CRESA: A Deep Learning Approach to Competing Risks, Recurrent Event Survival Analysis

Garima Gupta, Vishal Sunder, Ranjitha Prasad, and Gautam Shroff

TCS Research Lab, New Delhi
{gupta.garima1,s.vishal3,ranjitha.prasad,gautam.shroff}@tcs.com

Abstract. Survival analysis refers to a gamut of statistical techniques developed to infer the *survival time* from time-to-event data. In particular, we are interested in *recurrent* event survival analysis in the presence of one or more *competing risks* in each recurrent time-step, in order to obtain the probabilistic relationship between the input covariates and the distribution of event times. Since traditional survival analysis techniques suffer from drawbacks due to strong parametric model constraints and constant hazard based assumptions, we propose a modern deep learning based flexible probabilistic framework for cause-specific recurrent survival analysis. In single-risk scenarios, we propose an LSTM-based model where the time-steps represent the recurrent events for each participant whose covariates may be static or time-varying. To cater to multi-risk scenarios, we build on the single-risk LSTM model and introduce a cumulative incidence curve approach to handle the multiple competing risks using a joint distribution over the event times and each of the competing risks over multiple time-steps and term the proposed novel architecture as *CRESA*. We use the concordance index per risk and the maximum absolute error in every time-step as the metrics of performance. We demonstrate a superior predictive performance of the proposed approach (single and multiple risk scenarios) as compared to traditional model-based approaches, and deep learning based approaches for synthetic and state-of-the-art public datasets.

Keywords: Recurrent neural networks, Competing risks, Hazard function, Deep learning, LSTM, Cox models, Frailty

1 Introduction

In the broad field of study of temporal data, survival analysis is a well-known statistical technique for the study of temporal events. Classic applications of survival analysis has been in the field of reliability engineering especially for equipments under stress, where accurately measuring the uncertainty associated with events related to the critical parameters of an individual or equipment is paramount. With the advent of data collection technologies, survival analysis has been widely used in the field of medical statistics for patient monitoring, treatment plans etc. Typically in survival analysis, time-to-an-event data is modeled using a parametric probabilistic function of some observed, or partially observed covariates. The fundamental issue with most time-to-event data is the presence of *censored* observations, i.e.,the observations pertaining to those participants whose event of interest is not observed as it was the end of observation period or due to participants dropping out in the observation period. Note that neglecting censored data can introduce a bias in the inference process, and hence, analyzing such time-to-event data necessitates significantly different statistical and machine learning techniques. One of the popular naive non-parametric technique for obtaining the empirical estimate of the survival function is by using the Kaplan-Meier (KM) method [11]. The drawback of this technique is that it does not incorporate the available covariates in the data. The most popular semi-parametric technique for survival analysis is the Cox proportional hazards (CPH) model [3], which incorporates the

available covariates using an exponential probabilistic framework along with strong assumptions regarding the proportionality of hazards in the underlying stochastic process. Several other semi-parametric models explore specific forms of the underlying stochastic process such as a Weiner or a Markov process [15,4,17].

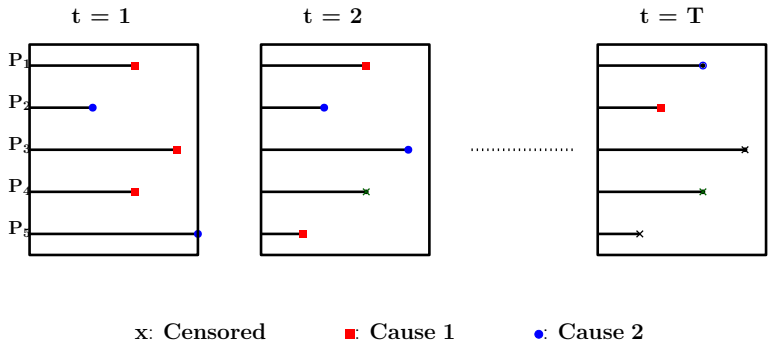

Fig. 1. Typical scenarios encountered in cause-specific recurrent survival analysis depicted on 5 participants, $P_1, \ldots, P_5$. P_1 and P_2 report events in all the time-steps, however the cause of event is not necessarily same in all time-steps. P_3, P_4 and P_5 are censored after a few time-steps.

Recent advances in deep learning have transformed clinical practices especially in the field of machine-learning based diagnostic applications, health-care applications, etc. Although the first paper incorporating neural networks for survival data modeling was two decades ago [5], a renewed interest in survival analysis has emerged since the advent of such modern approaches. Typically, a deep learning architecture is employed to learn the parameters of a general CPH model. For instance, [27] replaces the exponential component of a CPH model with a deep convolutional network. In [12] and [19], the authors train the neural network based on a loss function pertaining to a CPH model to learn the first hitting-time of the event of interest. Further, in [16], the authors propose an LSTM-based feature extractor in conjunction with a conventional CPH model to predict assets' health. In spite of incorporating powerful deep learning based techniques, such approaches necessitate the assumption of a constant hazard rate.

