DPWTE: A Deep Learning Approach to Survival Analysis Using a Parsimonious Mixture of Weibull Distributions

Achraf Bennis(✉), Sandrine Mouysset(✉), and Mathieu Serrurier(✉)

I.R.I.T - Université Toulouse III Paul Sabatier, Toulouse, France
{achraf.bennis,sandrine.mouysset,mathieu.serrurier}@irit.fr

Abstract. Survival analysis is widely used in medicine, engineering, finance, and many other areas. The fundamental problem considered in this branch of statistics is to capture the relationship between the covariates and the event time distribution. In this paper, we propose a novel network-based approach to survival analysis, called DPWTE, that uses a neural network to learn the distribution of the event times. DPWTE makes an assumption that (individual) event time distribution follows a finite mixture of Weibull distribution whose parameters are functions of the covariates. In addition, given a fixed upper bound of the mixture size, the model finds the optimal combination of Weibull distributions to model the underlying distribution. For this purpose, we introduce the *Sparse Weibull Mixture* layer, in the network, that selects through its weights, the Weibull distributions composing the mixture, whose mixing parameters are significant. To stimulate this selection, we apply a sparse regularization on this layer by adding a penalty term to the loss function that takes into account both observed and censored events, i.e. events that are not observed before the end of the period study. We conduct experiments on real-world datasets showing that the proposed model provides a performance improvement over the state-of-the-art models.

Keywords: Survival analysis · Deep learning · Weibull distribution

1 Introduction

Survival analysis, also known as time-to-event analysis, concerns the prediction of when a future event will occur. Applications of survival analysis can be found in many areas such as prediction of cardiovascular death and failure times of power grids. Survival analysis has primarily focused on interpretability at the expense of predictive accuracy. This is eventually the reason why machine-learning-based classifiers are commonly used in real-world applications while it would be more useful to apply survival methods. Certainly, some classifiers may have the best accuracy. However, these binary models can only provide predictions for a predetermined point in time. One loses the interpretability and flexibility which are guaranteed by the modeling of the event densities as a function of time. Moreover, in survival data, it is common that a part of a population

I. Farkaš et al. (Eds.): ICANN 2021, LNCS 12892, pp. 185–196, 2021.
https://doi.org/10.1007/978-3-030-86340-1_15

in which the event is not observed within the relevant time period, and could potentially occur after this recorded time or removed from the study, producing so-called *censored* data. In this case, the individuals of this sub-population provided us with censored times rather than event times. While this type of data is not taken into consideration by standard classifiers, survival analysis bridges this gap. In this work, we propose a novel approach to survival analysis: the event time distribution is assumed to follow a finite mixture of Weibull distributions, whose parameters depend on an individual's covariates. No particular assumption about the nature of the relationship between the parameters and the features is made. The main idea behind the proposed model called DPWTE, that stands for *Deep Parsimonious Weibull Time-to-Event*, is to estimate the optimal combination of Weibull distributions that models the underlying distribution using a neural network. This paper makes the following contributions:

- The event times are assumed to be drawn from a random variable following a finite mixture of Weibull distributions.
- DPWTE extends the idea behind DeepWeiSurv [3]. In fact, the latter considers the size of the combination p, as a parameter of the model whose different values are to be tested. While DPWTE, starting with an upper bound of the mixture size, learns the optimal combination of Weibull distributions (among the initial mixture) that can model the underlying distribution. For this purpose, we introduce a layer which we call the Sparse Weibull Mixture (SWM) layer on which we apply a sparse regularization. By doing this, we enforce the selection of the Weibull distributions that have a significant contribution to the time-to-event modeling.
- The censored observations are considered in the conception of the model.

This paper is organized as follows: In Sect. 2, we summarize the previous related works. In Sect. 3, we review some basic definitions in survival analysis and Weibull distributions. In Sect. 4, we describe the proposed model with a special focus on the role of the SWM layer. Section 5 is dedicated to the experiments conducted on real-world datasets.

