Knowledge and Information Systems
https://doi.org/10.1007/s10115-020-01472-1

REGULAR PAPER

Survival neural networks for time-to-event prediction in longitudinal study

Jianfei Zhang[1,3] · Lifei Chen[1,2] · Yanfang Ye[3] · Gongde Guo[2] · Rongbo Chen[1] · Alain Vanasse[4,5] · Shengrui Wang[1,2]

Received: 15 September 2019 / Revised: 6 April 2020 / Accepted: 11 April 2020

Abstract
Time-to-event prediction has been an important practical task for longitudinal studies in many fields such as manufacturing, medicine, and healthcare. While most of the conventional survival analysis approaches suffer from the presence of censored failures and statistically circumscribed assumptions, few attempts have been made to develop survival learning machines that explore the underlying relationship between repeated measures of covariates and failure-free survival probability. This requires a purely dynamic-data-driven prediction approach, free of survival models or statistical assumptions. To this end, we propose two real-time survival networks: a time-dependent survival neural network (TSNN) with a feed-forward architecture and a recurrent survival neural network (RSNN) incorporating long short-term memory units. The TSNN additively estimates a latent failure risk arising from the repeated measures and performs multiple binary classifications to generate prognostics of survival probability, while the RSNN with time-dependent input covariates implicitly estimates the relation between these covariates and the survival probability. We propose a novel survival learning criterion to train the neural networks by minimizing the censoring Kullback–Leibler divergence, which guarantees monotonicity of the resulting probability. Besides the failure-event AUC, C-index, and censoring Brier score, we redefine a survival time estimate to evaluate the performance of the competing models. Experiments on four datasets demonstrate the great promise of our approach in real applications.

Keywords Time-to-event prediction · Longitudinal study · Survival neural network · Classification probability · Time-dependent covariate

1 Introduction

Longitudinal study is an observational research method in which quantitative and/or qualitative data are collected by repeated measures (i.e., at several time points) of the same covariates, following particular individuals over a prolonged period of time—e.g., years or decades [7]. Longitudinal studies involve a great deal of effort but offer several benefits. The main goal is to investigate how changes in the variables impact outcomes over different periods of time [47]. For example, the Canadian Longitudinal Study on Aging (CLSA) was

Extended author information available on the last page of the article

Published online: 21 May 2020

designed to follow more than 50,000 participants between the ages of 45 and 85 until 2033 or their death.[1] The CLSA gathers information on biological, medical, psychological, social, lifestyle and economic factors. Researchers working on CLSA data try to gain knowledge about the effect of these factors, both separately and in combination, on the development of disease and disability as people age.

In longitudinal studies, time-to-event data arise when interest is focused on the time (years, months, weeks, or days) elapsing from the beginning of the study (i.e., baseline) until an event occurs. In real applications, the event of interest may be any failure experience: in manufacturing, for instance, it could be equipment failure or device fault; in healthcare, disease recurrence, hospital readmission or death; in finance, regime change in a stock market. This paper is about predicting non-recurring, single, adverse events. For instance, think about all the in-service machines we use daily, an engine-propelled vehicle on the way to work or a lift going up and down; or consider an in-patient in the early stages of breast cancer. Imagine that one of these should fail (e.g., equipment should break down, or a patient should worsen or die) one day from now. What impact would that have? The truth is that some failure events are merely an inconvenience or a financial loss, while others could mean life or death. For this reason, predicting the time of a failure event (failure event time) has long been of practical interest. With predicted event time, clinicians can answer patients' queries on probable outcomes in a timely fashion, and decision makers can obtain information about when a mechanical fault that can lead to complete system failure might take place.

Predicting event time actually involves answering questions like *How many days are left before the failure event?*. In the observational world, the outcome (i.e., response variable) is not only whether or not an event occurred, but also when that event occurred. However, naïve regression models are not able to include both the event and time aspects as the outcome in the model. Predicting event time accurately is very challenging and indeed almost impossible in most practical situations. Instead, we propose to turn our attention to the easier and more meaningful problem *how long, with what probability, will failure-free survival be maintained?*. This question has been addressed by a considerable number of survival analysis models developed to utilize the partial information on individuals with censored events (due to incomplete follow-up, or dropout of study participants) and provide unbiased estimates. With predicted outcomes, one may estimate that an 80-year-old female patient diagnosed with breast cancer has a 50% probability of surviving one year and a 20% probability of surviving 3 years; this prognostic can be crucial in the choice of treatments, lifestyle modifications and, sometimes, end-of-life care measures. The predicted probability of fault-free steering up to 50,000 km in a 15-year-old vehicle engine is 80%, but for 80,000 km the probability drops to 10%; this knowledge will allow for preventive maintenance (before 50,000 km), which may prolong engine usage and holds out the promise of considerable cost savings.

In survival analysis, observations in terms of specific covariates are generally collected at a single point in time at the beginning of the study (baseline), and participants are then followed for a period of time subsequent to this baseline time point to examine associations of these covariates with outcomes (i.e., event times). However, data collection for longitudinal studies usually involves a vast number of repeated measures that make the related covariates time-dependent. For example, HIV patients may be followed over time and undergo monthly measures such as CD4 counts; or aging studies investigating cognitive change repeatedly collect information about participants' cognitive function and date of death or dementia diagnosis (e.g., repeated measures in healthy aging records for Canadians in 2010; see Table 1), in terms of time-dependent covariates such as Sports, Mental, Fall-caused injury, and Smoking.

[1] https://www.clsa-elcv.ca/about-us/about-clsa-research-platform.

Table 1 An example of repeated measures: healthy aging records for Canadians in 2010

ID	Age	Province	Time (in past)	Sports	Mental health	Fall-caused injury	Smoking
1	45	Québec	12 months	Seldom	Excellent	No	Yes
			6 months	Sometimes	Fair	Yes	Yes
			1 month	Often	Poor	No	No
2	54	Ontario	12 months	Often	Excellent	No	Yes
			6 months	Often	Excellent	No	Yes
			1 month	Sometimes	Fair	No	Yes

Repeated measures require special techniques for valid analysis and inference. One may use methods such as time-varying covariate analysis [9,44,52,56], in which the single measures may be updated over time and replaced with subsequent measures to examine short-term associations, or time-averaged analysis [38,55], in which a participant's baseline measure is represented only by the values of covariates at a certain moment or a single estimate derived from the average of measures over a specified period. It is worth noting that historical measures have been proven in the literature [34,54] to latently affect the survival probability. For example, in an observational study of the effects of a drug on specific health indicators, a patient's current health status may influence the drug exposure or dosage received in the future. It is thus highly desirable to establish a model that can automatically account for time-dependent covariates.

