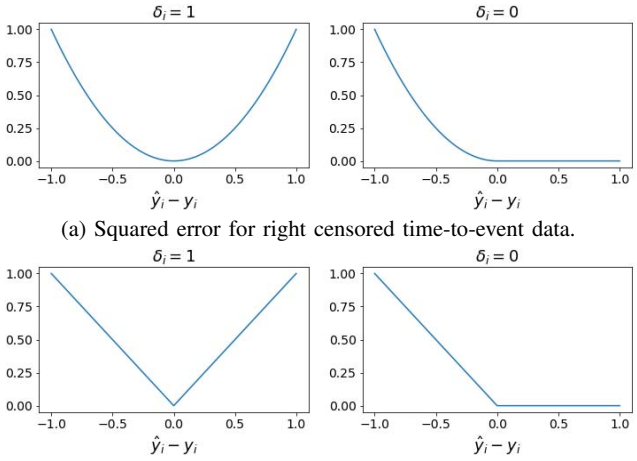
$$\begin{aligned} &\mathcal{L}_{\text{Ranking-Harrell}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}}) \\ :=& -\hat{c}_{\text{Smooth-Harrell}} \\ =& -\frac{\sum_{i,j} \mathbf{1}_{y_i<y_j} \cdot \delta_i \cdot S(\hat{y}_j - \hat{y}_i, \sigma)}{\sum_{i,j} \mathbf{1}_{y_i<y_j} \cdot \delta_i}, \end{aligned} \tag{7}$$

$$\begin{aligned} &\mathcal{L}_{\text{Ranking-Uno}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}}) \\ :=& -\hat{c}_{\text{Smooth-Uno}} \\ =& -\frac{\sum_{i,j} \hat{G}(y_i)^{-2} \cdot \mathbf{1}_{y_i<y_j} \cdot \delta_i \cdot S(\hat{y}_j - \hat{y}_i, \sigma)}{\sum_{i,j} \hat{G}(y_i)^{-2} \cdot \mathbf{1}_{y_i<y_j} \cdot \delta_i}, \end{aligned} \tag{8}$$

where σ is a tuning parameter that controls the smoothness of the approximation.

D. Point-wise Loss: Smoothed Normalized MSE and MAE

The mean squared error (MSE) and the mean absolute error (MAE) are two most commonly used metrics to evaluate the performance of regression models that generate continuous output. These two losses are point-wise since they are calculated via averaging point-wise squared/absolute differences between the estimated values and the corresponding actual values for all instances. However, due to the presence of censoring in time-to-event data, the above-mentioned two evaluation metrics and other standard evaluation metrics for regression are not suitable for survival analysis [38]. More specifically, for a censored instance its exact survival time is unknown; therefore, the difference between the estimated survival time and the actual survival time is not calculable.



(a) Squared error for right censored time-to-event data.

(b) Absolute error for right censored time-to-event data.

Fig. 1: Illustration of the squared error and absolute error for the right censored time-to-event data. The left two subfigures are uncensored instance ($\delta_i = 1$) and the other ones are right censored instance ($\delta_i = 0$).

To overcome this calculation limitation, we modify the squared error and absolute error to enable them to deal with censored instances in the scenario of right censoring. The precise survival time of a right censored instance is unknown, but it should be greater than or equal to the corresponding observed censored time. Therefore, as shown in Figure 1, once the predicted survival time of a right censored instance ($\delta_i = 0$) is greater than or equal to the corresponding censored time ($\hat{y}_i > y_i$), the modified squared error (Figure 1.a) and absolute error (Figure 1.b) are designed to neglect the difference between the predicted survival time and the censored time. For uncensored instances ($\delta_i = 1$), the difference between the predicted survival time and the actual survival time are measured via standard squared error and absolute error, respectively. Hence, the modified MSE and MAE are defined as follows:

$$\text{MSE}_{surv} = \frac{1}{\sum_i w_i} \sum_i (\hat{y}_i - y_i)^2 \cdot w_i, \tag{9}$$

$$\text{MAE}_{surv} = \frac{1}{\sum_i w_i} \sum_i |\hat{y}_i - y_i| \cdot w_i, \tag{10}$$

where

$$w_i = \begin{cases} 0 & \text{if} \quad \delta_i = 0 \ \& \ \hat{y}_i > y_i \\ 1 & \text{if} \quad \text{Otherwise} \end{cases},$$

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Details of the datasets used in this paper.