2 Related Work

Kaplan-Meier estimator is the most widely used in survival analysis which has the advantage of being able to learn very flexible survival curves, but it doesn't incorporate individual covariates. However, the semi-parametric Cox Proportional Hazards [4] (CPH) model incorporates the covariates but assumes that the risk effect is linear with respect to the covariates, which may be too simplistic since, in the real-world data, the covariate effects are often non-monotonic. The ability of neural networks to learn nonlinear functions has encouraged many researchers to model the relationship between the covariates and the survival data. An extension of CPH with neural networks was first proposed by Faraggi and Simon [6] who replaced the linear risk of the Cox regression model, with one hidden layer multi-layer perceptron but without performance improvement.

Katzman et al. [10] revisited the Cox model in the framework of deep learning, which removes the proportionality constraint and showed that it outperforms CPH in terms of concordance index [8]. Most of the previous works benchmark their methods against the random survival forests (RSF) [9] which computes a random forest using the log-rank test as the splitting criterion, and is considered as a flexible continuous-time method that is not constrained by the proportionality assumption. Other previous works proposed network-based methods based on Cox regression such as SurvivalNet [14] and Zhu et al. [15,16] who proposed a convolutional network model that replaces multi-layer perceptron architecture of DeepSurv [10] and applied this methodology to pathological images. An alternative approach to survival analysis is to discretize the duration and compute the survival function on this predetermined time grid. Lee et al. [12] proposed a network used in competing risks setting, called DeepHit, that estimates the probability distribution and combines the log-likelihood with a ranking loss. Fotso [7] proposed N-MTLR which, using a multi-task regression, calculates the survival probabilities on the points of the time grid. Unlike discrete-time models, DeepWeiSurv [3] models a continuous survival function that allows estimating the survival probability at any survival time horizon.

3 Background

In this section, we briefly review some basics in survival analysis and Weibull distributions.

3.1 Survival Analysis

Let $X = \{x_i, y_i = (t_i, \delta_i)\}_{i=1}^{N}$ a survival data, of covariates $x_i \in \mathbb{R}^d$ and event pairs (t_i, δ_i), where $(t_i)_{1 \leq i \leq N}$ is the times recorded represented by the random variable T, and $(\delta_i)_{1 \leq i \leq N} \in \{0, 1\}^d$ is the event indicator. Typically, $\delta_i = 1$ if the event associated to the i^{th} individual is observed, otherwise, $\delta_i = 0$ which indicates censoring. The survival function is defined by the following equation:

$$S_T(t_h) = P(T > t_h) = 1 - F_T(t_h) \tag{1}$$

Survival models characterize S_T, defined as the complementary of the cumulative density function F_T, and thus the fraction of the population that survives up to a time horizon t_h given a covariate $\mathbf{x}$. Therefore the aim of these models is to estimate the probability of the occurrence of the event after or at t_h.

3.2 Mixture Weibull Distributions Estimation

We suppose that T follows a finite mixture of two-parameter Weibull distributions conditionally to the baseline data features. In this context, it is easy to calculate F_T at any time t. As this latter totally depends on the mixture parameters, we only need to estimate each couple of parameters of Weibull distributions that compose this mixture as well as its weighting coefficients. Let

T follows $\mathcal{W}_p$ a mixture of p Weibull distributions denoted by $W(\beta_i, \eta_i)$ with α_i, β_i and η_i are respectively the weighting coefficient, shape and scale of the i^{th} Weibull distribution of density $f_{W(\beta_i,\eta_i)}$ and survival function $S_{W(\beta_i,\eta_i)}$. Then the density and survival function of $\mathcal{W}_p$ can be written as follows:

$$f_{\mathcal{W}_p} = \sum_k \alpha_k f_{W(\beta_k,\eta_k)} \qquad S_{\mathcal{W}_p} = \sum_k \alpha_k S_{W(\beta_k,\eta_k)} \tag{2}$$

The log-likelihood of $\mathcal{W}_p$, considering the censored data, is defined as follows:

$$\mathcal{LL}(\beta, \eta, \alpha|y) = \overbrace{\sum_{i=1}^{n} \delta_i log\ f_{\mathcal{W}_p}(t_i)}^{\mathcal{LL}_{\delta=1}} + \overbrace{\sum_{i=1}^{n}(1-\delta_i) log\ S_{\mathcal{W}_p}(t_i)}^{\mathcal{LL}_{\delta=0}} \tag{3}$$