Overall, current methods have serious limitations with respect to longitudinal studies and practical use. Most survival analysis models suffer from the implausibility of the survival study hypothesis and the need for prior knowledge, e.g., the distributional assumption regarding event times widely postulated in parametric approaches [20]. In addition, the relation between the time-dependent covariates and the event time is assumed to be linear, possibly with some interaction terms; this limits the kinds of relationship that may be discovered. Interestingly, neural networks have achieved significant advances in nonlinear modeling and state-of-the-art performance in survival analysis studies [16,17,22,25,49]. We believe that a neural network classifier is a potential candidate for performing survival prediction, without any statistical assumptions. In our recent work [53], a time-dependent survival neural network (TSNN) was developed to account for time-dependent covariates and their nonlinear relation, achieving high performances in survival prediction.

In light of the discussion above, we extend our previous work [53] and propose to predict time-to-event in longitudinal studies by means of our TSNN and the specifically developed recurrent survival neural network (RSNN). These two real-time survival networks yield multiple classification outcomes over time, performing prognostics of event-free survival probability:

- TSNN additively estimates a latent risk of event based on the repeated measures of time-dependent covariate, characterized by a multi-output one-hidden-layer feed-forward network architecture.
- Given a set of time-dependent input covariates, RSNN implicitly estimates the relation between these covariates and survival probability, using a network of long short-term memory (LSTM) units.

The designated censoring Kullback–Leibler (KL) divergence for quantifying the dissimilarity between the binary classification probabilities and the actual survival statuses. A generalized survival learning approach is then used to minimize the censoring KL divergence, running under a constraint that guarantees monotonicity of the resulting probability. We tested TSNN and RSNN on four real-world datasets from the fields of engineering, medicine and aging studies, in comparison with the baseline competitors considered by the previous work [53] and, additionally, the Cox regression model [10], the Cox model using the average of repeated measures (CoxAvg) [51], and the RNN for survival analysis (RNN-SURV) [16]. Experimental results demonstrate the promise of real-time survival networks in real applications. The paper makes the following primary contributions:

- A purely dynamic-data-driven prediction approach, free of any existing survival model or statistical assumptions, is developed for survival prediction.
- Time-to-event prediction via survival regression analysis is transformed into multiple nonlinear classifications via feed-forward neural networks and recurrent neural networks.
- Repeated measures in longitudinal studies are analyzed, while the underlying relations between the time-dependent covariates and the event time are considered.
- A survival criterion is proposed to allow neural networks to learn from time-to-event data with censored response variables (i.e., survival times).
- An estimate of survival time and corresponding survival error metric used to evaluate the absolute error of survival prediction.

2 Related work

This paper is about longitudinal and time-to-event data analysis, so we will review the work related to survival models for time-to-event data analysis. Broadly speaking, survival analysis methods can be classified into two main categories: statistical methods and machine-learning-based methods, which share the common goal of predicting time of event. Statistical methods focus more on event time distributions and the properties of the parameter estimation. Machine learning methods are usually applied to complex problems such as massive high-dimensional data and nonlinear data fitting.

2.1 Statistical survival models

Statistical models can be grouped into parametric, nonparametric, and semi-parametric approaches, developed primarily for retrospective cohort studies, each of which has their inherent disadvantages.

- In nonparametric approaches, an empirical estimate of the survival function typically uses the Nelson–Aalen estimator [1], the Kaplan–Meier estimator [21] or the life-table method [11]. These approaches are intended to generate unbiased descriptive statistics, but generally cannot be used to assess the effect of multiple covariates on the response variable (i.e., survival probability).
- Parametric approaches commonly assume that the event time is drawn from an exponential, Weibull, Gompertz–Makeham, (log-)normal, logistic, (log-)logistic, or gamma distribution. Typical examples include the accelerated failure time (AFT) model [45], the Buckley–James model [6], and penalized regression [40]. These approaches suffer from a critical weakness, relying as they do on the assumption that the underlying failure distribution (i.e., how the probability of failure changes over time) has been correctly specified. If the distribution does not correspond to the inferences, these approaches can be grossly invalid.

- Most researchers in the survival analysis field have been more inclined to use the semi-parametric Cox proportional-hazards model [10], because of its ease of use, proven effectiveness and interpretability of results. The extensive models are constrained by regularized coefficients [29,42] or trained by a gradient-boosting algorithm [5]. However, semi-parametric approaches make an assumption on how the covariates influence the risk of failure, which is often violated in practical use.

2.2 Machine learning for survival analysis

The increasing availability of a wide variety of data (e.g., time-dependent covariates) poses more challenges to the statistical approaches and is stimulating numerous research efforts that use machine learning methods in conjunction with survival models.

2.2.1 Neural network-based methods

- *Feed-forward neural networks* Liestbl et al. [30] subdivided time into multiple intervals, assumed the hazard of event to be constant in each interval and proposed a feed-forward neural network with a single hidden layer that outputs the conditional event probabilities for each patient. This work was then expanded in [4], but even in this later work the value of the estimate for a given patient is not utilized in computing the estimate for the same patient. In order to generalize the Cox model, Faraggi and Simon [13] utilized nonlinear functions instead of the traditional linear combinations of covariates, by modeling the relationship between the input covariates and the corresponding risk with a single-hidden-layer feed-forward neural network.
- *Deep learning* Although the feed-forward network can preserve most of the advantages of a typical Cox proportional-hazards hypothesis, it was still not the optimal way to model the baseline variations. This was the rationale for another study [22], which described the interactions between a patient's covariates and treatment effectiveness in order to provide personalized treatment recommendations. A deep neural network was used in [25] to learn the distribution of survival times directly and to allow the possibility of assessing the relationship between covariates and risk over time, without assumptions about the underlying stochastic process.
- *Recurrent neural networks* Giunchiglia et al. [16] proposed a recurrent neural network model that computes the survival function by considering a series of binary classification problems, each leading to the estimation of the survival probability in a given interval of time. For check-in time prediction, Yang et al. [49] built a recurrent-censored regression model to capture the spatiotemporal nature of check-in data, where a gated recurrent units (GRUs) structure was designed to learn a latent representation of historical check-ins by a user. Another application is to predict user return time: Grob et al. [17] constructed an RNN-based survival model based on user sessions and their associated covariates.