	METABRIC	GBSG	NWTCO	FLCHAIN	SUPPORT	Telco-CLT	Telco-CLV	Employee
# of instance	1904	2232	4028	6524	8873	7043	7043	14999
# of uncensored	1103	1267	571	1962	6036	1869	1869	3571
# of features	9	7	6	8	14	26	26	17
Target range	[0, 355.2]	[0.26283368, 87.359344]	[4,6209]	[0,5166]	[3, 2029]	[0, 72]	[18.8, 8684.8]	[2,10]
Target mean	125.0251	44.4916	2276.6802	3647.5021	478.6354	32.3712	2281.9170	3.4982
Target std	76.3142	27.6236	1639.9755	1458.1762	560.8000	24.5595	2265.2700	1.4601

encodes our aforementioned strategy for dealing with right censoring, which can also be formulated as:

$$w_i = \mathbf{1}_{\hat{y}_i < y_i} \cdot (1 - \delta_i) + \delta_i. \tag{11}$$

However, neither MSE_{surv} nor MAE_{surv} can be directly used as the $\mathcal{L}_{\text{Pointwise}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}})$ term in Eq.(2), because the range of values for them are $[0, +\infty)$, which makes them hard to be jointly optimized in Eq.(2). To deal with this problem, we propose the use of normalized MSE/MAE for survival analysis in the scenario of right censoring, denoted as $NMSE_{\text{surv}}$ and $NMAE_{\text{surv}}$, which are defined as follows:

$$NMSE_{\text{surv}} = \frac{\sum_i (\hat{y}_i - y_i)^2 \cdot w_i}{\sum_i (y_i - \bar{y})^2 \cdot w_i}, \tag{12}$$

$$NMAE_{\text{surv}} = \frac{\sum_i |\hat{y}_i - y_i| \cdot w_i}{\sum_i |y_i - \bar{y}| \cdot w_i}, \tag{13}$$

where $\bar{y}$ is the mean of observed time. Here, we assume a carefully designed and optimized survival prediction model performs at least as well as $\bar{y}$, and hence the range of values for $NMSE_{\text{surv}}$ and $NMAE_{\text{surv}}$ is $[0, 1]$, which is similar as the range of values for C-index. Therefore, the ranking loss $\mathcal{L}_{\text{Ranking}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}})$ and point-wise loss $\mathcal{L}_{\text{Pointwise}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}})$ in Eq.(2) can be jointly optimized easily. Note that, as $\mathbf{1}_{\hat{y}_i < y_i}$ in the definition of w_i is not differentiable w.r.t. $\hat{y}_i$, so similar to what we did for the smoothed C-index, we employ the sigmoid function $S(\mu, \sigma) = \frac{1}{1+\exp(-\mu/\sigma)}$ to approximate the indicator function in Eq. (11). The smoothed w_i is therefore defined as:

$$\tilde{w}_i = S(y_i - \hat{y}_i, \sigma) \cdot (1 - \delta_i) + \delta_i,$$

where σ controls the smoothness of the approximation, and the derived two versions of point-wise losses are defined as:

$$\mathcal{L}_{\text{Pointwise-NMSE}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}}) := NMSE_{\text{smooth-surv}} = \frac{\sum_i (\hat{y}_i - y_i)^2 \cdot \tilde{w}_i}{\sum_i (y_i - \bar{y})^2 \cdot \tilde{w}_i} \tag{14}$$

$$\mathcal{L}_{\text{Pointwise-NMAE}}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}}) := NMAE_{\text{smooth-surv}} = \frac{\sum_i |\hat{y}_i - y_i| \cdot \tilde{w}_i}{\sum_i |y_i - \bar{y}| \cdot \tilde{w}_i}. \tag{15}$$

IV. Experiment

In this section, we first describe the datasets and comparison methods used in our experiment, and then we demonstrate the prediction performance with different evaluation metrics. We also demonstrate the parameter tuning of the proposed method.

A. Experimental Datasets

For our evaluation, we use several public available large-scale real-world time-to-event datasets. We first use some datasets from the field of healthcare, which have been studied in [30] and [31], and the preprocessed datasets are made publicly available through the *pycox* python package[2] and the *DeepSurv* python package[3].

- The dataset generated by the Molecular Taxonomy of Breast Cancer International Consortium (**METABRIC**).
- The dataset from the the Rotterdam & German Breast Cancer Study Group (**GBSG**).
- The dataset provided by the the National Wilm's Tumor (**NWTCO**).
- The Assay of Serum Free Light Chain (**FLCHAIN**) dataset.
- The generated dataset from the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (**SUPPORT**).

In addition to the healthcare datasets noted above, we also examined data relating to business analysis and human resource (HR) management.

- Prediction of customer lifetime (CLT) and customer lifetime value (CLV) are vital problems for business. The *Kaggle.com* initiated a competition "Telco Customer Churn" [4] to analyze customer churn. Based on this competition we create two datasets, named as "**Telco-CLT**" and "**Telco-CLV**", to study the customer lifetime and customer lifetime value, respectively.
- Employees attrition can be very costly for companies, in [39] a human resource dataset (**Employee**) is provided to predict employee turnover. The raw dataset can be downlodad from the following link [5].

