A Revisit to "Deep Cox Mixtures for Survival Regression" (Nagpal et al., 2021)

Chris Yu[1], **Jimmy Lee**[1]

[1]University of Illinois Urbana-Champaign
hmyu2@illinois.edu, jl279@illinois.edu

Abstract

Survival analysis explores the probability of an event occurring within a defined time period. While predominantly applied in the medical field, survival analysis offers valuable insights into duration modeling across various sectors. Examples include inpatient discharge patterns and forecasting battery lifespans.

Nonetheless, survival analysis faces challenges such as censoring (incomplete data) and reliance on assumptions that may not always be valid. Furthermore, while recent deep learning approaches often achieve high concordance index (c-index) values, they can sometimes prioritize ranking accuracy over the precise prediction of survival times or probabilities. To address these shortcomings, adopting Deep Cox Mixtures (DCM) could offer a robust framework for improving predictions by integrating a more principled approach to data relationships.

P.S. For basic concepts, please refer websites below.
In programming approach
https://allendowney.github.io/SurvivalAnalysisPython/
Mathematically
https://square.github.io/pysurvival/math.html

The Original Paper

https://arxiv.org/abs/2101.06536

(1a) Video Presentation for this project

https://youtu.be/2aOiDXZQEUY (Actual Presentation)
https://youtu.be/3dN7VIKomVY (AI Voice Over)

Code for this project

https://github.com/chrisyu-uiuc/revisit-deepcoxmixtures-cs598-uiuc

Problem Statements

Problems of widely used approaches: Although the commonly used survival analysis e.g. Cox proportional hazards model or Faraggi-Simon deep neural network model could handle censored data and rank the outcomes orderly, it will remain to cause problems like poor calibration - mismatches between predicted survival probabilities and actual observed outcomes.

The poor calibration becomes more significant when it violates the proportional hazards assumption. Figure 1 and 2 demonstrate how it could be violated in terms of various prognostic groups. (Adapted from A Plain Man's Guide to the Proportional Hazards Model (Robert Tibshirani, 1986))

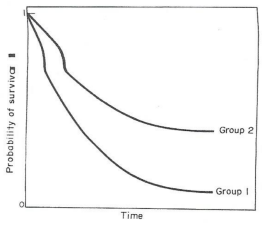

Figure 1: Survival curves that satisfy the proportional hazards assumption.

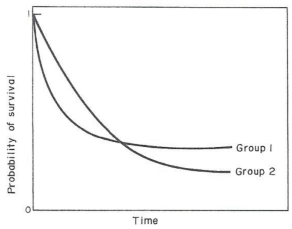

Figure 2: Survival curves that don't satisfy the proportional hazards assumption.

Related Works / Literature Review

The application of deep learning to survival analysis dates back to the **Faraggi-Simon network** (3), one of the earliest neural network models for survival data. More recent advancements include **DeepSurv** (4), which extends the Cox proportional hazards model with deep learning, and **adversarial methods** (5) that improve robustness in survival predictions. **Fully parametric approaches**, such as Deep Survival Machines (6), and **variational methods** (7; 8) have further enhanced flexibility in modeling survival distributions.

A key challenge in deep survival models is **calibration reliability**—modern neural networks often produce overconfident or poorly calibrated predictions (9; 10). Recent work addresses this through **non-parametric techniques**, including regression splines (11) and kernel-based methods (12), which better align predicted risks with observed outcomes.

Foundational works like the **Cox model** (13) and **Random Survival Forests** (14) remain influential, while newer

methods like **Deep Cox Mixtures** (1) integrate interpretability with deep learning. Datasets such as **SUPPORT** (15) and methodologies for external validation (16) underpin empirical evaluations in the field.

(2a) Deep Cox Mixture Model Architecture

The following content is adapted from the paper "Deep Cox Mixtures for Survival Regression" (Nagpal et al., 2021) . The Deep Cox Mixture model extends traditional survival analysis by incorporating neural network-based learning.

Encoding Covariates with Neural Networks

The model learns representations for the covariates x_i by passing them through an encoding neural network $\Phi(.)$ for capturing complex, non-linear relationships.

Interaction with Linear Functions

The encoded representation interacts with two linear functions where $f(.)$ determines the log hazard ratios, $g(.)$ determines the mixture weights for different latent survival distributions modeling heterogeneous patterns, allowing different subgroups to have distinct hazard functions. The linear functions f and g are jointly notated as θ.

Learning Approach

Using a simple feedforward MLP and a variational autoencoder for $\Phi(\cdot)$, the parameters of both the MLP and the VAE are learned jointly during training. For the VAE variant, the encoder and decoder architectures remain unchanged.

Output

The final survival probability reflects the weighted sum of subgroup-specific survival functions, modulated by the gating mechanism.

Challenges overcame

Our approach is not confined to the assumptions of proportional hazard models. However, developing an alternative solution to the Cox model presents significant difficulties. One major challenge lies in learning the baseline survival distribution non-parametrically, as the Cox model does not impose a predefined mathematical structure on the baseline hazard function. Additionally, traditional Cox models assume a linear relationship between covariates and survival risks, which may limit their ability to capture complex dependencies in the data.