2.2.2 Other machine learning methods

Besides neural networks, typical examples include multi-task learning, Gaussian process, Bayesian inference, active learning, transfer learning, feature engineering, etc.

- *Multi-task learning* The Cox regression model was used by Wang et al. [43], in conjunction with multi-task learning, encoding relatedness between tasks via coefficient regularization. A multi-task logistic regression (MTLR) model was first established by Lin et al. [31] to learn patient-specific survival distributions, directly modulating survival probability via a combination of multiple local logistic regression models in a dependent manner. In the critical phase of handling censored data, Li et al. [27] reformulated the survival problem as a multi-task learning problem. Later on, they succeeded in predicting talent career paths through multi-task learning—formulating the prediction of survival status at a sequence of time intervals [26]. Bellot and van der Schaar [3] leveraged an interpretation of boosting algorithms in a multi-task learning framework, while making each task-specific time-to-event distribution a component of a multi-output function.

- *Gaussian process and Bayesian inference* Kim and Pavlovic [24] developed scalable variational inference algorithms for a Gaussian process survival analysis model and uncertainty in the hazard function modeling. Furthermore, Fernández et al. [14] provided a semi-parametric Bayesian model for survival analysis by a Gaussian process, which modulates the hazard function by the multiplication of a parametric baseline hazard and a nonparametric part. Similarly, in [2] a nonparametric Bayesian model was developed by deep (multi-task) Gaussian processes. This method is Bayesian since the authors assigned a prior distribution over a space of vector-valued functions of the patients' covariates, and updated the posterior distribution given a time-to-event dataset.
- *Active learning and transfer learningl* Vinzamuri et al. [41] presented an active regularized Cox regression framework which effectively integrates active learning and the Cox regression using a model discriminative gradient sampling strategy and robust regularization. The transfer-learning-based Cox model [28] uses auxiliary data to augment learning when there is an insufficient number of training samples. This model uses the 1,2-norm penalty to encourage multiple covariates to share similar sparsity patterns, thus learning a shared representation across source and target domains, potentially improving the model's performance on time-to-event data.
- *Feature engineering* Li et al. [29] proposed a unified model for regularized parametric survival regression for an arbitrary survival distribution, using the elastic net [57] as a sparsity-inducing penalty to effectively deal with high-dimensional data. Yu et al. [50] performed a feature selection via affine projections in the Cox model to achieve privacy preservation when different clinical institutes share clinical data with each other.
- *Decision trees* Ishwaran et al. [19] proposed the random survival forests model, an ensemble tree method for analytics of right-censored survival data. It utilizes a nonparametric Nelson–Aalen estimator to predict the time to censored failures for establishing terminal nodes of the forest.

2.3 Time-varying analysis

The association between repeated measures and the outcome has been modeled in various ways. In practice, the covariates' effects (e.g., the effect of a treatment) may change over time with longer follow-up [15]. To accommodate such situations, there has been a surge of interest in learning time-varying coefficients instead of time-invariant ones. The varying coefficient models are a very important tool to explore the dynamic pattern. They are a natural extension of classical parametric models, with good interpretability, and are becoming more and more popular in data analysis [12].

- For the Cox model, Tian et al. [39] estimated time-varying coefficients by maximizing a kernel-weighted partial likelihood, while Sun et al. [37] employed a local empirical partial likelihood smoothing. Time-varying coefficients were also used in [32] to describe the potential time-varying effects of covariates on breast cancer (in the New York University women's health cohort study).

- The proportionality assumption may not hold in practice when covariate effects change over time. The semi-proportional hazards model (SPH) [52] uses locally time-varying coefficients, thereby relaxing the proportional hazards assumption for the sake of practicality, while retaining that model's simplicity.
- Rather than the Cox regression, [31] utilize a logistic regression in which the time-varying coefficients are regularized and smoothed.

3 Proposed approach

In this section, we first introduce the motivation of using multiple survival classifications and then provide the two real-time survival networks and evaluate the survival probability, finally the learning phrase.

3.1 Motivation of using multiple survival classifications

Using a similar presentation to that employed in longitudinal studies, we denote by T a continuous nonnegative random variable representing the time of event. Supposing that an event occurs in a specific time period, say $[0, t]$, then we have a cumulative distribution function in closed form $F(t) = \Pr(T \leq t)$ for time of event, which is particularly useful for analyzing time-to-event data with censoring. A binary classification performed to predict failure of a machine in a given t-day time window would allow us to answer the question *Will the machine remain failure-free over the next t days?*. Thus, we can transform the original time-to-event prediction problem into a series of binary classification problems, as long as each has an output probability that the actual survival time is not earlier than t, denoted $\Pr(T > t)$. Hence, we think of a classifier that can output multiple binary classification probabilities over disjoint time snapshots $\tau_1 < \tau_2 < \cdots < \tau_K$, each depicting the probability of remaining event-free at time τ_k, answering questions like *How does the risk of event change over time?*.

Each classification can be performed by means of a nonparametric logistic sigmoid function σ, in the general form $S(t) = \sigma(-q \ln(\alpha t)) = (1+(\alpha t)^q)^{-1}$, where the random variable α allows us to differentiate between the classes—"failure event" and "failure-event-free (survival)." To analyze the simultaneous effects of covariates $\mathbf{x} = (x_1, x_2, \ldots, x_V) \in \mathbb{R}^V$ on an event, in a general case, one usually introduces those variables that affect α but not q by overwriting $\ln \alpha$ with a link function of $\mathbf{x}$, that is, $\psi(\mathbf{x}) = \ln \alpha$ and therefore $S(t|\mathbf{x}) = \sigma(\psi(\mathbf{x}))$. Given V covariates for N individuals, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathbb{R}^{N \times V}$, the classifier is designed to generate an outcome vector of survival probabilities at τ_k for these individuals. Survival probability curves can be plotted as long as the classifier yields K outcomes over the disjoint time points.