The raw datasets used in the above three tasks in business analysis and HR management contain categorical, boolean, and ordinal variables, and we conduct data preprocessing, e.g., encoding categorical features via one-hot-encoding and quantizing ordinal variables.

Table I provides the details of the datasets that were used. In this table, the row entitled "# of instance" corresponds to the number of instances in each dataset; the row entitled "# of unensored" corresponds to the number of uncensored instances

[2]https://github.com/havakv/pycox

[3]https://github.com/jaredleekatzman/DeepSurv

[4]https://www.kaggle.com/blastchar/telco-customer-churn

[5]https://github.com/square/pysurvival/blob/master/pysurvival/datasets/employee_attrition.csv

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Performance comparison of the proposed methods and other existing related methods using Harrell's C-index (along with their corresponding standard deviations).

	METABRIC	GBSG	NWTCO	FLCHAIN	SUPPORT	Telco-CLT	Telco-CLV	Employee
Cox	0.6369 (0.0242)	0.5715 (0.1169)	0.5857 (0.1422)	0.8050 (0.0712)	0.5701 (0.0146)	0.8548 (0.0111)	**0.8720 (0.0100)**	0.9034 (0.0125)
Tobit	0.6411 (0.0279)	0.6275 (0.0701)	0.7000 (0.0414)	0.8990 (0.0360)	0.5673 (0.0147)	**0.8705 (0.0109)**	0.8706 (0.0088)	0.8199 (0.0145)
Weibull	0.6350 (0.0243)	0.6242 (0.0697)	0.7025 (0.0355)	**0.9031 (0.0340)**	0.5683 (0.0147)	0.8674 (0.0120)	0.8706 (0.0092)	0.8110 (0.0173)
Logistic	0.6407 (0.0273)	0.6266 (0.0692)	0.6985 (0.0431)	0.9002 (0.0351)	0.5682 (0.0148)	0.8706 (0.0110)	0.8708 (0.0090)	0.8258 (0.0134)
Loglogistic	0.6424 (0.0301)	0.6267 (0.0697)	0.7033 (0.0355)	0.9013 (0.0341)	0.5729 (0.0118)	0.8685 (0.0114)	0.8706 (0.0090)	0.8400 (0.0132)
Lognormal	0.6414 (0.0317)	0.6266 (0.0695)	0.7043 (0.0356)	0.8990 (0.0364)	0.5726 (0.0117)	0.8694 (0.0111)	0.8714 (0.0086)	0.8393 (0.0126)
RSF	0.6506 (0.0204)	0.6253 (0.0873)	0.6898 (0.0422)	0.6403 (0.1849)	0.6136 (0.0084)	0.8501 (0.0156)	0.8527 (0.0134)	0.9291 (0.0241)
BoostCI	0.6298 (0.0204)	0.6296 (0.0783)	0.7114 (0.0360)	0.8992 (0.0473)	0.5489 (0.0125)	0.8279 (0.0145)	0.8535 (0.0099)	0.9091 (0.0229)
MTLR	0.6522 (0.0339)	0.6368 (0.0595)	0.7047 (0.0331)	0.8656 (0.0632)	0.5491 (0.0153)	0.8543 (0.0101)	0.8391 (0.0181)	0.8969 (0.0110)
DeepSurv	0.6553 (0.0205)	0.6081 (0.0529)	0.5577 (0.1020)	0.8530 (0.0481)	0.6149 (0.0073)	0.8537 (0.0115)	0.8704 (0.0090)	0.9061 (0.0240)
CoxTime	0.6684 (0.0247)	0.6422 (0.0751)	0.6930 (0.0366)	0.8875 (0.0368)	0.6216 (0.0118)	0.8537 (0.0119)	0.8684 (0.0081)	0.9113 (0.0137)
DeepHit	0.6712 (0.0315)	0.6418 (0.0643)	0.7086 (0.0409)	0.8406 (0.0999)	0.5796 (0.0150)	0.8482 (0.0129)	0.8381 (0.0240)	0.9018 (0.0177)
Harrell+NMSE	0.7158 (0.0177)	0.6480 (0.0718)	0.7195 (0.0345)	0.8854 (0.0522)	0.6195 (0.0089)	0.8659 (0.0076)	0.8715 (0.0079)	0.9466 (0.0159)
Harrell+NMAE	0.7182 (0.0181)	0.6523 (0.0702)	0.7222 (0.0389)	0.8927 (0.0536)	**0.6219 (0.0096)**	0.8660 (0.0069)	0.8697 (0.0084)	0.9465 (0.0149)
Uno+NMSE	**0.7242 (0.0194)**	0.6481 (0.0759)	**0.7242 (0.0373)**	0.8869 (0.0424)	0.6183 (0.0088)	0.8653 (0.0071)	0.8711 (0.0078)	**0.9478 (0.0172)**
Uno+NMAE	0.7177 (0.0245)	**0.6531 (0.0705)**	0.7224 (0.0351)	0.8908 (0.0470)	0.6205 (0.0093)	0.8646 (0.0086)	0.8717 (0.0080)	0.9468 (0.0148)

in each dataset; the row entitled "# of features" corresponds to the number of features in each dataset; the rows entitled "Target range", "Target mean", "Target std" correspond to the value range, mean, and standard deviations of the target value in each dataset, respectively.