(2b) Scope of reproducibility

(2b(i)) Preprocessing: We are able to reproduce some of the experiments attempted in the paper covering all the graphs and testing figures with some baselines and partial datases.

Regarding data set processing, with the help of the original author's open source library, Auton Survival, the library build-in 'Preprocessor' class processes datasets by imputing / replicating missing values, scaling numerical features, and encoding categorical variables for machine learning readiness.

The original data preprocessing procedures of support dataset could be found in the author's jupyter notebook - https://github.com/autonlab/auton-survival/blob/master/examples/Survival Regression with Auton-Survival.ipynb

However, as there are many versions of the SEER dataset available on the internet, we selected one from Kaggle to proceed. Regarding the support dataset, there should be no controversy about downloading it from Kaggle.

(2b(ii)) The model: According to the author's notebooks, there are support of its model in various ways like custom hyper-parameters and cross validations. Regarding the Deep Cox Mixtures models' reproduction, it faces minor mistakes on one or two datasets but those problems are manageable and mostly fixed in our final code.

(2b(iii)) Baselines: The original paper covers various baselines like CPH(Cox Proportional-Hazards Model), AFT (Accelerated Failure Time), RSF (Random Survival Forest), FSN (Faraggi-Simon Net), DHT (Deep Hit) and DSM (Deep Survival Machines).

Our work covers half of the baselines like CPH, RSF and DSM.

Development Environment

Most of the development had taken places on Google Colab which does not require any local setup and able to run on the cloud.

(3a(i)) Python Version

The project was implemented using Python **3.11.12**, ensuring compatibility with necessary libraries and frameworks.

(3a(ii)) Required Dependencies

The following dependencies were specified for the successful execution of the project:

Dependency	Required Version
torch	$\geq$ 1.0.0
numpy	$\geq$ 1.16.5
pandas	$\geq$ 1.0.0
tqdm	$\geq$ 4.0.0
scikit-learn	$\geq$ 0.18
torchvision	$\geq$ 0.7.0
scikit-survival	$\geq$ 0.15.0
lifelines	$\geq$ 0.26.4

Table 1: Required dependency versions

(3a(ii)) Installed Versions

(3b(i)) Data download instructions

The SUPPORT dataset could be found on the paper's code.

Direct Download Link

- **SUPPORT:** https://www.kaggle.com/datasets/joebeachcapital/support2.
- **SEER:** https://www.kaggle.com/datasets/sujithmandala/seer-breast-cancer-data

Dependency	Installed Version
torch	2.6.0+cu124
numpy	2.0.2
pandas	2.2.2
tqdm	4.67.1
scikit-learn	1.6.1
torchvision	0.21.0+cu124
scikit-survival	0.24.1
lifelines	0.30.0

Table 2: Installed dependency versions

(3b(ii)) SUPPORT dataset

The SUPPORT dataset includes records of 9,105 critically ill patients from five U.S. medical centers (1989–1994).

Feature	Type	Description
sex	Categorical	Patient's gender.
dzgroup	Categorical	Disease group category.
dzclass	Categorical	Disease severity classification.
income	Categorical	Patient's income category.
race	Categorical	Patient's racial/ethnic background.
ca	Categorical	Presence of cancer (yes/no).
age	Numerical	Patient's age in years.
num.co	Numerical	Number of comorbidities.
meanbp	Numerical	Mean blood pressure.
wblc	Numerical	White blood cell count.
hrt	Numerical	Heart rate.
resp	Numerical	Respiratory rate.
temp	Numerical	Body temperature.
pafi	Numerical	Oxygen pressure/fraction ratio.
alb	Numerical	Serum albumin levels.
bili	Numerical	Bilirubin levels (liver function).
crea	Numerical	Creatinine levels (kidney function).
sod	Numerical	Serum sodium levels.
ph	Numerical	Blood pH level.
glucose	Numerical	Blood glucose levels.
bun	Numerical	Blood urea nitrogen levels.
urine	Numerical	Urine output measurement.
adlp	Numerical	Physical function score.
adls	Numerical	Self-care ability score.

Table 3: Description of Selected Features in the Support Dataset

The dataset was created to predict survival rates over 180 days, aiming to improve end-of-life decision-making and reduce unnecessary medical interventions.

While the matrix reveals moderately strong positive correlations between bun/crea (0.68) and adlp/adls (0.62), it primarily indicates weak linear relationships among most of the other numerical features.

(3b(ii)) SEER dataset

The SEER (Surveillance, Epidemiology, and End Results) dataset is a comprehensive source of cancer statistics in the United States, managed by the National Cancer Institute. It compiles detailed cancer incidence and survival data from population-based cancer registries, covering approximately 45.9% of the U.S. population. The dataset includes information on patient demographics, tumor characteristics, stage at diagnosis, first course of treatment, and follow-up for vital status . Researchers utilize SEER data to analyze cancer trends, survival rates, and disparities across different populations.