Thus, we estimate $\mathcal{W}_p$ parameters (α, β, η) by solving the *Maximum Likelihood Estimation* problem defined by the following equation:

$$(\hat{\beta}, \hat{\eta}, \hat{\alpha}) = \underset{\substack{\beta,\eta,\alpha \\ \beta \geq 1}}{\arg\min}\{-\mathcal{LL}(\beta, \eta, \alpha|y)\} \tag{4}$$

As we notice in Eq. (4), we set a constraint linked to the shape parameter. In fact, by definition, β and η are strictly positive. However, to assure the convexity of the $\mathcal{LL}$, we need to consider that β is at least equal to 1. Let μ_i be the mean lifetime of the i^{th} individual. Given that the mean of a mixture μ is a weighted combination of the means of the distributions that compose this mixture and knowing the single Weibull's mean [2], we have:

$$\mu_i = \sum_k \alpha_k \eta_{ik} \Gamma(1 + \frac{1}{\beta_{ik}}) \tag{5}$$

where β_{ik} and η_{ik} are the i^{th} components of β_k and η_k respectively. μ_i can be used as an estimate of the survival time of the individual i.

4 Deep Parsimonious Weibull Time-to-Event Model

In this section, we first describe the architecture of DPWTE (Sect. 4.1). Then, we explain the role of the Sparse Weibull Mixture layer (Sect. 4.2). After that, we describe the post-training steps (Sect. 4.3). Finally, we present the loss function used to train DPWTE (Sect. 4.4).

4.1 Description

As for DeepWeiSurv [3], we consider the relationship between the features and $\mathcal{W}_p$ parameters. Estimation of the mixture parameters is therefore equivalent to model this dependence. In fact, DPWTE learns the following function:

$$\begin{aligned} f_p : \mathbb{R}^d &\rightarrow \ \mathbb{R}^{p\times 3} \\ x_i &\mapsto (\alpha, \beta, \eta) \end{aligned} \tag{6}$$

The aim is therefore to train the network in order to learn the above function and thus the estimate of the triplet (α, β, η) that minimizes the log-likelihood of the distribution. DPWTE consists of a common sub-network which takes the observations X as an input and outputs a latent vector Z, this latter serves in turn as an input to both the *classifier* and *regression* sub-networks whose tasks are learning α and (β, η) respectively. Figure 1 represents the global architecture of DPWTE. For the regression sub-network, we use ELU[1] (by setting its constant to 1) as the activation function for both output layers. As the codomain of ELU in this case is $[-1, +\infty[$, to respect the optimization problem constraints as seen in Eq. (4), the network will learn $\beta + 2$ and $\eta + 1 + \varepsilon$, $\varepsilon > 0$. As for the classifier sub-network, we use the softmax activation function and interleave the SWM layer, which is described in Sect. 4.2, between the softmax and the output layer of this network. At the architecture level, the only difference between DPWTE and DeepWeiSurv is the so-called SWM layer through which the proposed model implicitly selects the significant contribution distribution.

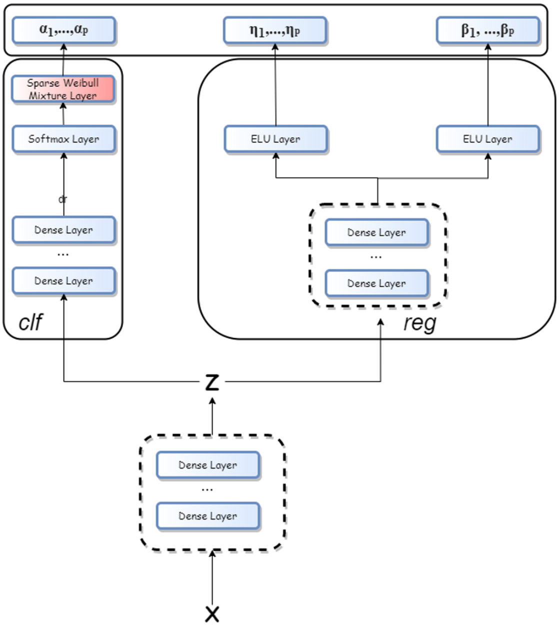

Fig. 1. The global architecture of DPWTE: clf and reg denotes the classifier and regression sub-networks respectively.