Often, survival analysis data comprises of *competing risks*, where there are at least two possible causes for an event, but only one such event type can actually occur at the time of the event. A typical *cause-specific* approach for analyzing such data is to infer for each event type separately, considering the data reporting alternate events as censored observations. However such approaches may be biased and inaccurate since these competing risks may not be independent. Furthermore, fitting the KM-based survival curve obtained for each event type is also inaccurate [13]. An alternative to the cause censoring approach is the *Cumulative Incidence Curve (CIC)* approach, which estimates the marginal probability of an event. In [7] and [6], the authors model the hazard function by using the cumulative incidence function (CIF). In [20], the authors proposed to fit a single CPH model rather than separate models for each event-type, thus eliminating the need for censoring different event causes. However, all the above models make strong assumptions about the structure of the underlying stochastic processes and the hazard function. This has led to the advent of non-model based approaches to handle the competing risks scenario. In [1] and [22], the authors propose a non-CPH type, deep multi-task Gaussian process and deep exponential family based models to capture the interactions between the data covariates and cause-specific survival times. More recently, the DeepHit [14] approach was proposed where a deep learning model is incorporated to learn the joint distribution of cause-specific survival times and events of interest directly, without relying on any constant-hazard rate assumption.

Typically, survival data consists of instances where the event experienced by the participant is not necessarily *death* or disappearance from the study, and participants experience events multiple times in the same observation time period. For example, in the reliability domain, an engine may report a failure due to valve-related issues, and another failure due to piston-related issues in an observation window, as depicted in Fig. 1. Such a scenario where, for a given participant, cause-specific events are reported more than once are referred to as *recurrent events with competing risks* [13]. KM-type non-parametric estimators have been proposed for non-cause specific recurrent event survival analysis [24]. Typically, to handle scenarios where the recurring events for each participant are identical, the semi-parametric counting process algorithm [2] has been employed, where multiple time-intervals are considered as independent and a semi-parametric stratified Cox based approach has been used [26,13,25]. Note that very few papers handle scenarios which consider cause-specific recurring events in survival analysis. Multistage models with competing risks have been known to handle such scenarios with limited success [21].

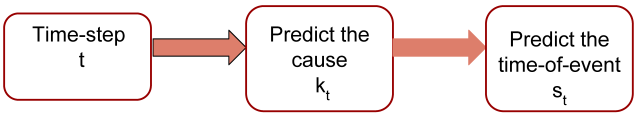

Fig. 2. Prediction Hierarchy: At each time-step t, predict the cause of the event (k_t) and the cause-specific time-to-event(s_t) for $t = 1, \ldots T$.

Contributions: In this work, we propose a general deep learning architecture which we call as *CRESA* for handling recurrence in the context of cause-specific survival analysis. Our approach consists of designing a deep neural network that directly learns the distributions of time-to-event and competing events. Our proposed technique does not suffer from the drawbacks related to conventional CPH models since we do not assume any underlying stochastic process or proportionality of hazards. Our loss-function can be split into two components, where the first component caters to the log-likelihood of the cause-specific recurrent event survival data, taking into account the censoring of the data and the second component considers the cause-specific ranking loss based on concordance [14]. The features of CRESA are as follows:

- In scenarios consisting of a single-risk ($C = 1$), CRESA is an LSTM-based deep neural network for recurrent event prediction for T time-steps, for scenarios where the covariates remain static or vary across time steps.
- In scenarios consisting of multiple risks ($C > 1$), CRESA is an LSTM-based deep neural network based architecture consisting of C cause specific sub-networks in order to handle C competing risks at each time-step. Hence, our neural network architecture is capable of predicting the time of the event and the cause of event at every time-step, as depicted in Fig. 2

To the best of authors' knowledge, this is the first instance where recurrent neural network based cause-specific models are proposed in the context of survival analysis. Since no assumptions are made on the underlying time-varying stochastic process governing the covariates and hitting times, this is one of the most flexible frameworks that handles most of the complex events that occur in a typical survival-analysis scenario. We use the concordance index (CI) per risk, and mean absolute error (MAE) at every time-step as the metric of performance. We demonstrate the efficacy of the proposed approach by comparing its predictive performance with traditional CPH model, Random survival forest (RSF) [9] and frailty based recurrent approaches [23]. We also compare the proposed approach with the more recent deep learning based DeepHit [14]

and DeepSurv [12] approaches. We test the different approaches on two real-life datasets and a synthetic dataset and demonstrate that the proposed technique performs better than traditional and modern approaches.

2 Cause-specific Recurrent Event Survival Analysis

In this section, we describe the cause-specific recurrent survival data, the neural network model which we develop for handling recurrent events, as well as the loss function that we use to train and test the model.

2.1 Survival Data

Typically, survival data for each participant is characterized by the individuals' covariates, time of event and censoring information. However, in the context of cause-specific recurrent event survival data, each participant is characterized by the following:

- Observed covariates in the given time-step.
- Time elapsed since the previous event or start of time.
- A label indicating the type of event in the given time-step.
- A label indicating the cause of the event in the given time-step.