3.2 Real-time survival networks

Basically, generalized linear models (GLMs) postulate the link function $\psi(\mathbf{x})$ in linear form. In most practical applications, however, the functional relationship between covariates and the output is not linear. For this reason, instead of replacing the linear function by the network output, we resort to artificial neural networks capable of performing a probabilistic

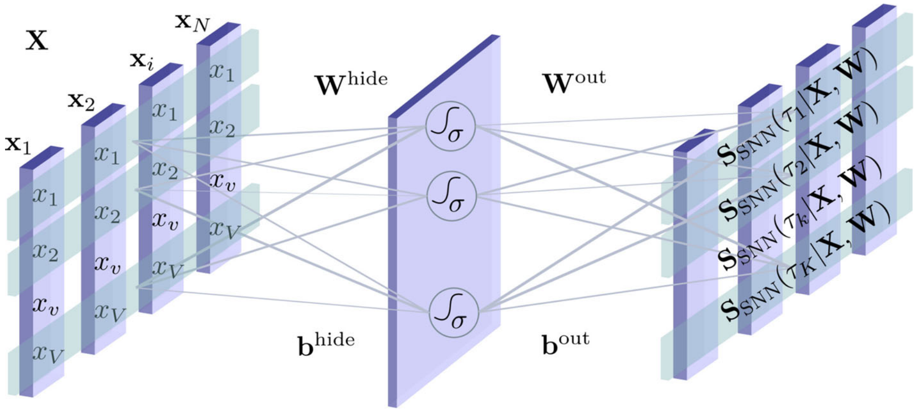

$\mathbf{X}$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_i$
$\mathbf{x}_N$
x_1
x_2
x_v
x_V
$\mathbf{W}^{\text{hide}}$
$\mathbf{W}^{\text{out}}$
$\mathbf{b}^{\text{hide}}$
$\mathbf{b}^{\text{out}}$
$\int_\sigma$
$\mathbf{S}_{\text{SNN}}(\tau_1|\mathbf{X},\mathbf{W})$
$\mathbf{S}_{\text{SNN}}(\tau_2|\mathbf{X},\mathbf{W})$
$\mathbf{S}_{\text{SNN}}(\tau_k|\mathbf{X},\mathbf{W})$
$\mathbf{S}_{\text{SNN}}(\tau_K|\mathbf{X},\mathbf{W})$

Fig. 1 Baseline survival neural network in a feed-forward architecture

classification and automatically identifying the nonlinear relationship between covariates and our desired survival probability. In this section, we introduce the feed-forward neural network applicable for survival prediction and propose two survival neural networks that allow dynamic data (time-dependent covariates) to drive the survival learning inference, i.e., free of survival models or statistical assumptions, to perform the binary classifications. In principle, these survival networks are combinations of multiple classifiers, each performing a binary classification on every individual that may or may not be event-free.

3.2.1 Time-dependent survival neural network

We concentrate our attention on a one-hidden-layer feed-forward neural network, i.e., a three-layer network with V input neurons, K output neurons and D hidden neurons, as shown in Fig. 1. The input layer's role is solely to distribute the inputs to the hidden layer, where the neuron $v = 1, 2, \ldots, V$ takes value x_v and the hidden neuron $d = 1, 2, \ldots, D$ computes a sum of all the inputs weighted by $\mathbf{w}_d^{\text{hide}} \in \mathbb{R}^V$, adds a bias b_d^{hide}, and applies an activation function to obtain its output. The outputs of the hidden layer subsequently become the inputs of the output layer, in which the output neuron $k = 1, 2, \ldots, K$ computes a sum of these inputs weighted by $\mathbf{w}_k^{\text{out}} \in \mathbb{R}^D$, adds a bias b_k^{out}, and then applies the activation function to obtain survival probabilities. Given the individuals $\mathbf{X}$, with the weights $\mathbf{W}^{\text{hide}} \in \mathbb{R}^{D\times V}$ and $\mathbf{W}^{\text{out}} \in \mathbb{R}^{K\times D}$ and the biases $\mathbf{b}^{\text{hide}} \in \mathbb{R}^D$ and $\mathbf{b}^{\text{out}} \in \mathbb{R}^K$ for computing the hidden and output layers, respectively, we scale the N outputs of our baseline survival neural network (SNN) at τ_k to the range of the logistic sigmoid function σ applied component-wise to the vector, i.e.,

$$\mathbf{S}_{\text{SNN}}(\tau_k|\mathbf{X},\mathbf{W}) = \sigma\left(\mathbf{w}_k^{\text{out}} \cdot \sigma\left(\mathbf{W}^{\text{hide}}\mathbf{X} + \mathbf{b}^{\text{hide}}\right) + b_k^{\text{out}}\right) \tag{1}$$

Note that the exponential component in Eq. 1 can serve as the risk of event, like the conventional cumulative risk found in the Cox [10] and AFT models [45]. Obviously, such an event risk is supposed to be totally independent of any historical covariate values. To address this issue, we propose the form $\gamma(t', t)$ to stand for the decay ratio of the event risk. By such decay, we can model the amount of the latent risk produced by the values at time t' remaining at time $t(\geq t')$. This can be an exponential function of time in the form $\gamma(t', t) = \exp\{\xi(t' - t)\}$. Simply, we make the decay coefficient ξ take a positive