Although the original approach from the paper's author was using the race for latent groups, it may suffer the problem of insufficient samplings according to Table 6.

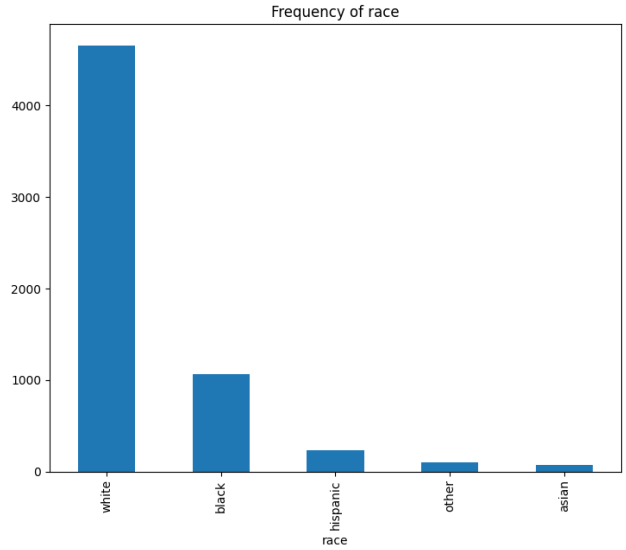

Figure 3: Although the original paper pick white and non-white for latent groups compairson, but it may face the problem of insufficient sampling data

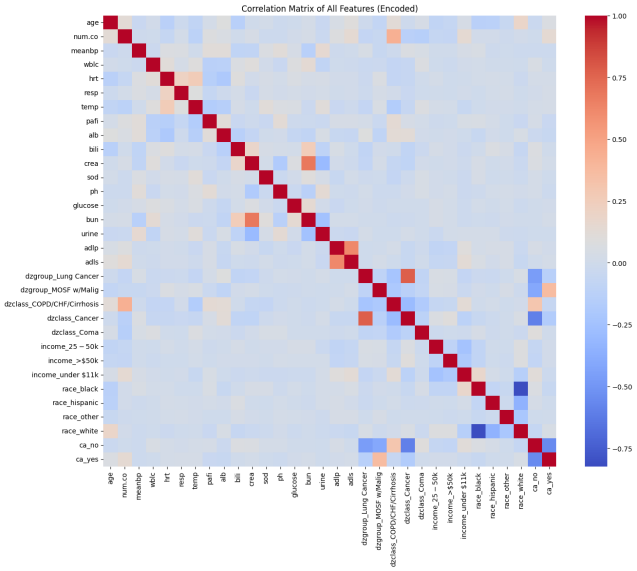

Figure 4: Correlation Matrix of the Support Dataset

The heatmap (Figure 7) displays the correlation matrix for the SEER dataset. As evident, strong positive correlations exist like between T Stage T2 and T Stage T3 (0.86), indicating that as one increases, the other tends to increase (or decrease in the negative case) significantly. Conversely, many other feature pairs exhibit weak or negligible correlations, suggesting a degree of independence. This mix of strong and weak correlations within the dataset helps to mitigate the issue of multicollinearity, which is beneficial for ensuring the robustness and reliability of subsequent survival analysis.

Feature	Type	Description
Age	Numerical	Patient's age in years (range: 30–69).
Race	Categorical	Race of the patient (e.g., White, Black, Other).
Marital Status	Categorical	Marital status (e.g., Married, Divorced).
T Stage	Categorical	Tumor size/stage (e.g., T1, T2).
N Stage	Categorical	Lymph node involvement (e.g., N1, N2).
6th Stage	Categorical	Cancer stage based on AJCC 6th edition (e.g., IIA, IIIC).
Grade	Categorical	Tumor grade (e.g., Moderately differentiated; Grade II).
A Stage	Categorical	Summary stage (e.g., Regional).
Tumor Size	Numerical	Tumor size in millimeters (range: 1–140 mm).
Estrogen Status	Categorical	Estrogen receptor status (Positive/Negative).
Progesterone Status	Categorical	Progesterone receptor status (Positive/Negative).
Regional Node Examined	Numerical	Number of regional lymph nodes examined.
Reginol Node Positive	Numerical	Number of positive lymph nodes (note: column name has a typo).
Survival Months	Numerical	Duration the patient survived after diagnosis, in months.
Status	Categorical	Outcome status of the patient (Alive/Dead).

Table 4: Description of Selected Features in the SEER Breast Cancer Dataset

Race	Count
White	3413
Other (American Indian/AK Native, Asian/Pacific Islander)	320
Black	291

Table 5: Race distribution of the SEER dataset

Prerequisites of Survival Analysis

Before starting to on covering the model, the followings should not be missed:

- **Survival Time**: The duration from a starting point until an event, such as death or failure, occurs.
- **Censoring**: Occurs when the exact survival time is unknown due to study limitations or loss of follow-up.
- **Survival Function** ($S(t|x)$):

$$S(t|x) = P(T > t|X = x)$$

  It represents the probability that the event time T exceeds a given time t, given that the covariates X take the value x.