We assume that the survival time is discrete and finite, i.e., the survival time takes values in a set $\mathcal{M} = \{1, \ldots, M\}$, defined for a maximum time horizon M. We assume that a participant may experience more than one type of competing risk, such that the corresponding label takes values from $\mathcal{C} = \{1, \ldots, C\}$. The fundamental difference between other temporal data and recurrent event survival data is the presence of *censored observations* in some or all time-steps. For example, in the context of reliability data, it is possible that a failure (considered as an event) occurs twice in the observation time period, leading to two same or different recurrent events experienced by the same participant. However, the participant may still be in the observation time period and report no further failures after the second time-step, which implies that the instance is censored beyond the second time-step. Handling such time-dependent and cause-specific censored data is a crucial aspect of this work.

Each instance is therefore a triple $(\mathbf{x}_t, s_t, d_t, k_t)$ where $\mathbf{x}_t \in \mathbf{X}_t$, denotes the set of covariates in the time-step t. Here, $s_t \in \mathcal{M}$ the time at which the event has occurred, $d_t \in \{0, 1\}$ denotes whether the i-th participant is censored or not, and $k_t \in \mathcal{C}$ denotes the label on the cause due to which an event is reported at time step t. We are given a dataset $D = (\mathbf{x}_t^{(i)}, s_t^{(i)}, d_t^{(i)}, k_t^{(i)})$, where $i = 1, \ldots, N$ denotes the number of participants. In the sequel, we describe the neural network model and the loss function we use to train the model using the above described dataset.

2.2 Model Description

Conventional approaches to survival analysis such as KM plots and the vanilla Cox models fail to provide meaningful insights into the cause-specific survival analysis based prediction tasks. This has led to alternative approaches such as the CIC approach which uses the marginal probabilities of an event in the presence of competing events, and does not require the assumption that these risks are independent. The corresponding cause-specific cumulative incidence function is given by [6]

$$F_k(s^*|\mathbf{x}^*) = P(S \leq s^*, K = k^*|\mathbf{X} = \mathbf{x}^*) = \sum_{s=0}^{s^*} P(S = s, K = k^*|\mathbf{X} = \mathbf{x}^*), \tag{1}$$

i.e., the CIF gives the probability that a particular cause k^* occurs on or before time s^* given the covariates $\mathbf{x}^*$. However, the scenarios that we consider in this paper involves predicting the *recurrent* hitting times, and hence we define the *recurrent* CIF (RCIF), which is the probability that a given event occurs on or before time s_t^* for different time steps t, given the time-dependent covariates $\mathbf{x}_t^*$, given as

$$F_k(s_t^*|\mathbf{x}_t^*) = P(S_t \leq s_t^*, K_t = k_t^*|S_{t-1} = s_{t-1}^*, \ldots, S_1 = s_1^*, K_{t-1} = k_{t-1}^*, \ldots, K_1 = k_1^*, \mathbf{X}_t = \mathbf{x}_t^*)$$
$$= \sum_{s_t=0}^{s_t^*} P(S_t = s_t, K_t = k_t^*|S_{t-1} = s_{t-1}^*, \ldots, S_1 = s_1^*, K_{t-1} = k_{t-1}^*, \ldots, K_1 = k_1^*, \mathbf{X}_t = \mathbf{x}_t^*). \quad (2)$$

Here, the function $P(X = x)$ represents the pdf of the random variable X evaluated at its realization x. Note that the true RCIF is not known, and it is not possible to directly compute it from the dataset. Hence, in order to quantitatively measure how models discriminate across cause-specific risks among participants, it is essential to design the neural network model that computes an approximate RCIF, which is subsequently incorporated into the loss function.

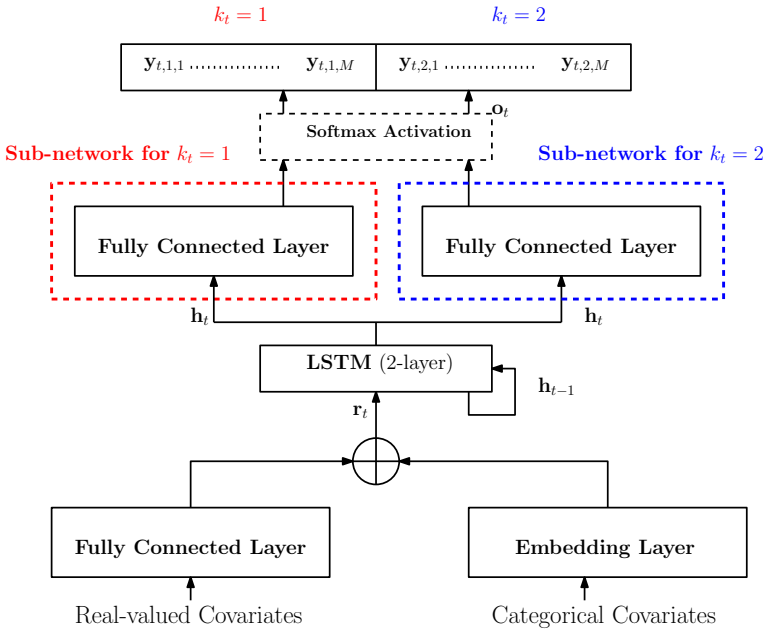

Fig. 3. CRESA consists of LSTM layer, cause-specific sub-networks and a single softmax activation layer.