B. Comparison Methods

The comparison methods used in our experiments are summarized below. We also briefly describe the basic idea of each method and provide the links to their implementation:

- **The Cox proportional hazards model (Cox)**: The Cox model is the most commonly used method in survival analysis [5], and it is trained using the *coxph* function in the *survival* R package[6] [40].
- **Parametric censored regression models**: The likelihood function of parametric censored regressions contains two parts, i.e., product of *death density probability* for all uncensored instances and product of *survival probability* for all censored instances. In the *survival* R package[6] the parametric censored regression methods can be fitted via the *survreg* function with the commonly used distributions such as normal, Weibull, logistic, log-logistic, and log-normal. Parametric censored regression with the normal distribution is known as Tobit regression [41].
- **Random Survival Forests (RSF)**: RSF is a random forests method for censored data [32]. A survival tree is grown for each bootstrap sample and the prediction error is calculated for the ensemble. The implementation of RSF can be found in the R package *randomForestSRC*[7].
- **Boosting concordance index (BoostCI)**: BoostCI is an approach where the gradient boosting algorithm is employed to optimize the smoothed concordance index [35], and the R language implementation of BoostCI can be found in the *mboost* package[8].
- **Multitask logistic regression (MTLR)**: MTLR decomposes

[6]https://cran.r-project.org/web/packages/survival/index.html

[7]https://cran.r-project.org/web/packages/randomForestSRC/index.html

[8]https://cran.r-project.org/web/packages/mboost/index.html

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

TABLE III: Performance comparison of the proposed methods and other existing related methods using MAE for uncensored instances (along with their standard deviations).

	METABRIC	GBSG	NWTCO	FLCHAIN	SUPPORT	Telco-CLT	Telco-CLV	Employee
Cox	72.62 (4.61)	16.13 (5.45)	338.57 (54.74)	1605.93 (303.45)	239.07 (6.33)	19.74 (1.00)	1527.50 (96.46)	1.7936 (0.1622)
Tobit	75.14 (4.68)	28.77 (1.81)	5667.44 (468.50)	2519.66 (1477.58)	491.07 (9.92)	19.83 (1.04)	1572.37 (92.61)	1.2463 (0.1307)
Weibull	111.40 (10.14)	46.77 (9.81)	138155.50 (24932.64)	10846.06 (15074.97)	663.36 (38.77)	167.03 (71.38)	16081.09 (6386.73)	2.1136 (0.2118)
Logistic	74.00 (4.90)	27.50 (1.87)	5372.86 (594.03)	2491.07 (1478.92)	416.57 (9.71)	19.09 (1.09)	1506.18 (95.50)	0.9841 (0.1201)
Loglogistic	74.93 (6.45)	27.02 (5.53)	73874.05 (13041.36)	8025.57 (11744.56)	267.17 (11.02)	100.23 (41.81)	9712.62 (3787.11)	0.8980 (0.1087)
Lognormal	74.16 (6.20)	27.56 (5.56)	89589.38 (15127.11)	12696.72 (21259.70)	262.98 (10.04)	93.51 (35.13)	9151.44 (3259.86)	1.0703 (0.1143)
RSF	53.70 (3.47)	16.40 (6.38)	360.76 (48.65)	1626.28 (463.04)	217.18 (8.80)	12.92 (0.73)	1792.05 (116.16)	0.3243 (0.0791)
BoostCI	100.05 (7.57)	28.77 (8.23)	390.47 (46.69)	2138.52 (318.40)	205.45 (7.69)	18.10 (0.81)	1531.83 (101.47)	1.4760 (0.0664)
MTLR	67.50 (4.20)	17.13 (2.07)	1533.07 (895.07)	1191.57 (116.96)	363.22 (9.85)	16.22 (1.06)	1352.81 (88.78)	0.4916 (0.0642)
DeepSurv	71.30 (4.17)	16.41 (5.22)	323.71 (49.82)	1575.37 (262.24)	239.93 (6.72)	19.05 (0.86)	1565.24 (100.38)	1.6162 (0.1765)
CoxTime	67.10 (5.14)	16.23 (5.18)	323.60 (49.99)	1543.36 (470.30)	229.67 (7.23)	18.55 (1.21)	1591.90 (94.96)	1.0078 (0.0964)
DeepHit	70.22 (3.38)	20.38 (2.70)	3764.95 (380.66)	1481.15 (256.20)	418.07 (22.69)	23.17 (0.94)	2148.76 (182.98)	0.4246 (0.1050)
Harrell+NMSE	49.71 (2.92)	17.90 (1.02)	**322.75 (61.15)**	**1085.87 (122.96)**	187.92 (7.19)	10.97 (0.73)	1386.80 (101.54)	0.3330 (0.0531)
Harrell+NMAE	**47.80 (3.11)**	15.55 (1.95)	389.49 (46.01)	1201.48 (273.71)	**185.65 (7.06)**	**10.40 (0.66)**	1525.44 (101.84)	**0.2817 (0.0492)**
Uno+NMSE	50.74 (2.54)	17.90 (1.07)	340.80 (60.13)	1166.66 (157.70)	189.20 (7.00)	11.12 (0.61)	**1311.13 (88.74)**	0.3477 (0.0557)
Uno+NMAE	48.76 (2.81)	**15.55 (1.78)**	388.58 (47.46)	1130.80 (125.00)	185.99 (7.17)	10.94 (0.62)	1516.66 (101.95)	0.2953 (0.0515)