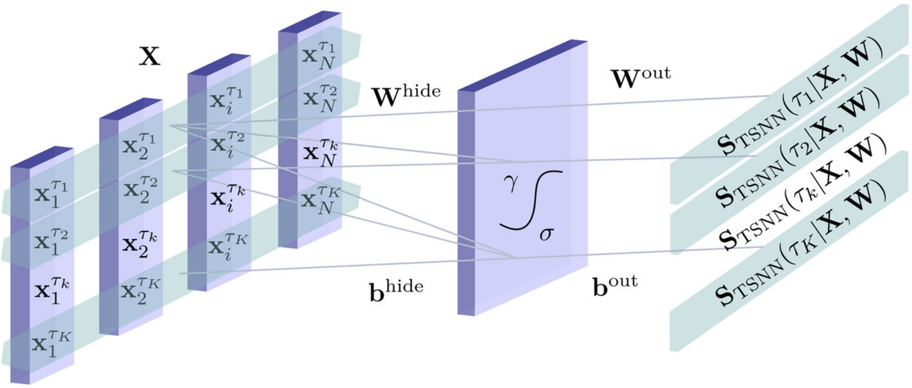


Fig. 2 Time-dependent survival neural network

value and thus $0 < \gamma \leq 1$. Note that such a positive decay ratio indicates that the risk will shrink over time but not vanish. Given the observations for individuals at time t, say $\mathbf{X}^t = (\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_N^t)$, and all historical values observed at time points $j \in \Omega(t)$ before t, we estimate the event risk in an additive manner and compute the TSNN's N outcomes at τ_k, as follows:

$$\mathbf{S}_{\text{TSNN}}(\tau_k|\mathbf{X},\mathbf{W}) = \left(1 + \frac{1}{|\Omega(\tau_k)|} \sum_{j\in\Omega(\tau_k)} \exp\{\xi(j-\tau_k)\} \exp\left\{-\mathbf{w}_k^{\text{out}}\boldsymbol{\phi}(\mathbf{X}^j) - b_k^{\text{out}}\right\}\right)^{-1}$$

$$\boldsymbol{\phi}(\mathbf{X}^j) = \left(1 + \frac{1}{|\Omega(j)|} \sum_{u\in\Omega(j)} \exp\{\xi(u-j)\} \exp\left\{-\mathbf{W}^{\text{hide}}\mathbf{X}^u - \mathbf{b}^{\text{hide}}\right\}\right)^{-1}.$$

As shown in Fig. 2, the survival probability at time τ_k is estimated according to the time-dependent input covariates which are repeatedly measured at time $\tau_1, \tau_2, \ldots, \tau_{k'} (\leq \tau_k)$. This is the reason why we can call this neural network time-dependent.

3.2.2 Recurrent survival neural network

Unlike feed-forward neural networks, recurrent neural networks (RNNs) can use their internal state (memory) to process time-dependent inputs, with the output of a hidden unit at the current timestep being fed back into the hidden unit so that it forms part of the input for the preceding timesteps. This allows the exhibition of temporal dynamic behavior by feeding neural outputs of the activations of the preceding step back into the network. Hence, RNNs are extremely expressive and flexible. Long short-term memory (LSTM) units [18] were developed to overcome the practical issues associated with long-term dependencies in traditional RNNs by learning what information they should keep from the previous time step and what information they should forget.

The schematic of our RSNN includes LSTM units, as shown in Fig. 3. Every unit features an input, three gates (input, forget, and output), and an output activation function. The output of the unit is recurrently connected back to the unit input and all of the gates, given by

$$\mathbf{H}^{\tau_k} = \mathbf{O}^{\tau_k} \odot \tanh(\mathbf{C}^{\tau_k}),$$

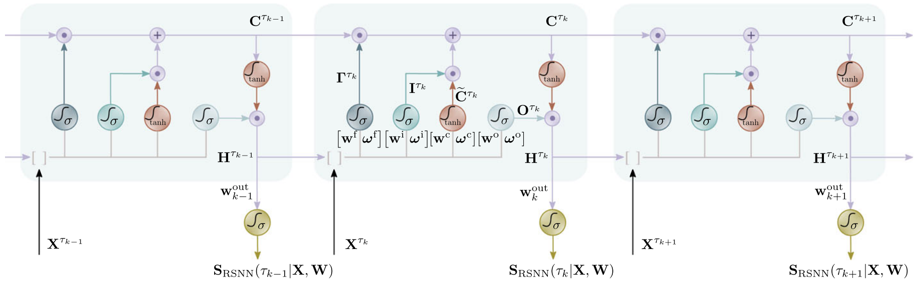

$\mathbf{C}^{\tau_{k-1}}$
tanh
σ
$\mathbf{H}^{\tau_{k-1}}$
$\mathbf{w}^{\text{out}}_{k-1}$
$\mathbf{X}^{\tau_{k-1}}$
$\mathbf{S}_{\text{RSNN}}(\tau_{k-1}|\mathbf{X},\mathbf{W})$
$\mathbf{C}^{\tau_k}$
$\mathbf{\Gamma}^{\tau_k}$
$\mathbf{I}^{\tau_k}$
$\widetilde{\mathbf{C}}^{\tau_k}$
$\mathbf{O}^{\tau_k}$
$[\mathbf{w}^{\text{f}}\ \boldsymbol{\omega}^{\text{f}}]\,[\mathbf{w}^{\text{i}}\ \boldsymbol{\omega}^{\text{i}}]\,[\mathbf{w}^{\text{c}}\ \boldsymbol{\omega}^{\text{c}}]\,[\mathbf{w}^{\text{o}}\ \boldsymbol{\omega}^{\text{o}}]$
$\mathbf{H}^{\tau_k}$
$\mathbf{w}^{\text{out}}_{k}$
$\mathbf{X}^{\tau_k}$
$\mathbf{S}_{\text{RSNN}}(\tau_k|\mathbf{X},\mathbf{W})$
$\mathbf{C}^{\tau_{k+1}}$
$\mathbf{H}^{\tau_{k+1}}$
$\mathbf{w}^{\text{out}}_{k+1}$
$\mathbf{X}^{\tau_{k+1}}$
$\mathbf{S}_{\text{RSNN}}(\tau_{k+1}|\mathbf{X},\mathbf{W})$