Accordingly, in order to compute the estimates of the RCIF, we propose the following model, as illustrated in Fig. 3. We encode all real-valued covariates, ${\mathbf{x}_t}^{\text{real}}$ using a single fully connected layer with ReLU activation, denoted as MLP$(\cdot)$, and the categorical covariates, ${\mathbf{x}_t}^{\text{cat}}$ using an embedding lookup, denoted as Lookup$(\cdot)$. Using the concatenation operation denoted as $\bigoplus$, the dense representation, $\mathbf{r}_t$ of the covariates is computed as follows:

$$\mathbf{r}_t = \text{MLP}\left({\mathbf{x}_t}^{\text{real}}\right) \bigoplus \text{Lookup}\left({\mathbf{x}_t}^{\text{cat}}\right). \quad (3)$$

At each time step t, $\mathbf{r}_t$ is given as an input to the 2-layer LSTM [8] in order to obtain the hidden representation $\mathbf{h}_t \in \mathbb{R}^n$, where n is the number of hidden units in the LSTM, as follows:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{r}_t, \mathbf{h}_{t-1}). \tag{4}$$

The hidden state $\mathbf{h}_t$ is used to obtain C cause-specific representations by incorporating C single layer MLPs, represented as $\text{MLP}_k(\cdot)$. For brevity and purposes of this work we limit ourselves to cases where $C = 2$ but the model is easily generalizable. Thus, we obtain a final representation $\mathbf{o}_t$ as

$$\mathbf{o}_t = \text{MLP}_1(\mathbf{h}_t) \bigoplus \text{MLP}_2(\mathbf{h}_t). \tag{5}$$

Finally, $\mathbf{o}_t$ undergoes a softmax transform, which results in the final probability distribution function $\mathbf{y}_t = [y_{t,1,1}, ..., y_{t,1,M}, y_{t,2,1}, ..., y_{t,2,M}]$ is given by

$$\mathbf{y}_t = \textbf{Softmax}(\mathbf{o}_t). \tag{6}$$

Note that we consider the censored cases as a separate class in itself, i.e, the last class M accounts for the censored event. Hence, given a participant with covariates $\mathbf{x}_t$, an element y_{t,s_t,k_t} gives an estimate of $P(S_t = s_t, K_t = k_t | s_{t-1}, \ldots, s_1, k_{t-1}, \ldots, k_1, \mathbf{x}_t)$, i.e., $y_{t,s_t,k_t} = \hat{P}(S_t = s_t, K_t = k_t | s_{t-1}, \ldots, s_1, k_{t-1}, \ldots, k_1, \mathbf{x}_t)$. This architecture is independent of any model-based assumptions, and allows us to learn potentially non-linear relationships between covariates and risks, without any assumptions on the proportionality of hazards as in the Cox-based models [3].

2.3 Loss Function

A necessary component of the loss function is the log-likelihood of cause-specific recurrent event survival data, where the probabilistic information of the uncensored and censored observations are obtained from the failure density and the cumulative hazard function, respectively [13]. The loss function corresponding to the log-likelihood of data given by:

$$\mathcal{L}_1 = \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{1}_{(d_t^{(i)}=1)} \log \left(y^{(i)}_{t^{(i)}, s_t^{(i)}, k_t^{(i)}} \right) + \mathbb{1}_{(d_t^{(i)}=0)} \log \left(1 - \sum_{k=1}^{K} \hat{F}_k \left(s_t^{(i)} | \mathbf{x}_t^{(i)} \right) \right). \tag{7}$$

Here $\hat{F}_k \left(s_t^{(i)} | \mathbf{x}_t^{(i)} \right)$ represents the estimate of the risk, computed using (2) by substituting the true RCIF by their estimates. In addition to the log-likelihood, it has been observed that it is essential to penalize the cost function based on relative risks of uncensored participants. The central idea is from concordance [14] which states that at a given time-step t, a participant that experiences a time-to-event s_t should have a higher risk at s_t than a patient who survived longer than s_t. The risk-based penalty denoted by $\mathcal{L}_2$, is as follows:

$$\mathcal{L}_2 = \sum_{t=1}^{T} \alpha_{k,t} \sum_{i \neq j} A_t^{(k,i,j)} \eta \left[\hat{F}_k \left(s_t^{(i)} | \mathbf{x}_t^{(i)} \right), \hat{F}_k \left(s_t^{(i)} | \mathbf{x}_t^{(j)} \right) \right], \tag{8}$$

where $\alpha_{k,t}$ is a parameter that trades off between the log-likelihood and the concordance based loss for the k-th cause in the t-th time-step. Further, $\eta[x, y]$ is any convex loss function. Note that, $A_t^{(k,i,j)}$ indicates if participant i and j are both uncensored, with i experiencing the cause specific risk k, and hence can be compared for relative risks in a given time-step, i.e.,

$$A_t^{(k,i,j)} = \mathbb{1} \left(k_t^{(i)} = k_t, s_t^{(i)} < s_t^{(j)} \right). \tag{9}$$

Hence, the overall cost function is given by $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$. Time complexity of $\mathcal{L}_1$ is a linear function of N while time complexity for $\mathcal{L}_2$ is $\mathcal{O}(N^2 T)$.