the time into a series of intervals and the survival distribution is calculated as a sequence of dependent logistic regressions [16]. A Python implementation of MTLR is available in the *pycox* package[2].

- **DeepSurv**: DeepSurv is an extension of Cox proportional hazards model that employs a deep neural network to replace the linear regression in standard Cox model thereby handling the non-linearity [30]. A Pytorch implementation of DeepSurv can be found in the *pycox* package[2].
- **CoxTime**: CoxTime is a relative risk model that extends Cox regression beyond the proportional hazards [31], and its Pytorch implementation is available in the *pycox* package[2].
- **DeepHit**: DeepHit is a deep learning model designed for discrete-time survival analysis, where a feed forward DNN that incorporates both ranking loss and binary loss (log-likelihood) at each discrete time point is used to predict the probability density values at each time point [15]. A Pytorch implementation of Deephit can also be found in the *pycox* package[2].

Note that, since we aim at conducting survival analysis on large-scale datasets, we do not compare our model with sparse learning based survival analysis models such as Cox-Lasso [7] and MTLSA [17], which are designed for survival analysis on instance limited high-dimensional time-to-event data. We also do not compare our model with RNN based models [18], [19], as they are designed for discrete-time survival analysis.

C. Results and Analysis

As noted above, we introduce two ranking based losses, i.e., $-\hat{c}_{\text{Smooth-Harrell}}$ and $-\hat{c}_{\text{Smooth-Uno}}$, and two point-wise based losses, i.e., $NMSE_{\text{smooth-surv}}$ and $NMAE_{\text{smooth-surv}}$. Therefore, we have four training losses combinations: **Harrell+NMSE**, **Harrell+NMAE**, **Uno+NMSE**, and **Uno+NMAE**, respectively, and **Harrell+NMSE** stands for the losses combination that respectively employs $-\hat{c}_{\text{Smooth-Harrell}}$ and $NMSE_{\text{smooth-surv}}$ as ranking loss and point-wise loss, i.e.,

$$\mathcal{L}(\mathbf{y}, \boldsymbol{\delta}, \hat{\mathbf{y}}) = \alpha \cdot NMSE_{\text{smooth-surv}} - (1-\alpha) \cdot \hat{c}_{\text{Smooth-Harrell}},$$

and so on for the other three combinations.

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

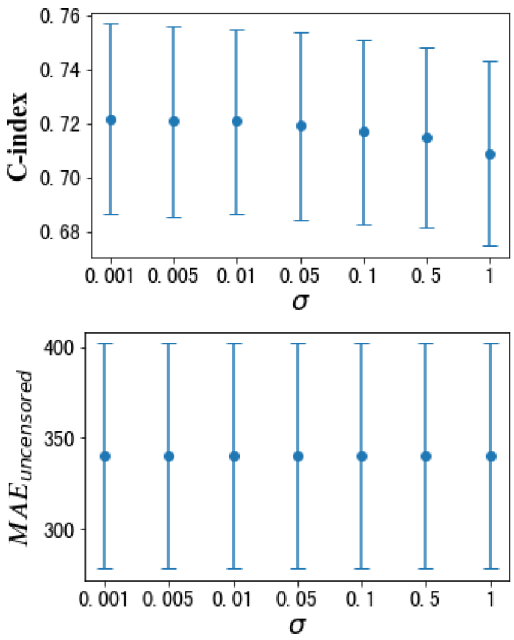


Fig. 2: The effect of σ when applying the Uno+NMSE training loss on the **NWTCO** dataset.

In our experiments, for the sake of a fair comparison, we use the same neural network architectures i.e., multi-layer perceptron (MLP) with two 32-node hidden layers, for the three state-of-the-art deep survival analysis models (DeepSurv, CoxTime and Deephit) and our proposed training losses.