Fig. 3 The recurrent survival neural network (RSNN) consists of long short-term memory units

where $\odot$ is a Hadamard (element-wise) product and the output gate $\mathbf{O}^{\tau_k}$ modulates the amount of memory content exposure. Unlike the recurrent unit which simply computes a weighted sum of the input covariates and applies a nonlinear function, each LSTM unit maintains a memory $\mathbf{C}^{\tau_k}$ at time τ_k. This memory cell is updated by partially forgetting the previous memory $\mathbf{C}^{\tau_{k-1}}$ and adding new (candidate) memory content such as

$$\mathbf{C}^{\tau_k} = \mathbf{\Gamma}^{\tau_k} \odot \mathbf{C}^{\tau_{k-1}} + \mathbf{I}^{\tau_k} \odot \widetilde{\mathbf{C}}^{\tau_k}.$$

The input weights in terms of input gate, forget gate, output gate, and cell unit in hidden layer are denoted by $\mathbf{w}^{\text{i}}, \mathbf{w}^{\text{f}}, \mathbf{w}^{\text{o}}, \mathbf{w}^{\text{c}} \in \mathbb{R}^{D\times V}$, respectively, the recurrent weights by $\boldsymbol{\omega}^{\text{i}}, \boldsymbol{\omega}^{\text{f}}, \boldsymbol{\omega}^{\text{o}}, \boldsymbol{\omega}^{\text{c}} \in \mathbb{R}^{D\times D}$, and the bias as $\mathbf{b}^{\text{i}}, \mathbf{b}^{\text{f}}, \mathbf{b}^{\text{o}}, \mathbf{b}^{\text{c}} \in \mathbb{R}^{D}$. The input gate $\mathbf{I}^{\tau_k}$, forget gate $\mathbf{\Gamma}^{\tau_k}$, output gate $\mathbf{O}^{\tau_k}$, and candidate cell $\widetilde{\mathbf{C}}^{\tau_k}$ are computed by

$$\begin{pmatrix} \mathbf{I}^{\tau_k} \\ \mathbf{\Gamma}^{\tau_k} \\ \mathbf{O}^{\tau_k} \\ \widetilde{\mathbf{C}}^{\tau_k} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(\begin{pmatrix} [\mathbf{w}^{\text{i}}\ \boldsymbol{\omega}^{\text{i}}] \\ [\mathbf{w}^{\text{f}}\ \boldsymbol{\omega}^{\text{f}}] \\ [\mathbf{w}^{\text{o}}\ \boldsymbol{\omega}^{\text{o}}] \\ [\mathbf{w}^{\text{c}}\ \boldsymbol{\omega}^{\text{c}}] \end{pmatrix} \begin{bmatrix} \mathbf{X}^{\tau_k} \\ \mathbf{H}^{\tau_{k-1}} \end{bmatrix} + \begin{pmatrix} \mathbf{b}^{\text{i}} \\ \mathbf{b}^{\text{f}} \\ \mathbf{b}^{\text{o}} \\ \mathbf{b}^{\text{c}} \end{pmatrix} \right),$$

where $[\mathbf{X}^{\tau_k}\ \mathbf{H}^{\tau_{k-1}}]^\top$ is the concatenation of the two vectors: input covariates $\mathbf{X}^{\tau_k}$ and $\mathbf{H}^{\tau_{k-1}}$. With input $\mathbf{X}$ at τ_k, the recurrent survival neural network can yield N survival probabilities:

$$\mathbf{S}_{\text{RSNN}}(\tau_k|\mathbf{X}, \mathbf{W}) = \sigma\left(\mathbf{w}_k^{\text{out}}\mathbf{H}^{\tau_k} + b_k^{\text{out}}\right).$$

In particular, the presence of the hidden neurons provides a nonlinear dependence of the outputs on the input covariates. The weights describe the nonlinearity in how the survival probability varies in response to these covariates. Our approach can be thought of as a generalization of multi-task classification, which enables flexible modeling of survival probability in parallel. Each task executes on all training individuals but has an individual covariate input. As was discussed in [27], such multi-task transformation will further reduce the prediction error on each task and hence provide a more accurate estimate than models which aim at modeling the probabilities at once.

3.3 Survival probability evaluation

In the observational world, one needs to know whether event, dropout, or study cutoff comes first when building a model to capture information regarding covariates leading to an event. Survival (i.e., event-free) for individual i means that i is still at risk of the event: that is, the event has not yet occurred. Censoring means that i dropped out of the study or has not experienced the event by the end of the study. Hence, for all (right-)censored samples, the

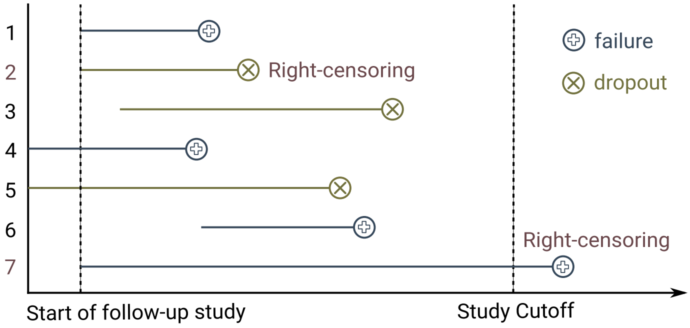


Fig. 4 An example of failures and censored individuals in time-to-event data

unobserved exact survival times are longer than the censoring times. For example, as shown in Fig. 4, the failure event times of the second and seventh individuals are right-censored, since the second dropped out study before the study cutoff and the seventh was still failure-free by the study cutoff. All the others are neither failures nor right-censored due to their unknown start times of follow-up.

We denote the time-to-event response variables by $(\mathcal{T}, \zeta)$, where $\mathcal{T}$ is the observed time, i.e., the minimum of time of event T and time of (right-)censoring C, i.e.,

$$\mathcal{T} = \min\{T, C\} = \begin{cases} T, & \text{if } \zeta = 1 \\ C, & \text{otherwise } (\zeta = 0) \end{cases}$$

The other response variable $\zeta = \mathbb{1}\{T \leq C\}$ equals 1 if the event happened and 0 otherwise. That is, $\zeta \in \{0, 1\}$ indicates either censorship or event occurrence. We shall assume without loss of generality that the individuals are ascendingly sorted according to observation, and T and C are conditionally independent given covariates.