3 Experiments

In this section, we first describe the datasets we employed, followed by a description of the baselines we used for comparison. We then elaborate on the training details followed by the results and discussion. We provide results for both single-risk and multi-risk cases.

3.1 Dataset I: MIMIC III Clinical Dataset

MIMIC-III (Medical Information Mart for Intensive Care) is a large, freely available clinical dataset developed by the MIT Lab for Computational Physiology [10]. It comprises of more than $40,000$ patient instances with information relating to patients admitted to critical care units at a hospital. This dataset contains multiple instances of the same patient being admitted due to a cause-specific risk, and discharged from ICU. We are interested in predicting the length of stay at ICU for recurrent admissions due to same or different competing risk. In a survival framework, we consider discharge of a patient from the ICU as the event of interest, and the observations are censored in the event of patients' death. We consider time-varying real-valued features ($\mathbf{x}_t{}^{\text{real}}$) such as blood pressure, glucose, heart rate, oxygen saturation, respiratory rate, temperature, pH, and categorical features ($\mathbf{x}_t{}^{\text{cat}}$) such as weight, height, ethnicity, gender, Glascow coma scale eye opening, Glascow coma scale motor response, Glascow coma scale verbal response as input covariates for LSTM. We use this dataset for single-risk and multi-risk survival recurrent analysis.

Single-risk Dataset: Single-risk dataset consists of 13021 instances of patients who are admitted into ICU recurrently, due to a single risk. These patients are admitted to the ICU recurrently for a maximum of $T = 5$ number of time-steps, and the time-to-event is one of $M = 101$ target classes.

Multi-risk Dataset: We consider $C = 2$ competing risks for predicting $M = 43$ number of classes (time-of-event occurrences) for each risk, recurring for $T = 5$ number of time steps.

3.2 Dataset II: Engine Failures Dataset

The engine failures dataset is a proprietary dataset which consists of instances of engines whose failures and months in service have been recorded for insurance related investigations. In this dataset, engine failure refers to the failure of a specific part of the engine, and the engine does not stop running after a failure. Cause-specific recurrent failures of the different parts have been reported which allows us to perform multi and single-risk, recurrent event survival analysis on this dataset.

We use real-valued features ($\mathbf{x}_t{}^{\text{real}}$) such as the number of miles covered, time duration since the build of engine and time since the engine is in-service, and categorical features ($\mathbf{x}_t{}^{\text{cat}}$) such as engine type, horse power, rotations per minute, application of engine, design configuration, equipment manufacturer, model name, etc. For this dataset the covariates do not change across time steps, and we expect that the hidden state of the LSTM will be the key differentiator while making predictions for different time steps.

Single-risk Dataset: Single-risk dataset consists of 18221 instances of engines which fail recurrently due to single risk over time. These engines experience a maximum of $T = 3$ recurrences (time-steps) of the risk, and their time-to-event is one of $M = 27$ classes.

Multi-risk Dataset: Multi-risk dataset consists of 31948 instances of engines with $C = 2$. The number of time steps is chosen as $T = 3$ and $M = 27$ represents the number of classes per part failure.

Engines which report failure for less than $T = 3$ are considered as censored for remaining time-steps.

3.3 Dataset III: Synthetic Dataset

We create a synthetic dataset with $C = 2$, $T = 7$, and $M = 7$ per risk, in order to demonstrate the time-tracking efficacy of the proposed framework. We constructed two stochastic processes over T time-steps with parameters and the hitting times inspired by the synthetic dataset proposed in [14]. The covariates for the participant i at time-step t are sampled as $\mathbf{x}^{(i)}_{1,\,t}, \mathbf{x}^{(i)}_{2,\,t}, \mathbf{x}^{(i)}_{3,\,t} \sim \mathcal{N}(0, \mathbf{I}_4)$. Further, the hitting times for a given time-step t are obtained as

$$s^{(i)}_{1,\,t} \sim \exp((\gamma_3^T \mathbf{x}^{(i)}_{3,\,t})^2 + (\gamma_1^T \mathbf{x}^{(i)}_{1,\,t})), \quad s^{(i)}_{2,\,t} \sim \exp((\gamma_3^T \mathbf{x}^{(i)}_{3,\,t})^2 + (\gamma_2^T \mathbf{x}^{(i)}_{2,\,t})). \tag{10}$$

In order to account for the recurrence of events, the hitting-times are varied over the time-steps using an autoregressive model given by

$$s^{(i)}_{k,\,t} = \rho_k s^{(i)}_{k,(t-1)} + \sqrt{1 - \rho_k^2} z^{(i)}_{k,\,t}, \tag{11}$$

where $z^{(i)}_{k,t} \sim \mathcal{N}(0, \mathbf{I})$ and the correlation co-efficient is chosen as $\rho_1 = \rho_2 = 0.6$. For convenience, we set $\gamma_1 = \gamma_2 = \gamma_3 = 0.4$, $s^{(i)}_t = \min[s^{(i)}_{1,\,t}, s^{(i)}_{2,\,t}]$ and participants with $s^{(i)}_t \geq 21$ are considered as censored events.

All datasets are divided into training, validation and test sets in the ratio $3 : 1 : 1$. Statistics of censored and uncensored participants across time steps is mentioned in Table 1.