In Figure 2 we present the effect of σ. We observe that the $MAE_{uncensored}$ is not sensitive to σ and the C-index of the model becomes stable when $\sigma \leq 0.01$, which is also observed in other experimental settings with various training losses combination for the other datasets. Thus, we set $\sigma = 0.01$ in the rest of our experiments to avoid "$\pm$inf" in the calculation of $\exp(-\mu/\sigma)$. We also tune the parameter α from 0 to 1 to study its effects on our proposed models. However, for the sake of fairness, in Table II and Table III we only report the results when $\alpha = 0.5$ instead of reporting the best results of our proposed models on the grid search of α. Thus, the comparison of methods is standardized.

In our experiment, we conducted 10-fold cross validation and report the mean and standard deviations of two evaluation metrics. In Table II, we provide the evaluation results of the proposed methods and other existing related methods using Harrell's C-index, which is defined in Eq.(4) and Eq.(5) for Cox based models and other survival time orientated methods, respectively. The best result for each dataset is highlighted in bold, and the results show that our proposed models either outperform or are competitive with the other state-of-the-art survival analysis models.

The C-index measures the model performance from the perspective of viewing survival analysis as a ranking problem. We would also like to evaluate the model performance of survival prediction models from the perspective of regression. In Table III, we report the MAE for uncensored instances [1], which is calculated as:

$$MAE_{\text{uncensored}} = \frac{1}{\sum_i \delta_i} \sum_i (|\hat{y}_i - y_i| \cdot \delta_i), \qquad (16)$$

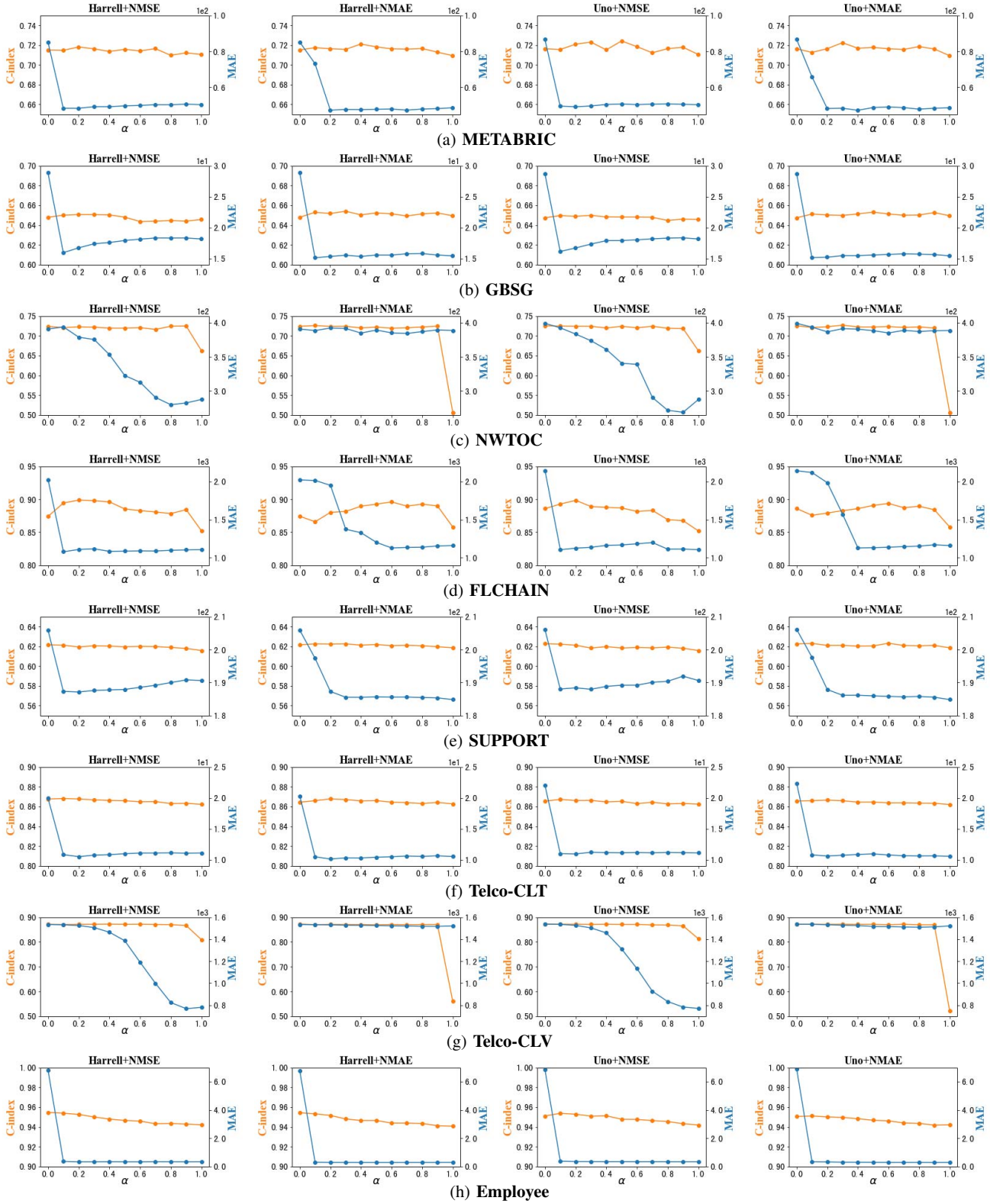
where $\hat{y}_i$ is the estimated survival time. The Cox based models (Cox, DeepSurv and CoxTime) and BoostCI aim at learning a risk score for each instance, while they fail to directly predict the survival time. For these methods we employ the Kaplan-Meier curve [23] to first estimate the *baseline survival curve* and then estimate survival time for each instance, accordingly. The results in Table III show that our proposed models outperform the other state-of-the-art survival analysis models in the task of survival time prediction.