[1] We choose ELU because it becomes smooth slowly, whereas ReLU sharply smoothes. That means that with ELU we have enough gradient to learn the parameters.

4.2 Sparse Weibull Mixture Layer

It should be recalled that we seek to learn the optimal mixture of Weibull distributions that models $\mathcal{D}$, which leads us to estimate the optimal size p that we denote by $\tilde{p}$. We initially set p to an upper bound p_{max}. For this purpose, we introduce the SWM layer just before the output layer of the classifier sub-network. This layer performs an element-wise multiplication of its weights by the softmax layer output. As we see in Fig. 2, we put $\alpha_k = \omega_k \odot q_k$. In order to get an idea of the importance of each Weibull distribution, through its associated probability, we need to have the following constraints: $(\omega_k, \alpha_k) \in [0,1]^2, k = 1, .., p$ and $\sum_{k=1}^{p} \alpha_k = 1$. However, we cannot guarantee the constraint on ω_k even if we initialize them manually and thus the constraint on α_k either. To ensure implicitly these constraints, we apply the following transformations: $\forall k \in [|1, p]$,

$$(T1) \;\; \omega_k \leftarrow \frac{|\omega_k|}{\sum_{j=1}^{p} |\omega_j|} q_k \in [0,1], \forall k \in [|1,p] \qquad (T2) \;\; \alpha_k \leftarrow \frac{\alpha_k}{\sum_{k=1}^{p} \alpha_k}$$

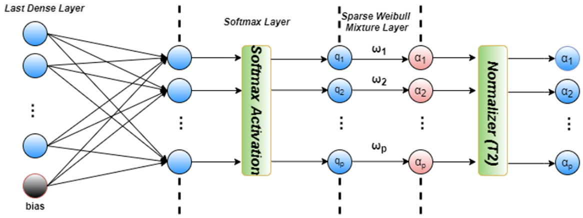

Fig. 2. Softmax and SWM layers of the classifier sub-network.

4.3 Post-Training Steps: Selection of Weibull Distributions to Combine for Time-to-Event Modeling

So far, we have not yet estimated the value of $\tilde{p}$. The training phase is the same as for DeepWeiSurv regardless of the loss function (described in Sect. 4.4). However, after the network is trained, we select the triplets $(\alpha_k, \beta_k, \eta_k)$ such as α_k is greater or equal to certain threshold denoted by ω_{th} that we fix beforehand. As the distribution of α changes after this selection while the probability constraint must be maintained, we apply T2 to the new α. Thus, if $A = \{(\alpha_k, \beta_k, \eta_k) | \alpha_k \geq \alpha_{th}\}$ is the set of selected triplets for modeling, then:

1. $\tilde{p} = Card(A)$
2. $\alpha = (\alpha_k, \alpha_k \geq \alpha_{th}) \xrightarrow[T_2]{} \alpha'$
3. $\beta = (\beta_k, \alpha_k \geq \alpha_{th}) \xrightarrow[offset(+2)]{} \beta'$
4. $\eta = (\eta_k, \alpha_k \geq \alpha_{th}) \xrightarrow[offset(+1+\epsilon)]{} \eta'$
5. the event times distribution can be modeled by $\sum_{(\alpha_k, \beta_k, \eta_k) \in A} \alpha'_k W(\beta'_k, \eta'_k)$

This post-processing is described by the Fig. 3.