Table 1. Number of censored and uncensored participants across time steps for each competing risks in 3 datasets

			t_1	t_2	t_3	t_4	t_5	t_6	t_7
Synthetic	Risk-1	Censored	597	1093	1562	2000	2452	2914	3280
		Uncensored	24322	23874	23443	23000	22508	22199	21635
	Risk-2	Censored	639	1082	1525	1999	2440	2850	3282
		Uncensored	24442	23951	23470	23001	22600	22037	21803
MIMIC	Risk-1	Censored	0	297	371	389	402	-	-
		Uncensored	2259	2045	1986	1973	1962	-	-
	Risk-2	Censored	0	888	970	989	990	-	-
		Uncensored	9408	1016	224	87	36	-	-
Engine	Risk-1	Censored	0	11455	14321	-	-	-	-
		Uncensored	15376	3443	887	-	-	-	-
	Risk-2	Censored	0	13099	15622	-	-	-	-
		Uncensored	16572	3951	1118	-	-	-	-

3.4 Training Details

The model described in Sec. 2.2 is trained via backpropagation using an SGD optimizer with momentum and Nesterov. We also use cosine annealing of the learning rate with warm restarts as in [18] using $T_0 = 1$ and $T_{mul} = 2$, with minimum and maximum learning rates of 0.005 and 0.5 respectively. Embedding sizes and all neural network hidden states have dimension 64 and dropout is used for regularization. We consider time independent value for $\alpha_{k,t}$, i.e., $\alpha_k = \alpha_{k,t}$, for $t = 1, \ldots, T$. The values of α_1, α_2 and dropout are determined using grid-search. Similar to [14], for $\mathcal{L}_2$, we use $\eta[x, y] = \exp(-(x - y)/\sigma)$ is where σ is set to 1.

3.5 Baselines

We use the following baselines for the **single-risk** scenario:

1. **DeepSurv**: DeepSurv model [1] is proposed in [12]. Post training, we run this model separately for T time steps and report results for each.
2. **DeepHit**: We simulate T DeepHit units [14] with $C = 1$. The hyperparameter setting is the same as in [14].
3. **Shared Frailty model**: We fit a shared Gamma frailty model [13] [23] and predict the marginal probability for the T time steps for each participant.

We use the following baselines for the **multiple risk** scenario:

1. **CRESA**$_\alpha$ **($\alpha_1 = 0$, $\alpha_2 = 0$)**: We fix the values of α_1 and α_2 as 0 for our proposed model and compare results to investigate the effect that $\mathcal{L}_2$ has on the results.
2. **DeepHit**: We simulate T DeepHit units [14] to obtain different distributions for T time steps, with $C > 1$. The hyperparameter setting is the same as in [14].
3. **Random Survival Forest (RSF)**: RSF [9] is trained for all time-steps using competing risks data, and post-training it is tested separately on each time-step.

3.6 Performance metrics and Results

In this subsection we describe the performance metric used to evaluate the neural network model proposed in Sec. 2.2. We use concordance index per time-step as one of the performance metrics given by:

$$\begin{aligned} CI_t &= P\left(\hat{F}\left(s_t^{(i)}|\mathbf{x}_t^{(i)}\right) > \hat{F}\left(s_t^{(i)}|\mathbf{x}_t^{(j)}\right)|s_t^{(i)} < s_t^{(j)}\right) \\ &\approx \frac{\sum_{i\neq j} A_t^{(k,i,j)} \mathbb{1}\left(\hat{F}\left(s_t^{(i)}|\mathbf{x}_t^{(i)}\right) > \hat{F}\left(s_t^{(i)}|\mathbf{x}_t^{(j)}\right)\right)}{\sum_{i\neq j} A_t^{(k,i,j)}} \end{aligned} \tag{12}$$

The other performance metric that we use is MAE, which is defined as follows:

$$MAE_t = \frac{1}{N}\sum_{i=1}^{N} d_t^{(i)}|y_t^{(i)} - s_t^{(i)}|, \tag{13}$$

where $d_t^{(i)}$ is zero if instance i is censored, and hence, the absolute error between the predicted label class is compared to the true label class only for instances which experience events.

We perform comprehensive evaluation of CRESA, and compare it with deep and non-deep baselines mentioned in the previous subsection using concordance index per risk per time-step, and MAE per time-step as performance metrics.

The concordance index results for the single-risk CRESA are given in Table 2. For the MIMIC III dataset, we observe that DeepHit does slightly better than CRESA for the first time-step. For the subsequent time-steps, the CRESA outperforms both baselines by a huge margin. However, CRESA does better for all the time-steps in the case of engine failures dataset. The results clearly show that the temporal information is an important aspect for recurrent event survival analysis which is captured in our model by the LSTM, and hence, the LSTM-based CRESA model performs better than DeepHit which, by its nature, does not exploit any temporal information.