In Figure 3, we present the C-index and MAE for uncensored instances of the proposed losses by tuning α from 0 to 1. We can see that combining both ranking and point-wise losses ($0 < \alpha < 1$) generally perform better than just using one of them. When **only considering ranking based losses (i.e.,** $\alpha = 0$**)**, the MAE for uncensored instances are relatively high; on the other hand, usually it is hard to achieve a desired C-index when the model **only considers the point-wise losses (i.e.,** $\alpha = 1$**)**. From Figure 3, we can also conclude that there is no distinct difference between Harrell's C-index and Uno's C-index, when they are used as loss functions. However, the $NMSE_{\text{smooth-surv}}$ and $NMAE_{\text{smooth-surv}}$ exhibit very different behaviours in model training, especially when the range of target value is very large with a large standard deviation. As we can see, when α is increasing, the MAE for uncensored instances in the datasets **NWTOC** and **Telco-CLV** fail to reduce in both **Harrell+NMAE** and **Uno+NMAE** methods. This is because when the range of target value is very large with a large standard deviation, the gradient of $NMAE_{\text{smooth-surv}}$ is a small constant number. Therefore, under this condition we would better choose $NMSE_{\text{smooth-surv}}$ as our point-wise loss.

V. Conclusions

In this paper, we propose an objective function, combining both ranking based loss and point-wise regression based loss, in order to guide the training of a model for continuous-time survival analysis. More specifically, in our proposed model the smoothed C-index and the modified normalized MSE/MAE are jointly optimized, where the C-index measures the goodness of the orders of the predicted survival time and the modified normalized MSE/MAE measures the difference between the predicted survival time and the true survival time. Therefore, the prediction model is trained from both global and local perspectives. In the future, we plan to develop more point-wise based losses for right censored time-to-event data based on other regression losses, e.g., Huber loss and Quantile loss, and then integrate them in our objective function for more robust prediction.

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.



(a) **METABRIC**

(b) **GBSG**

(c) **NWTOC**

(d) **FLCHAIN**

(e) **SUPPORT**

(f) **Telco-CLT**

(g) **Telco-CLV**

(h) **Employee**

Fig. 3: The effect of α on both C-index and MAE for uncensored instances in the four proposed loss combinations.

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.

ACKNOWLEDGMENTS

This work was supported by the National Science and Engineering Research Council of Canada (NSERC) Discovery Grant RGPIN-2018-06591.

REFERENCES

[1] P. Wang, Y. Li, and C. K. Reddy, "Machine learning for survival analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[2] L. Wang, Y. Li, J. Zhou, D. Zhu, and J. Ye, "Multi-task survival analysis," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 485–494.

[3] M. Modarres, M. P. Kaminskiy, and V. Krivtsov, *Reliability engineering and risk analysis: a practical guide*. CRC press, 2009.

[4] H. Li, Y. Ge, H. Zhu, H. Xiong, and H. Zhao, "Prospecting the career development of talents: A survival analysis perspective," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 917–925.

[5] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 187–220, 1972.

[6] R. Tibshirani, "The lasso method for variable selection in the cox model," *Statistics in medicine*, vol. 16, no. 4, pp. 385–395, 1997.

[7] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for cox's proportional hazards model via coordinate descent," *Journal of statistical software*, vol. 39, no. 5, pp. 1–13, 2011.

[8] J. Katzman, U. Shaham, J. Bates, A. Cloninger, T. Jiang, and Y. Kluger, "Deep survival: A deep cox proportional hazards network," *arXiv preprint arXiv:1606.00931*, 2016.

[9] E. T. Lee and J. Wang, *Statistical methods for survival data analysis*. John Wiley & Sons, 2003, vol. 476.

[10] L.-J. Wei, "The accelerated failure time model: a useful alternative to the cox regression model in survival analysis," *Statistics in medicine*, vol. 11, no. 14-15, pp. 1871–1879, 1992.