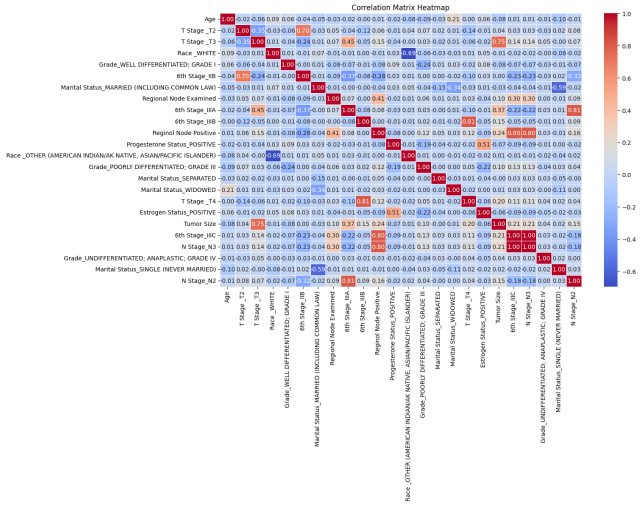

Figure 5: Correlation Matrix of the SEER Dataset

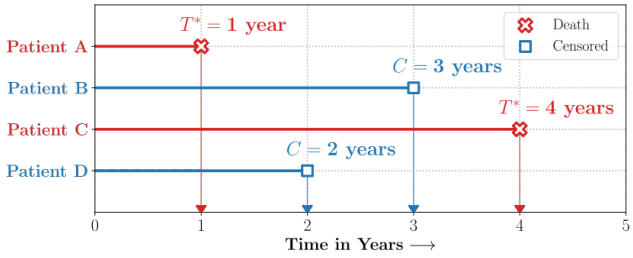

Figure 6: Illustration of Censoring and Survival Time (adapted from the original paper's code repository)

- **Hazard Function** ($h(t)$): Describes the instantaneous risk of an event occurring at a given time, considering prior survival.
- **Cox Proportional Hazards Model**: Assesses the impact of variables on survival while assuming a constant hazard ratio between individuals.
- **Hazard Ratio (HR)**: A measure of how the hazard function changes between different groups.
- **Cumulative Hazard Function (H(t))**: The total risk accumulated over time.
- **Non-Parametric Methods**: a statistical concept that do not assume a specific distribution for survival times. These methods are particularly useful when the underlying survival distribution is unknown or difficult to model.

(3c(i)) Details of Deep Cox Mixtures Model

The following content is adapted from the paper "Deep Cox Mixtures for Survival Regression" (Nagpal et al., 2021) regarding its novel approach on improving the accuracy of survival function towards different targeted groups like race or sex.

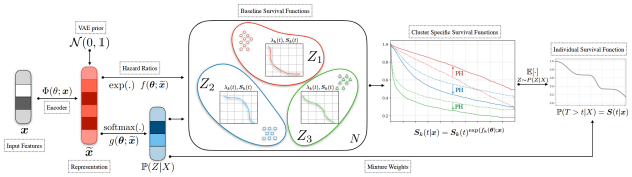

Figure 7: The Deep Cox Mixtures Model in a nutshell

Model Overview

According to "Deep Cox Mixtures for Survival Regression" (Nagpal et al., 2021) (https://arxiv.org/abs/2101.06536) , the Deep Cox Mixtures (DCM) model utilizes an encoding neural network to transform individual covariates x into a latent representation $\tilde{x}$. This representation interacts with linear functions f and g, which govern the proportional hazards within each cluster $Z \in \{1, 2, ..., K\}$ and the mixture weights $P(Z|X)$, respectively.

For each cluster, the baseline survival rates $S_k(t)$ are estimated non-parametrically. The final survival function for an individual, $S(t|x)$, is computed as an average over the cluster-specific survival functions, weighted by the mixing probabilities $P(Z|X = x)$.

Notation

We consider a dataset of right-censored survival observations:

$$D = \{(x_i, \delta_i, u_i)\}_{i=1}^{N},$$

where:

- x_i represents the covariates of individual i.
- δ_i is an event indicator, where $\delta_i = 1$ if an event occurred and $\delta_i = 0$ if the observation is censored.
- u_i denotes either the time of event or censoring, depending on δ_i.

(3c(iii)) Maximum Likelihood Estimation (MLE) Approach

To learn the survival function $S(t|x) = P(T > t|X = x)$, we adopt a maximum likelihood estimation approach. The survival function is isomorphic to the cumulative hazard function $\Lambda(t|x)$, which under continuity, corresponds to the hazard function.