[1] `https://github.com/jaredleekatzman/DeepSurv`

Table 2. Concordance indices for single-risk, recurrent survival analysis

Model used	CI_1	CI_2	CI_3	CI_4	CI_5
MIMIC III Clinical Dataset					
CRESA	0.671	**0.910**	**0.896**	**0.878**	**0.866**
DeepHit	**0.678**	0.808	0.723	0.665	0.633
DeepSurv	0.586	0.775	0.715	0.681	0.660
Shared Frailty	0.3386	0.489	0.534	0.5105	0.521
Engine Failures Dataset					
CRESA	**0.792**	**0.931**	**0.959**	-	-
DeepHit	0.765	0.759	0.849	-	-
DeepSurv	0.509	0.503	0.560	-	-
Shared Frailty	0.459	0.482	0.496	-	-

The performance of DeepHit and CRESA is similar for the first time-step since CRESA does not have access to any time dependent *hidden state*. As expected, the proposed approach performs better as compared to traditional, model-based shared frailty approach by a huge margin in all the time-steps.

Table 3. Concordance indices for multi-risk, recurrent survival analysis

	Risk-1							Risk-2						
	CI_1	CI_2	CI_3	CI_4	CI_5	CI_6	CI_7	CI_1	CI_2	CI_3	CI_4	CI_5	CI_6	CI_7
MIMIC III														
CRESA	0.680	**0.844**	**0.779**	**0.759**	**0.735**	-	-	0.646	**0.855**	**0.802**	**0.780**	**0.768**	-	-
CRESA$_\alpha$	**0.704**	0.761	0.667	0.631	0.602	-	-	0.650	0.778	0.684	0.654	0.632	-	-
DeepHit	0.658	0.686	0.620	0.595	0.578	-	-	0.629	0.773	0.673	0.643	0.621	-	-
RSF	0.474	0.494	0.518	0.521	0.546	-	-	**0.77**	0.74	0.77	0.77	0.746	-	-
Engine Failures Dataset														
CRESA	0.801	**0.874**	**0.907**	-	-	-	-	**0.864**	**0.912**	0.909	-	-	-	-
CRESA$_\alpha$	0.643	0.692	0.633	-	-	-	-	0.750	0.735	0.650	-	-	-	-
DeepHit	0.726	0.827	0.890	-	-	-	-	0.813	0.882	**0.944**	-	-	-	-
RSF	**0.81**	0.77	0.78	-	-	-	-	0.81	0.792	0.797	-	-	-	-
Synthetic Dataset														
CRESA	**0.762**	**0.754**	0.747	**0.729**	**0.728**	0.715	**0.715**	**0.774**	0.759	**0.737**	**0.736**	**0.73**	**0.715**	**0.72**
CRESA$_\alpha$	0.704	0.698	**0.751**	0.720	0.712	**0.736**	0.701	0.761	**0.761**	0.710	0.713	0.723	0.712	0.69
DeepHit	0.579	0.583	0.568	0.582	0.562	0.554	0.558	0.587	0.581	0.571	0.563	0.56	0.554	0.56
RSF	0.509	0.495	0.517	0.502	0.509	0.498	0.512	0.514	0.517	0.505	0.497	0.513	0.51	0.51

The concordance index results for the multi-risk CRESA are given in Table 3. First, we note that CRESA performs better compared to CRESA$_\alpha$ across datasets, from which we infer that $\mathcal{L}_2$ indeed has an impact on the final values of concordance index. Even in the presence of multiple risks, CRESA performs better compared to the baseline schemes such as DeepHit and RSF except in the first time-step and hence, CRESA continues to gain from the backbone LSTM architecture in the time-steps other than the first time-step, as in the single-risk scenario. From the results pertaining to synthetic dataset, we infer that the advantages that we obtain by using LSTM based architecture carries over to several time-steps.

Table 4. Mean Absolute Error (MAE) for multi-risk, recurrent survival analysis

Model used	MAE_1	MAE_2	MAE_3	MAE_4	MAE_5	MAE_6	MAE_7
MIMIC III Clinical Dataset							
CRESA	**0.7152**	**0.3475**	**0.3049**	**0.307**	**0.3042**	-	-
DeepHit	0.734	0.3544	0.3136	0.3099	0.3066	-	-
Engine Failures Dataset							
CRESA	**0.8998**	**0.441**	**0.3526**	-	-	-	-
DeepHit	0.989	0.5275	0.4161	-	-	-	-
Synthetic Dataset							
CRESA	**0.5833**	**0.5987**	**0.6078**	**0.6146**	**0.6396**	**0.6503**	**0.658**
DeepHit	0.6598	0.6708	0.6832	0.6927	0.7141	0.718	0.723

In Table 4, we present the MAE performance of CRESA as compared to DeepHit. First, we notice that the MAE performance of both the schemes are poor in the first time-step as compared to the subsequent time-steps, in the case of real-life datasets. However, the MAE results are uniform across all time-steps for the synthetic dataset. Note that real-life datasets have large number of classes as compared to the synthetic counterpart. In the first time-step, the true label distribution across the classes are close to being uniform in the real-life datasets, and hence, CRESA faces difficulty in learning the true distribution. However, in the subsequent time-steps, this distribution across the classes is skewed, and hence learning becomes easier. We suspect that a larger dataset and a more complex model will improve the MAE results. Although synthetic datasets have fewer classes, the true label distribution is uniform across time-steps, and hence, uniform trend the MAE results is observed over all the time-steps.