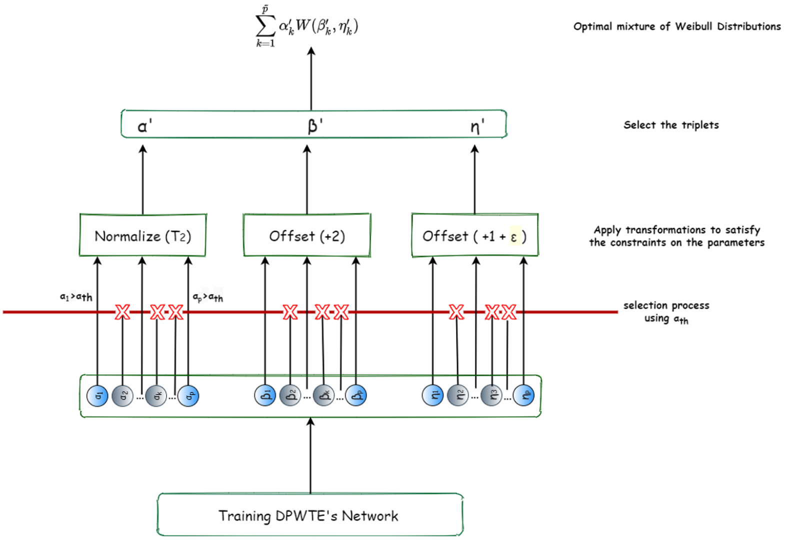

Fig. 3. Post-training steps to compose the optimal mixture of Weibull distributions.

4.4 Loss Function

As discussed above, DPWTE learns the optimal combination of Weibull distributions. To do so, we use the following loss function:

$$loss = -\mathcal{LL}(\beta, \eta, \alpha|(t_i, \delta_i)_i) + \lambda||\omega||_{\frac{1}{2}}, \tag{7}$$

where λ is the regularization parameter and $||\omega||_{\frac{1}{2}} = \sum_{k=1}^{p} \sqrt{|w_k|}$. The first element of the loss is the negative log-likelihood which is used as a loss function for DeepWeiSurv [3]. To stimulate the triplet selection process discussed in the previous section, we apply a *sparse* regularization on $\omega = (\omega_k)_{1 \geq k \geq p}$ by adding a penalty term (second operand) to the loss function, hence the name of SWM layer and the word ‘Parsimonious’ in the name of the model. The purpose behind the sparse regularization is to encourage sparsity in the vector ω or at least some ω_k to become almost zero, and then apply the threshold ω_{th}. Xu et al. [13] proposed $L_{0.5}$ as the new regularizer which is more sparse than the L_1 regularizer while it is still easier to be solved than the L_0 regularizer (because it is non-differentiable). The sparsity property of $L_{0.5}$ was demonstrated by Fan et al. [5].

5 Experiments on Real-World Datasets

In this section, we evaluate our proposed model on real data sets and compare its predictive performance with state-of-the-art methods. Table 1 gives an overview of descriptive statistics of these datasets. All the models are evaluated in the same experimental protocol.

Table 1. Descriptive statistics of real-world datasets

Datasets	No. uncensored	No. censored	No. features	Censoring time			Event time		
				Min	Max	Mean	Min	Max	Mean
SEER BC	9152(42.8%)	12221 (57.2%)	34	1	227	181.5	1	226	63.7
SEER HD	12014 (49.6%)	12221 (50.4%)					1	224	76.7
SUPPORT	5844 (68.1%)	2735 (31.9%)	36	344	2029	1060.2	3	1944	206.0
METABRIC	888 (44.8%)	1093 (55.2%)	21	1	308	116.0	1	299	77.8

5.1 Description of the Real-World Datasets

In this experiment, we use three real-worlds datasets:

- SEER[2]: a program that provides cancer incidence data from population-based cancer registries covering approximately 34.6% of the U.S. population. We focused on the patients recorded between 1998 and 2002 with Breast Cancer (BC) or Heart Disease (HD) or who have survived to the end of this period. We generated from this database two single-event datasets (BC and HD) keeping survivors in both of them.
- SUPPORT [11]: this dataset is good for learning how to fit nonlinear predictor effects. We studied 9105 patients, of which almost 32% are survivors, with their 36 attributes including age, sex, urine output creatinine, etc.
- METABRIC [3]: contains gene expressions and clinical features including age, tumor size, PR Status, etc.

5.2 Experimental Setting

For evaluation, we applied 5-fold cross validation: the data is randomly splitted into training and validation set (80-20 split). For each iteration, the models are fitted by the corresponding training set (4 folds) and evaluated on the validation set (1 fold) by calculating C^{td}. Once all iterations are executed, we obtain for each method and for each dataset, a vector (of size 5) containing C^{td} scores for each iteration. This experimental protocol is applied on the following models:

- Cox Proportional Hazards CPH [4] with a penalty term in the order of 10^{-1}.
- Random Survival Forest RSF [9] with 100 trees.
- DeepSurv [10] with 2 layers of 32 nodes.
- DeepHit [12] with a dropout probability of 0.6 between all the hidden layers.
- DeepWeiSurv [3] with $p = 10$.
- The proposed model DPWTE with $p_{max} = 10$ and $\lambda = 10^{-4}$.