3.3.1 Survival process

Given N_{tr} training individuals, the actual survival process for individual i can be represented as $\varepsilon_i(\tau_1)\ \varepsilon_i(\tau_2)\ \cdots\ \varepsilon_i(\tau_K)$. Each survival status $\varepsilon_i(\tau_k)$ indicates whether or not the event occurs by time τ_k, taking a value of 1 up to τ_k, 0 thereafter, and -1 for unknown cases. Once $\varepsilon_i(t)$ becomes "0," it will not turnover to "1." There are thus $K + 1$ legally possible sequences of the form $(1, 1, \ldots, 0, 0, \ldots)$, including the sequences composed of all "1"s and all "0"s. Supposing $\mathcal{K}_i^{\epsilon} = \{k : \varepsilon_i(\tau_k) = \epsilon_i\}$, the observed statuses are greater than or equal to unknown statues if the failure is (right-)censored, i.e., $\epsilon_i(\tau_k) \geq \varepsilon_i(\tau_{k'})$, $\forall k \in \mathcal{K}_i^1$ and $\forall k' \in \mathcal{K}_i^{-1}$. For an uncensored case, the survival statuses during the follow-up period are strictly greater than those after the event, i.e., $\varepsilon_i(\tau_k) > \varepsilon_i(\tau_{k'})$, $\forall k \in \mathcal{K}_i^1$ and $\forall k' \in \mathcal{K}_i^0$.

Table 2 shows an example of a survival process. For the uncensored individuals 1 and 3, the cells of the corresponding rows are labeled as "1" until the scale time and "0" for the remaining cells; for the censored individuals 2 and 4, the cells are labeled as "1" until the censoring time and "– 1" thereafter the remaining cells.

3.3.2 Censoring Kullback–Leibler divergence

Survival networks cannot be effective prediction models unless they achieve the objective that *the predicted event-free probabilities approach the actual survival process*. In order

Table 2 An example of generating survival process from time-to-event (TTE) data

i	TTE		Survival process (time in month)								
	$\mathcal{T}_i$	ζ_i	$\varepsilon_i(1)$	$\varepsilon_i(2)$	$\varepsilon_i(3)$	$\varepsilon_i(4)$	$\varepsilon_i(5)$	$\varepsilon_i(6)$	$\varepsilon_i(7)$	$\varepsilon_i(8)$	$\varepsilon_i(9)$
1	2	1	1	1	0	0	0	0	0	0	0
2	6	1	1	1	1	1	1	1	− 1	− 1	− 1
3	4	1	1	1	1	1	0	0	0	0	0
4	9	0	1	1	1	1	1	1	1	1	1

to quantify such approachability, for every individual i we define the censoring Kullback–Leibler (KL) divergence (an alternative to the relative error [36]) between the distributions of i's survival probability $S_i(\tau_k|\mathbf{X},\mathbf{W})$ and survival status $\varepsilon_i(\tau_k) \in \{0, 1\}$, as follows:

$$\mathcal{D}_i(\tau_k,\mathbf{W}) = \varepsilon_i(\tau_k)\ln\frac{\varepsilon_i(\tau_k)}{S_i(\tau_k|\mathbf{X},\mathbf{W})} + (1-\varepsilon_i(\tau_k))\ln\frac{1-\varepsilon_i(\tau_k)}{1-S_i(\tau_k|\mathbf{X},\mathbf{W})}. \qquad (2)$$

The optimal weights make $S_i(\tau_k|\mathbf{X},\mathbf{W})$ as close as possible to 1 if i remains failure-free by τ_k, and to 0 otherwise, while outputs of 1 and 0 are definitely true and definitely false predictions, respectively. Our learning criterion, then, is to minimize $\mathcal{D}_i(\tau_k,\mathbf{W})$ over time snapshots of $\mathcal{K}_i^0$ and $\mathcal{K}_i^1$ at which survival statuses are known, for all N_{tr} training individuals.

3.4 Survival learning

3.4.1 Learning objective

It is worth mentioning the known fact that S_k descends from 1 to 0, as time goes by, from the beginning to the end of life. Hence, the minimization should be constrained by this monotonicity. The proven penalty method converts the constrained optimization problem into a series of unconstrained optimization problems. Accordingly, we utilize the static penalty [33] incurred for violating the inequality constraints and minimize the average error computed by

$$\begin{aligned} E(\mathbf{W}^*) &= \min_{\mathbf{W}} \quad \frac{1}{N_{\text{tr}}}\sum_{i=1}^{N_{\text{tr}}}\left(\frac{1}{\left|\mathcal{K}_i^0 \bigcup \mathcal{K}_i^1\right|}\sum_{k\in\mathcal{K}_i^0\cup\mathcal{K}_i^1}\mathcal{D}_i(\tau_k,\mathbf{W})\right) \\ &\text{s. t.} \quad S_i(\tau_k|\mathbf{X},\mathbf{W}) - S_i(\tau_{k+1}|\mathbf{X},\mathbf{W}) > 0 \forall k = 1, 2\ldots, K-1,\ i = 1, 2, \ldots, N_{\text{tr}}. \end{aligned}$$

The penalty method, which converts a given constrained minimization problem into a series of unconstrained optimization problems, turns out to be an effective approach. The solution of the unconstrained optimization problem converges to the solution of the original problem. Accordingly, we benefit from the advantages of the exterior-point method (EPM) [48] and establish the following objective function which is incurred for violating the constraints on survival probabilities:

$$\mathcal{E}(\mathbf{W}^*) = \min_{\mathbf{W}} \quad E(\mathbf{W}) + \frac{1}{N_{\text{tr}}(K-1)}\sum_{i=1}^{N_{\text{tr}}}\sum_{k=1}^{K-1}(\max\{0, S_i(\tau_k|\mathbf{X},\mathbf{W}) - S_i(\tau_{k+1}|\mathbf{X},\mathbf{W})\})^2$$