[11] Y. Li, B. Vinzamuri, and C. K. Reddy, "Regularized weighted linear regression for high-dimensional censored data," in *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 2016.

[12] Z. Wang and C. Wang, "Buckley-james boosting for survival analysis with high-dimensional biomarker data," *Statistical Applications in Genetics and Molecular Biology*, vol. 9, no. 1, 2010.

[13] X. Zhu, J. Yao, and J. Huang, "Deep convolutional neural network for survival analysis with pathological images," in *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*. IEEE, 2016, pp. 544–547.

[14] R. Ranganath, A. Perotte, N. Elhadad, and D. Blei, "Deep survival analysis," in *Machine Learning for Healthcare Conference*, 2016, pp. 101–114.

[15] C. Lee, W. R. Zame, J. Yoon, and M. van der Schaar, "Deephit: A deep learning approach to survival analysis with competing risks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] C.-N. Yu, R. Greiner, H.-C. Lin, and V. Baracos, "Learning patient-specific cancer survival distributions as a sequence of dependent regressors," in *Advances in Neural Information Processing Systems*, 2011, pp. 1845–1853.

[17] Y. Li, J. Wang, J. Ye, and C. K. Reddy, "A multi-task learning formulation for survival analysis," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. ACM, 2016, pp. 1715–1724. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939857

[18] E. Giunchiglia, A. Nemchenko, and M. van der Schaar, "Rnn-surv: A deep recurrent model for survival analysis," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 23–32.

[19] K. Ren, J. Qin, L. Zheng, Z. Yang, W. Zhang, L. Qiu, and Y. Yu, "Deep recurrent survival analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4798–4805.

[20] F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati, "Evaluating the yield of medical tests," *Jama*, vol. 247, no. 18, pp. 2543–2546, 1982.

[21] H. Steck, B. Krishnapuram, C. Dehing-oberije, P. Lambin, and V. C. Raykar, "On ranking in survival analysis: Bounds on the concordance index," in *Advances in neural information processing systems*, 2008, pp. 1209–1216.

[22] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[23] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[24] O. Aalen, "Nonparametric inference for a family of counting processes," *The Annals of Statistics*, pp. 701–726, 1978.

[25] J. Buckley and I. James, "Linear regression with censored data," *Biometrika*, vol. 66, no. 3, pp. 429–436, 1979.

[26] H. H. Zhang and W. Lu, "Adaptive lasso for cox's proportional hazards model," *Biometrika*, vol. 94, no. 3, pp. 691–703, 2007.

[27] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[28] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American statistical association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[29] B. Vinzamuri, Y. Li, and C. K. Reddy, "Active learning based survival regression for censored data," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 241–250.

[30] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC medical research methodology*, vol. 18, no. 1, p. 24, 2018.

[31] H. Kvamme, Ørnulf Borgan, and I. Scheel, "Time-to-event prediction with neural networks and cox regression," *Journal of Machine Learning Research*, vol. 20, no. 129, pp. 1–30, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-424.html

[32] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer, "Random survival forests," *The annals of applied statistics*, pp. 841–860, 2008.

[33] P. K. Shivaswamy, W. Chu, and M. Jansche, "A support vector approach to censored targets," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 655–660.

[34] F. M. Khan and V. B. Zubek, "Support vector regression for censored data (svrc): a novel tool for survival analysis," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 863–868.

[35] A. Mayr and M. Schmid, "Boosting the concordance index for survival data–a unified framework to derive and evaluate biomarker combinations," *PloS one*, vol. 9, no. 1, 2014.

[36] F. E. Harrell Jr, K. L. Lee, and D. B. Mark, "Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors," *Statistics in medicine*, vol. 15, no. 4, pp. 361–387, 1996.

[37] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. Wei, "On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," *Statistics in medicine*, vol. 30, no. 10, pp. 1105–1117, 2011.

[38] P. J. Heagerty and Y. Zheng, "Survival model predictive accuracy and roc curves," *Biometrics*, vol. 61, no. 1, pp. 92–105, 2005.

[39] S. Fotso *et al.*, "Pysurvival: Open source package for survival analysis modeling," 2019–. [Online]. Available: https://www.pysurvival.io/

[40] T. Therneau, "A package for survival analysis in s. r package version 2.37-4," *URL http://CRAN. R-project. org/package= survival. Box*, vol. 980032, pp. 23 298–0032, 2013.

[41] J. Tobin, "Estimation of relationships for limited dependent variables," *Econometrica: journal of the Econometric Society*, pp. 24–36, 1958.

Authorized licensed use limited to: Kwame Nkrumah Univ of Science and Technology. Downloaded on August 24,2024 at 06:06:28 UTC from IEEE Xplore. Restrictions apply.