All the methods are trained via Adam optimizer with a learning rate of 10^{-4}. DPWTE has the shared sub-network which is 2 fully connected layers (the batch normalization is applied before the second layer). The regression sub-network

[2] https://seer.cancer.gov.

[3] https://ega-archive.org/studies/EGAS00000000083.

consists of 1 fully connected layer with batch normalization and two ELU layers as output layers, while the classifier sub-network is composed of 2 fully connected layers and a softmax layer followed by an SWM layer. Hidden layers are activated by ReLU. The network is trained via SGD optimizer and learning rate of 10^{-4}.

As evaluation metric, we use *concordance index* C^{td} [1] which calculates, among all the comparable pairs of observations (i, j) $(\delta_i = \delta_j = 1)$, the number of concordant ones:

$$C^{td} = \frac{\sum_{i,j} \mathbb{1}_{t_i > t_j} . \mathbb{1}_{\hat{t}_i > \hat{t}_j} . \delta_j}{\sum_{i,j} \mathbb{1}_{t_i > t_j} . \delta_j}, \tag{8}$$

C^{td} estimates the probability of the event $\{\hat{t_i} > \hat{t_j} | t_i > t_j\}$ which compares the rankings of two independent and comparable pairs (non censored) of survival times (t_i, t_j) and the times predicted $(\hat{t_i}, \hat{t_j})$.

5.3 Results

The results are summarized in Table 2 where we calculated the confidence interval and the average of the concordance index scores over the 5-fold cross-validation folds. In METABRIC, DeepHit and our proposed models provided a significant improvement in terms of concordance scores when compared to other competing methods, especially DPWTE, using one $(\tilde{p} = 1)$ Weibull distribution, provides a mean concordance index slightly greater than that of DeepHit and DeepWeiSurv, but with wider interval confidence. We can say that for METABRIC, DeepHit and DPWTE have practically the same ordering performance, when we take into account the trade-off between the mean and the variance of C^{td}. For the SUPPORT dataset, DeepHit outperforms, on average, the other models in terms of times ordering, but DeepSurv and DPWTE, using in average $\tilde{p} = 3$ Weibull distributions, minimized the difference between their respective concordances and that of DeepHit compared to RSF, CPH. In the SEER dataset, for Breast Cancer and Heart Disease populations alike, we can notice that DeepWeiSurv and DPWTE (using in average $\tilde{p} = 2$ for both datasets) have shown a large significant outperformance over the competing methods, with a slight improvement from DeepWeiSurv with $p = 2$ to DPWTE. We can also remark that the standard deviation of C^{td} for METABRIC is relatively greater than that of SEER and SUPPORT. We suspect this comes from the small size of METABRIC regarding the other datasets. Furthermore, another thing to point out is that for all the datasets, except METABRIC, the respective confidence intervals of DPWTE and DeepWeiSurv are narrower than those of the competing methods, which means that our proposed method produced a more stable estimation. DPWTE has clearly the best overall predictive performance.

Table 2. C^{td} calculated over 5-fold cross validation for each model and dataset (mean ± standard deviation) as well as the mean estimate $\tilde{p}$.

Models	Datasets			
	SEER BC	SEER HD	SUPPORT	METABRIC
CPH	0.831±7.5e–3	0.785±3.5e–3	0.805±7e–3	0.661±2.6e–2
DeepSurv	0.841±5.5e–3	0.786±7.5e–3	0.826±1.5e–3	0.662±1.8e–2
RSF	0.838±9.5e–3	0.755±1e–2	0.783±4.5e–3	0.667±3.1e–2
DeepHit	0.875±8e–3	0.846±4.5e–3	**0.835**±1.3e–2	0.821±1.1e–2
DeepWeiSurv	0.908±1.5e–3	0.863±1.1e–2	0.815±1.5e–2	0.819±1.3e–2
DPWTE	**0.912±1.5e–3**	**0.871±3.5e–3**	0.831±**9.5e–3**	**0.829±1.08e–2**
$\tilde{p}$	2	2	3	1