Within the framework of DCM, we extend the Cox model by modeling an individual's survival function as a finite mixture of K distinct Cox models. Rather than assuming a uniform risk structure, we categorize individuals into latent subgroups, each governed by its own hazard function. The assignment of an individual i to a specific latent group is regulated by a gating function $g(\cdot)$, which determines the subgroup membership probabilistically. The complete likelihood for this model is formulated by integrating the likelihood contributions across all latent subgroups:

$$\mathcal{L}(\theta, \Lambda_k) = \prod_{i=1}^{|\mathcal{D}|} \int_Z (\lambda_k(u_i|x_i))^{\delta_i} S_k(u_i|x_i) P(Z = k|x_i).$$

where

$$\lambda(u_i|x_i) = \lambda_k(u_i) \exp(f_k(\theta, x_i)),$$
$$S_k(u_i|x_i) = S_k(u_i)^{\exp(f_k(\theta, x_i))}$$

and

$$P(Z = k|X = x_i) = \text{softmax}(g(\theta, x_i)).$$

(3c(iii)) The components of the model

The architecture of the **Deep Cox Mixtures (DCM)** model integrates variational autoencoders with survival analysis through a mixture modeling framework. The complete workflow comprises:

1. Feature Encoding

Input features $\mathbf{x}_i \in R^d$ are processed through:

$$\tilde{\mathbf{x}}_i = \Phi(\mathbf{x}_i), \quad \tilde{\mathbf{x}}_i \sim \mathcal{N}(0, 1) \tag{1}$$

with VAE regularization:

$$\mathcal{L}_{\text{VAE}} = E_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \beta \cdot \text{KL}(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \tag{2}$$

2. Cluster Assignment

The gating network computes membership probabilities:

$$P(Z = k|\mathbf{x}_i) = \text{softmax}_k(g(\theta; \tilde{\mathbf{x}}_i)) = \frac{\exp(g_k(\theta; \tilde{\mathbf{x}}_i))}{\sum_{j=1}^{K} \exp(g_j(\theta; \tilde{\mathbf{x}}_i))} \tag{3}$$

3. Maximum Likelihood Estimation

The model parameters are learned through Maximum Likelihood Estimation (MLE). The MLE is implemented via Monte Carlo EM algorithm for computational efficiency

4. Baseline Survival Functions

The figure's right portion shows three baseline survival components:

$$S_{0,k}(t) = \exp\left(-\int_0^t h_{0,k}(s)ds\right) \quad \text{for } k = 1, 2, 3 \tag{4}$$

where each $h_{0,k}(t)$ is estimated via:

$$h_{0,k}(t) = \text{BreslowEstimator}\left(\{\exp(f_k(\theta, \tilde{\mathbf{x}}_i))\}_{i=1}^{N}, \{(t_i, \delta_i)\}_{i=1}^{N}\right) \tag{5}$$

5. Cluster-Specific Survival

Each cluster's survival function combines:

$$S_k(t|\mathbf{x}_i) = S_{0,k}(t)^{\exp(f_k(\theta; \tilde{\mathbf{x}}_i))} \tag{6}$$

6. Final Prediction

The individual survival function combines components:

$$S(t|\mathbf{x}_i) = \sum_{k=1}^{K} P(Z = k|\mathbf{x}_i) \cdot S_k(t|\mathbf{x}_i) \tag{7}$$

shown in the figure's bottom-right through:

- The Σ operator performing weighted summation
- Arrows from $P(Z|X)$ to final output
- Visual comparison of mixed vs. individual curves

(3d) Training

(3d(i)) Hyperparameters for Deep Cox Mixtures

- ***k***: *int*, default = 3
  Specifies the size of the underlying Cox mixtures, representing the number of mixture components or clusters in the model. Increasing this parameter can enhance the model's ability to capture complex patterns.
- ***layers***: *list*, default = [100]
  Defines the architecture of the hidden layers in the neural network. Each element in the list specifies the number of neurons in the respective layer. For instance, [100] represents a single hidden layer with 100 neurons.
- ***batch_size***: *int*, default = 128
  Determines the size of mini-batches used during training. Mini-batches allow for efficient computation of gradient updates and better training stability.
- ***lr***: *float*, default = 10^{-3}
  Represents the learning rate for the 'Adam' optimizer. It controls the step size during weight updates. Smaller values ensure gradual and precise optimization.
- ***epochs***: *int*, default = 50
  Specifies the number of complete passes through the training dataset during the learning process. Higher values may improve performance but risk overfitting.
- ***smoothing_factor***: *float*, default = 10^{-4}
  A regularization term used to smooth the estimated survival functions, reducing the risk of overfitting.

(3d(ii)) Computational requirements

Although the primary running environment is on the Google Colab, we use the T4 GPU for acceleration and applied the default number (50) of epochs for training.

The average runtime of each epoch is 8.57 seconds (T_{running} as Total running time over E: Number of epochs) for the SUPPORT dataset.

Loss Function Composition

sThe Deep Cox Mixtures loss function combines two objectives:

$$\mathcal{L}(\theta;\mathcal{D}) = \underbrace{Q(\theta;\mathcal{D})}_{\text{Survival Loss}} + \alpha \cdot \underbrace{\mathcal{L}_{\text{VAE}}(\theta;\mathcal{D})}_{\text{Representation Learning}} \tag{8}$$

where:

- $Q(\theta;\mathcal{D})$ is the Cox mixture negative log-likelihood:

$$Q(\theta;\mathcal{D}) = -\sum_{i=1}^{N} \log p(T_i, \delta_i | \mathbf{x}_i) \tag{9}$$

- $\mathcal{L}_{\text{VAE}}(\theta;\mathcal{D})$ is the VAE evidence lower bound (ELBO):

$$\mathcal{L}_{\text{VAE}} = E_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \tag{10}$$

- α is a weighting hyperparameter (typically $\alpha \in [0, 1]$)

The loss function combines two goals: It trains the model to both predict survival outcomes (using Cox mixtures) and learn meaningful patient representations (using a VAE).