4 Conclusions

In this paper, we proposed CRESA, a novel deep learning architecture for competing risk, recurrent event survival analysis. CRESA employed an LSTM based backbone recurrent neural network to estimate the RCIF, which is the joint distribution of the time-to-event and competing risks, as a function of the covariates in the data. For the single-risk scenario, the CRESA architecture reduces to an LSTM-based deep neural network for recurrent event prediction for T time-steps. We used a loss function that exploited the CIC-based probabilistic information from uncensored and right-censored participants and also penalized incorrect ordering of relative risks in every time-step. We compared the performance of CRESA with the performance of traditional approaches such as frailty-based CPH model and RSF, and modern deep learning approaches such as DeepHit and DeepSurv. We demonstrated that CRESA has a superior performance in terms of both, concordance index and MAE. We also noted that MAE has a strong dependence on the nature of the dataset, such as the number of classes and the distribution of true labels across the classes. Overall, CRESA lends itself as a flexible, non-model based approach to survival analysis for datasets that involve complex events such as recurrence of events and competing risks. In future, we would extend architecture and loss function of CRESA to handle interval and left censoring events as well.

References

1. Alaa, A.M., van der Schaar, M.: Deep multi-task Gaussian processes for survival analysis with competing risks. In: 30th Conference on Neural Information Processing Systems (2017)
2. Andersen, P.K., Gill, R.D.: Cox's regression model for counting processes: a large sample study. The annals of statistics pp. 1100–1120 (1982)
3. Cox, D.R.: Analysis of survival data. Routledge (2018)
4. Doksum, K.A., Hbyland, A.: Models for variable-stress accelerated life testing experiments based on wener processes and the inverse gaussian distribution. Technometrics **34**(1), 74–82 (1992)
5. Faraggi, D., Simon, R.: A neural network model for survival data. Statistics in medicine **14**(1), 73–82 (1995)
6. Fine, J.P., Gray, R.J.: A proportional hazards model for the subdistribution of a competing risk. Journal of the American statistical association **94**(446), 496–509 (1999)
7. Gray, R.J.: A class of K-sample tests for comparing the cumulative incidence of a competing risk. The Annals of statistics pp. 1141–1154 (1988)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
9. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S., et al.: Random survival forests. The annals of applied statistics **2**(3), 841–860 (2008)
10. Johnson, A.E., Pollard, T.J., Shen, L., Li-wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: MIMIC-III, a freely accessible critical care database. Scientific data **3**, 160035 (2016)
11. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. Journal of the American statistical association **53** (1958)
12. Katzman, J.L., Shaham, U., Cloninger, A., et al.: DeepSurv: Personalized treatment recommender system using a cox proportional hazards deep neural network. BMC medical research methodology **18**(1), 24 (2018)
13. Kleinbaum, D.G., Klein, M.: Survival analysis, vol. 3. Springer (2010)
14. Lee, C., Zame, W.R., Yoon, J., van der Schaar, M.: Deephit: A deep learning approach to survival analysis with competing risks (2018)
15. Lee, M.L.T., Whitmore, G.: Proportional hazards and threshold regression: their theoretical and practical connections. Lifetime data analysis **16**(2), 196–214 (2010)
16. Liao, L., Ahn, H.i.: Combining deep learning and survival analysis for asset health management. International Journal of Prognostics and Health Management (2016)
17. Longini, I.M., Clark, W.S., Byers, R.H., Ward, J.W., Darrow, W.W., et al.: Statistical analysis of the stages of HIV infection using a Markov model. Statistics in medicine **8** (1989)
18. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts (2016)
19. Luck, M., Sylvain, T., Cardinal, H., Lodi, A., Bengio, Y.: Deep learning for patient-specific kidney graft survival analysis. arXiv preprint arXiv:1705.10245 (2017)
20. Lunn, M., McNeil, D.: Applying Cox regression to competing risks. Biometrics pp. 524–532 (1995)
21. Meira-Machado, L., de Uña-Álvarez, J., Cadarso-Suárez, C., Andersen, P.K.: Multi-state models for the analysis of time-to-event data. Statistical methods in medical research **18**(2), 195–222 (2009)
22. Ranganath, R., Perotte, A., Elhadad, N., Blei, D.: Deep survival analysis. arXiv preprint arXiv:1608.02158 (2016)
23. Rondeau, V., Mazroui, Y., Gonzalez, J.R.: Frailtypack: an R package for the analysis of correlated survival data with frailty models using penalized likelihood estimation or parametrical estimation. J Stat Softw **47**(4), 1–28 (2012)
24. Wang, M.C., Chang, S.H.: Nonparametric estimation of a recurrent survival function. Journal of the American Statistical Association **94**(445), 146–153 (1999)
25. Wang, P., Li, Y., Reddy, C.K.: Machine learning for survival analysis: A survey. arXiv preprint arXiv:1708.04649 (2017)
26. Wei, L.J., Lin, D.Y., Weissfeld, L.: Regression analysis of multivariate incomplete failure time data by modeling marginal distributions. Journal of the American statistical association **84** (1989)
27. Zhu, X., Yao, J., Huang, J.: Deep convolutional neural network for survival analysis with pathological images. In: Bioinformatics and Biomedicine (BIBM). pp. 544–547. IEEE (2016)