5.4 Censoring Threshold Sensitivity Experiment

The main objective of this experiment is to measure the performance of DPWTE with respect to the censoring rate, that is, the ratio of censored events against the observed ones. Because of lack of space, we choose to run the experiment only on METABRIC (as the smallest dataset and thus more challenging) and SEER BC (as the dataset with the highest score). The main results are similar for SEER HD and SUPPORT. In this experiment, we apply the same experimental protocol as the previous one on different censoring thresholds. These thresholds, expressed in quantiles of the recorded times vector, are selected such as each quantile t_c adds a significant portion of censored data against the previous one and thus, change significantly the time distribution. Table 3 gives the distribution of data of each configuration. For METABRIC and SEER, we choose the following thresholds: $Q_1 = (q_{0.5}, q_{0.45}, q_{0.35}, q_{0.25})$ and $Q_2 = (q_{0.85}, q_{0.65}, q_{0.5}, q_{0.4}, q_{0.25})$ respectively. The *Added portion* column represents the percentage of data that became censored out of the initial set of censored data. For each value $t_c \in Q_i$, we apply 5-fold cross validation and then calculate the predictions for all time horizons $t_h \in Q_i$[4]. Then, we measure the quality of these predictions using C^{td}. Figure 4 shows the C^{td} scores calculated over the cross validation as well as the estimate $\tilde{p}$ for each scenario in both datasets. Firstly, we should note that the model performs well for SEER BC (higher average scores and narrower standard deviation as seen in the previous experiment). Furthermore, we can remark that in general, the further the censoring rate (for training) and the time horizon t_h (for predictions) is pushed back, the lower is the score. This result was expected because of the fact that the more we have non-censored data the easier it is to model the survival times distribution of the population. We also suspect the decreasing of $\tilde{p}$ comes from the fact that the more we increase the censoring rate the more the network ignores a part of the underlying distribution and

[4] $t_{METABRIC}$ is not a censoring threshold but represents the initial survival time vector as used in the previous experiment (see statistics in Table 1).

Table 3. Distribution of METABRIC (left) and SEER BC (right) for each selected censoring threshold.

t_c	No. censored	No. non-censored	Added portion
$t_{METABRIC}$	1093	888	–
$q_{0.5}$	1285	696	17.6%
$q_{0.45}$	1411	570	29%
$q_{0.35}$	1559	422	42.6%
$q_{0.25}$	1670	311	52.8%

t_c	No. censored	No. non-censored	Added portion
$q_{0.85}$	13270	8103	8.6%
$q_{0.65}$	15207	6166	24.4%
$q_{0.5}$	16568	4805	35.5%
$q_{0.4}$	17503	3870	43.2%
$q_{0.25}$	18912	2461	54.75%

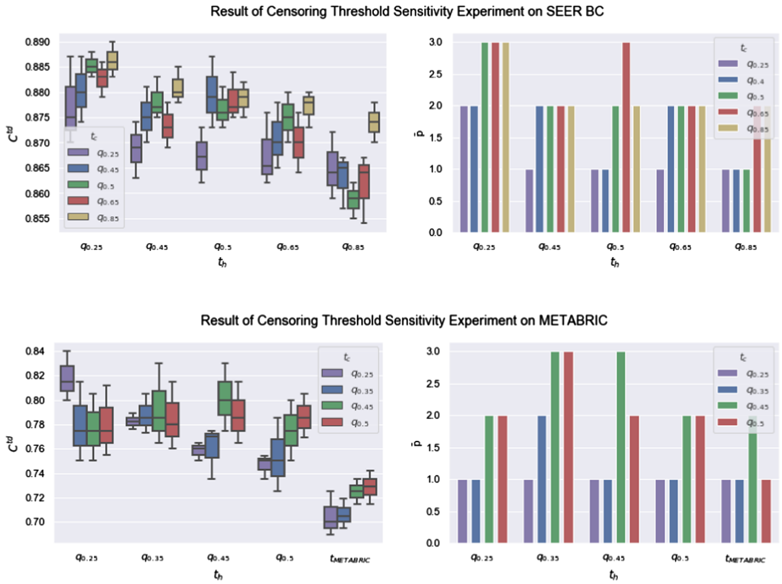

Fig. 4. Box plots (left) of C^{td} as well as the mean values of the estimate $\tilde{p}$ (right) calculated over the 5-fold cross validation for each censoring threshold t_c in both SEER BC (top) and METABRIC (bottom).