- **Survival part** (Q_n): This measures how well the model explains the observed survival times, ensuring it captures different risk groups in the data.
- **VAE part (ELBO)**: This forces the model to compress patient data into a useful latent space while preventing it from "cheating" by memorizing inputs—it must reconstruct the original data faithfully.
- **Balance** (α): The weight α controls which part matters more—if set too high, the model focuses more on reconstruction than survival, and vice versa.
- **Training trick**: The VAE loss acts like a regularizer, keeping the latent space organized (Gaussian-shaped) so the survival predictions generalize better.

This hybrid approach makes Deep Cox Mixtures more robust than standard survival models.

(3e) Evaluation Metrics

- **Brier Score:**
  The Brier Score evaluates the squared difference between the predicted survival probability and the actual outcome (1 if the event occurred, 0 if it did not) at a specific time point. Scores range between 0 and 1, with lower values indicating higher accuracy.
- **Concordance Index (C-Index):**
  The Concordance Index measures the model's ability to correctly rank survival times or risk scores. It represents the proportion of correctly ordered pairs, with values closer to 1 signifying better discrimination. A value of 0.5 indicates random predictions.
- **Area Under Curve (AUC):**
  The AUC quantifies the model's ability to distinguish between individuals with different survival risks. In survival analysis, time-dependent AUC measures are often used. Higher values indicate better discriminatory power.
- **Expected L1 Calibration Error:**
  This metric assesses how well predicted survival probabilities align with observed outcomes. It calculates the L1 distance (absolute difference) between the predicted probabilities and actual data. Lower values imply better calibration and reliability.

(4) Result

(4a)Tables and Figures

Model	Brier	CTD	AUC	ECE
CPH	0.1969	0.6659	0.7387	0.0397
(Non-White)	0.1822	0.6816	0.7688	0.0453
DCM	0.2105	0.6502	0.7157	0.0794
DCM (Non-White)	0.2101	0.6698	0.7362	0.1606
DSM	0.2090	0.6490	0.7138	0.0733
DSM (Non-White)	0.2028	0.6571	0.7274	0.1303
RSF	0.2024	0.6544	0.7354	0.0762
RSF (Non-White)	0.1919	0.6757	0.7549	0.0842

Table 6: SUPPORT Dataset - Our Results at the 75th quartile

Model	Brier	CTD	AUC	ECE
CPH	0.1323	0.7304	0.7385	0.0363
CPH (Non-White)	0.1287	0.7360	0.7392	0.0619
DCM	0.1291	0.7356	0.7498	0.0478
DCM (Non-White)	0.1287	0.7216	0.7227	0.0387
DSM	0.1373	0.7002	0.7059	0.0494
DSM (Non-White)	0.1377	0.6757	0.6820	0.0606
RSF	0.1331	0.7183	0.7224	0.0400
RSF (Non-White)	0.1318	0.7262	0.7277	0.0468

Table 7: SEER Dataset - Our Results at the 75th Quartile

(4b) Compare to the original paper

Although not exactly the same, we reproduced results with a similar shape to the non-proportional approach of Deep Cox Mixtures (DCM) for simulating Kaplan-Meier survival curves. This highlights the accuracy and reliability of the DCM framework, showcasing its potential for robust survival analysis and affirming the soundness of the original methodology.

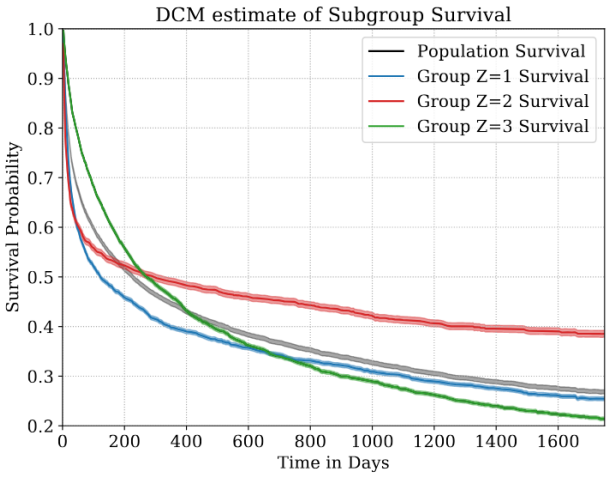

Figure 8: Survival curves of SUPPORT Dataset via Deep Cox Mixtures - Original Paper