thus model the latter with an insufficient combination of Weibull distributions. However, DPWTE still performing well even in the highly censored setting.

6 Conclusion

In this paper, we proposed a novel approach for survival analysis. A network-based model, assuming a Weibull mixture character of the survival time, was presented to address this problem. We could, by parametrizing the mixture with neural networks, model rich relationships between the covariates and event times. DPWTE leverages Weibull advantages, namely the fact that these distributions are known to be a good representation for survival time distribution and it also allows to consider any time horizon. This is because DPWTE learns a continuous probability density function and through the *Sparse Weibull Mixture* layer selects the optimal combination of Weibull distribution to model the underlying event-time distribution. We conducted experiments on real-world datasets where DPWTE has clearly outperformed the alternative approaches. Furthermore, we assessed the censoring sensitivity of our model with a real-data experiment which demonstrates its ability to generally handle highly censored settings and consider any survival time horizon. Interesting expansions include extending our methodology to models that handle competing events, time-dependent covariates. In addition, it would be interesting to explore other data types and sources that require some advanced network structures notably convolutions neural networks or generative adversarial models.

References

1. Antolini, L., Boracchi, P., Biganzoli, E.: A time-dependent discrimination index for survival data. Statist. Med. **24**(24), 3927–3944 (2005)
2. Balakrishnan, N., Johnson, N.L., Kotz, S.: Continuous univariate distributions (1994)
3. Bennis, A., Mouysset, S., Serrurier, M.: Estimation of conditional mixture Weibull distribution with right censored data using neural network for time-to-event analysis. In: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (eds.) PAKDD 2020. LNCS (LNAI), vol. 12084, pp. 687–698. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47426-3_53
4. Cox, D.R.: Regression models and life tables (with discussion). J. R. Statist. Soc. Ser. B. **34**, 187–220 (1972)
5. Fan, J., Peng, H., et al.: Nonconcave penalized likelihood with a diverging number of parameters. Ann. Statist. **32**(3), 928–961 (2004)
6. Faraggi, D., Simon, R.: A neural network model for survival data. Statist. Med. **14**(1), 73–82 (1995)
7. Fotso, S.: Deep neural networks for survival analysis based on a multi-task framework. arXiv preprint arXiv:1801.05512 (2018)
8. Harrell, F.E., Califf, R.M., Pryor, D.B., Lee, K.L., Rosati, R.A.: Evaluating the yield of medical tests. Jama **247**(18), 2543–2546 (1982)
9. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S., et al.: Random survival forests. Ann. Appl. Statist. **2**(3), 841–860 (2008)
10. Katzman, J.L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., Kluger, Y.: Deep survival: a deep cox proportional hazards network. stat **1050**, 2 (2016)
11. Knaus, W.A., et al.: The support prognostic model: objective estimates of survival for seriously ill hospitalized adults. Ann. Intern. Med. **122**(3), 191–203 (1995)
12. Lee, C., Zame, W.R., Yoon, J., van der Schaar, M.: Deephit: a deep learning approach to survival analysis with competing risks. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
13. Xu, Z., Zhang, H., Wang, Y., Chang, X., Liang, Y.: L 1/2 regularization. Sci. China Inf. Sci. **53**(6), 1159–1169 (2010)
14. Yousefi, S., et al.: Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. Sci. Rep. **7**(1), 1–11 (2017)
15. Zhu, X., Yao, J., Huang, J.: Deep convolutional neural network for survival analysis with pathological images. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 544–547. IEEE (2016)
16. Zhu, X., Yao, J., Zhu, F., Huang, J.: Wsisa: making survival prediction from whole slide histopathological images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7234–7242 (2017)