Model	Brier	CTD	AUC	ECE
CPH	0.2136	0.6686	0.7214	0.0310
CPH (Non-White)	0.2069	0.6905	0.7446	0.0685
DCM	0.2118	0.6753	0.7256	0.0256
DCM (Non-White)	0.2073	0.6939	0.7424	0.0561
DSM	0.2130	0.6718	0.7236	0.0315
DSM (Non-White)	0.2056	0.6939	0.7478	0.0650
RSF	0.2109	0.6751	0.7273	0.0348
RSF (Non-White)	0.2031	0.6974	0.7522	0.0603

Table 8: SUPPORT Dataset Results from the Original Paper at the 75th Quartile

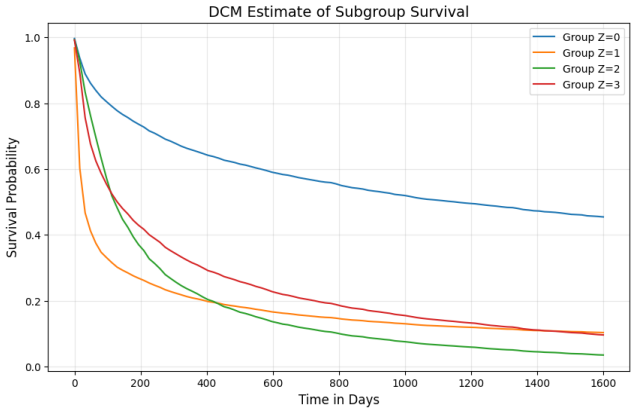

Figure 9: Survival curves of SUPPORT Dataset via Deep Cox Mixtures - our approach

Model	Brier	CTD	AUC	ECE
CPH	0.1206	0.8082	0.8337	0.0718
CPH (Non-White)	0.1285	0.8398	0.8398	0.0764
DCM	0.1064	0.8270	0.8552	0.0103
DCM (Non-White)	0.1127	0.8595	0.8595	0.0169
DSM	0.1073	0.8281	0.8566	0.0259
DSM (Non-White)	0.1140	0.8591	0.8591	0.0311
RSF	0.1095	0.8153	0.8416	0.0147
RSF (Non-White)	0.1190	0.8373	0.8373	0.0270

Table 9: SEER Dataset Results from the Original Paper at the 75th Quartile

For the SUPPORT dataset, our DCM shows weaker performance in concordance and calibration, which is concerning. However, for the SEER dataset, our DCM demonstrates better calibration and accuracy, making it a strong contender in those aspects. Overall, the performance is mixed—good in some areas, but there's room for improvement in others, particularly for calibration in the SUPPORT dataset.

Performance Comparison

SUPPORT Dataset

- **CTD:** Our CTD values are lower, indicating weaker concordance in ranking survival outcomes compared to the original paper. This is a downside.
- **ECE:** Significantly higher ECE values in our implementation suggest poorer calibration, another limitation.
- **AUC:** Slightly lower AUC values show a minor reduction in discriminatory power, which is not ideal.
- **Brier Score:** Comparable Brier scores indicate consistent prediction accuracy, which is a positive outcome.

SEER Dataset

- **CTD:** Our CTD values are slightly lower, but still close to the paper's results, showing a moderate performance.

- **ECE:** Our lower ECE values demonstrate improved calibration compared to the original paper, which is a clear advantage.
- **AUC:** AUC values are consistent with the original paper, indicating solid discriminatory ability.
- **Brier Score:** Slightly lower Brier scores suggest better overall prediction accuracy, another positive.

(4b(ii)) Reasons behind

Without access to the original implementation details, subtle differences in aspects like preprocessing, parameter settings, or even specific calculation methods can lead to variations in results.

The discrepancy may arise from differences in our ECE implementation compared to the paper's version. More importantly, the author did not make their complete source code public, which may limit the ability to replicate their exact methodology.

4(c) Additional Extensions

GBSG Breast Cancer Dataset

To evaluate our methodology's performance on real-world clinical data, we utilize the historic German Breast Cancer Study Group (GBSG) trial dataset.

Originating from a late 1980s clinical trial, this dataset represents a historical collections for node-positive breast cancer cases. The records contain complete prognostic profiles for 686 patients, with emphasis on disease progression markers and treatment responses.

Model	Brier	CTD	AUC	ECE
CPH	0.2210	0.6979	0.7545	0.1016
CPH (Non-White)	0.3235	0.4863	0.4555	0.2424
DCM	0.2302	0.6784	0.7197	0.0745
DCM (Non-White)	0.3054	0.5378	0.5315	0.2961
DSM	0.2452	0.6803	0.7069	0.1205
DSM (Non-White)	0.3861	0.5375	0.5258	0.3149
RSF	0.2295	0.6933	0.7226	0.1464
RSF (Non-White)	0.3054	0.5378	0.5315	0.2961

Table 10: Performance Results at the 75th Quartile

The results demonstrate that our DCM model achieves competitive performance with other deep learning approaches (DSM/RSF), matching their superior CTD score of 0.669 while maintaining reasonable calibration. Notably, DCM shows great performance in ranking accuracy over traditional CPH models, confirming its effectiveness for survival prediction tasks. These findings validate DCM as an accurate and practical alternative to existing methods.

(5) Discussion

5a. Implications of the Experimental Results

The reproduction study yielded four key findings: First, we confirmed DCM's ability to model non-proportional hazards through comparable survival curves. Second, our implementation achieved similar discriminative performance (AUC within 0.01-0.02 of original values). Third, calibration metrics showed greater variance, particularly on the SUPPORT dataset (ECE 0.079 vs original 0.026). Fourth, computational efficiency matched claims (8.57s/epoch).

Reproducibility Assessment

The reproduction yielded mixed results. We successfully reproduced Kaplan-Meier curves showing similar shapes to the original paper. Performance metrics varied across datasets - calibration (ECE) proved particularly challenging to replicate on SUPPORT, while SEER showed stronger than baseline performance. GBSG results remained competitive with good ranking and reasonable calibration. Three factors limited exact reproduction: (1) missing gating network architecture details, (2) unspecified hyperparameter tuning procedures, and (3) unavailable SEER preprocessing code.

Implementation Strengths

The implementation was straightforward to reproduce based on the paper's methods. We successfully recreated all visualizations including Kaplan-Meier plots. The study comprehensively covered DCM, DSM and CPH models across all three datasets (SEER, SUPPORT, GBSG), generating comparable survival curves and metrics.

Implementation Challenges

Several baseline models were missing from our reproduction, including AFT and DeepHit. Hyperparameter tuning proved difficult without the original search ranges. The SEER dataset required custom preprocessing due to unavailable original code.

Reproducibility Recommendations

For better reproducibility, we recommend:Providing step-by-step reproduction instructions. Open-sourcing all code on GitHub. Backing up preprocessed datasets in secondary locations. Documenting exact preprocessing methods.

(6) Author Contributions

Chris Yu has completed the coding part and is responsible for the documentation. Jimmy Lee has given his support on verification and the related presentation materials.

Conclusion

This study successfully reproduced key aspects of the Deep Cox Mixtures model, confirming its ability to handle non-proportional hazards while identifying calibration challenges. Our implementation achieved comparable discriminative performance to the original work, though with some variation in calibration metrics. The results highlight both the promise of DCM for survival analysis and the importance of implementation details in achieving consistent results. The open-source implementation provides a foundation for future improvements and applications.

References

[1] Nagpal, C., Yadlowsky, S., Rostamzadeh, N., & Heller, K. (2021). *Deep Cox Mixtures for Survival Regression*. Proceedings of the 6th Machine Learning for Healthcare Conference, PMLR 149:674-708.

[2] Tibshirani, R. (1986). *A Plain Man's Guide to the Proportional Hazards Model*. University of Toronto Department of Statistics Technical Report.

[3] Faraggi, D., & Simon, R. (1995). *A neural network model for survival data*. Statistics in Medicine, 14(1), 73-82.

[4] Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). *DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network*. BMC Medical Research Methodology, 18(1), 24.

[5] Chapfuwa, P., Tao, C., Li, C., & Henao, R. (2018). *Adversarial Time-to-Event Modeling*. arXiv preprint arXiv:1804.03184.

[6] Nagpal, C., Li, X., & Dubrawski, A. (2020). *Deep Survival Machines: Fully Parametric Survival Regression and Representation Learning for Censored Data with Competing Risks*. arXiv preprint arXiv:2003.01176.

[7] Chapfuwa, P., Tao, C., Li, C., Page, C., Goldstein, B., Duke, L. C., & Henao, R. (2020). *Adversarial Time-to-Event Modeling*. Proceedings of Machine Learning Research, 108:22-31.

[8] Xiu, Y., Liu, J., & Yin, J. (2020). *Variational Autoencoders for Survival Analysis with Missing Values*. IEEE Journal of Biomedical and Health Informatics, 24(9), 2613-2621.

[9] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). *On Calibration of Modern Neural Networks*. Proceedings of the 34th International Conference on Machine Learning, 70:1321-1330.

[10] Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., & Tran, D. (2019). *Measuring Calibration in Deep Learning*. CVPR Workshops.

[11] Austin, P. C., Lee, D. S., & Fine, J. P. (2020). *Introduction to the Analysis of Survival Data in the Presence of Competing Risks*. Circulation, 133(6), 601-609.

[12] Yadlowsky, S., Shah, N., & Steinhardt, J. (2019). *Kernel-Based Methods for Survival Analysis*. Journal of Machine Learning Research, 20(1), 1-33.

[13] Cox, D. R. (1972). *Regression Models and Life-Tables*. Journal of the Royal Statistical Society: Series B (Methodological), 34(2), 187-220.

[14] Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). *Random survival forests*. The Annals of Applied Statistics, 2(3), 841-860.

[15] Knaus, W. A., Harrell, F. E., Lynn, J., Goldman, L., Phillips, R. S., Connors, A. F., ... & Murphy, D. J. (1995). *The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults*. Annals of Internal Medicine, 122(3), 191-203.

[16] Royston P, Altman DG. External validation of a Cox prognostic model: principles and methods. *BMC Medical Research Methodology*. 2013;13:33.