3.4.2 Training

We train the TSNN using the forward-only Levenberg–Marquardt algorithm presented in [46], which inherits the speed advantage of the Gauss–Newton algorithm and the stability of the steepest descent method. The Levenberg–Marquardt algorithm blends these two approaches by updating the weights ($\mathbf{w}^{\text{hide}}$ and $\mathbf{w}^{\text{out}}$) $\in \mathbb{R}^{(V+D)}$ at the $(m+1)$th iteration according to the weights at the mth iteration, i.e., $\mathbf{w}^{(m+1)} = \mathbf{w}^{(m)} - (\mathbf{J} \cdot \mathbf{J} + \mu\mathbf{I})^{-1}\mathbf{Je}$, where $\mathbf{I} \in \mathbb{R}^{(V+D)}$ is the identity matrix and $\mathbf{J} \in \mathbb{R}^{(N_{\text{tr}} \times K) \times (V+D)}$ stands for the Jacobian matrix that can be calculated according to the method introduced by Wilamowski and Yu [46]. Each element e_{ik} of $\mathbf{e} \in \mathbb{R}^{N_{\text{tr}} \times K}$ is the error at τ_k with input $\mathbf{x}_i$. When the positive damping parameter μ is very small (near zero), the Levenberg–Marquardt is replaced by the Gauss–Newton algorithm. Conversely, when μ becomes very large, the steepest descent method is used to obtain the update $\mathbf{w}^{(m+1)} = \mathbf{w}^{(m)} - \frac{1}{\mu} \frac{\partial \mathcal{E}}{\partial \mathbf{w}^{(m)}}$. The Levenberg–Marquardt performs a training process for the kth output at τ_k, as follows: first, we initialize $\mathcal{E}$ with the initial weights $\mathbf{w}^{(0)}$ (randomly generated). Then, the loop starts, by updating the weights and $\mathcal{E}$. If the current $\mathcal{E}$ is increased as a result of the update, the weights are reset to the previous values and μ is increased by a factor of 10, after which the weights are updated again. If the current $\mathcal{E}$ is decreased, the new weights are retained as the current ones and μ is decreased by a factor of 10. The algorithm repeats this updating loop until $\mathcal{E}$ no longer changes. For training the RSNN, the back-propagation through time (BPTT) algorithm [35] is used. In each training iteration, the model updates its parameters according to the KL divergence. It attempts to reach the optimal parameter using the mini-batch gradient descent until it achieves convergence, with the batch size set to 32.

4 Experiments

4.1 Data and pre-processing

Four time-to-event datasets were drawn from the Prognostics Data Repository provided by the PCoE at NASA Ames; the Surveillance, Epidemiology, and End Results (SEER) statistics database; and the Canadian Community Health Survey (CCHS) statistical surveys.

- In the Engine dataset, 388 engines' cycles were considered unobserved, for a 27.4% censoring rate. The objective was to predict the number of operational cycles remaining until compressor and fan degradation.
- The Battery dataset comprises the first 20 cells from the PDR's Randomized Battery Usage dataset, each with 42 galvanostatic voltage curves. Failure was censored for 45.8% of batteries, and 10 covariates were extracted from the time series of temperature and current (mA) every 30 s.
- For the Cancer dataset (drawn from the SEER Breast Cancer data), survival times were computed by subtracting the date of diagnosis from the date of last contact (the study cutoff).
- The Aging dataset contains data on healthy aging acquired directly between December 2008 and November 2009 from respondents in a survey, which focused on the health of Canadians aged 45 and over, examining the various factors that impact healthy aging. A total of 7611 valid interviews covering the population living in the ten provinces were used.

Table 3 Statistics of the four time-to-event datasets

Dataset	Size	Dimensionality	Censoring (%)	Missing value (%)	Failure event of interest
Engine	1416	21	27.4	11.3	Compressor and fan degradation
Battery	842	10	45.8	5.9	30% Fade in rated battery capacity
Cancer	3390	18	19.3	15.7	Breast cancer caused death
Aging	7611	35	34.5	26.2	Retirement and disability

Table 3 summarizes the statistics, including data size (number of individuals or participants), dimensionality (number of covariates), censoring rate, missing-value percentage, and failure event of interest. Categorical covariates were transformed into numerical values by means of the probabilistic frequency estimator presented in [8]. Afterward, missing values were filled in via a linear regression provided by [23]. In order to reduce data redundancy and improve data integrity, all values were normalized.

4.2 Competitors

We compared the proposed survival networks with several state-of-the-art methods.

- Cox [10] extends the Cox model to time-dependent covariates and has a survival function $S(t) = S_{\text{base}} \exp(\boldsymbol{\beta}\mathbf{x}^t)$ with the baseline probability S_{base} when $\mathbf{x}^t = (0, 0, \ldots, 0)$ and the regression coefficients $\boldsymbol{\beta}$ describing how the survival probability responds to the covariates.
- CoxAvg [51] uses the average of repeated measures $\bar{\mathbf{X}}$ for every covariate, that is, $S(t) = S_{\text{base}} \exp(\boldsymbol{\beta}\bar{\mathbf{X}}^t)$.
- CoxNN [13] replaces the linear exponent of the Cox hazard by a nonlinear artificial neural networks output. The survival probability becomes $S(t) = S_{\text{base}} \exp(\phi(\mathbf{x}^t))$, where the $\phi(\mathbf{x}^t)$ is the outcome of an artificial neural network.
- RNN-SURV [16] computes survival probability in a given interval of time by a recurrent neural network (with LSTM cells). We set this network to have 2 feed-forward layers and 2 recurrent layers and used the cross-entropy function as the loss function.
- AFT [45] generates survival probability by $S(t) = \exp(-(t/\exp(\boldsymbol{\beta}\mathbf{x}^t))^{\frac{1}{\eta}})$, where we assume the survival time T has a Weibull distribution, that is, we have $T \sim W(\exp(\boldsymbol{\beta}\mathbf{x}^t), \frac{1}{\eta})$.
- EN-BJ [6] extends the least squares estimator to the semi-parametric linear regression model in which the error distribution is completely unspecified.
- MTLR [31] models survival probabilities for individuals with event and for censored individuals. The logistic regression coefficients are time-varying.
- RSF [19] estimates conditional cumulative failure hazard by aggregating tree-based Nelson–Aalen estimators.

We also studied SNN which does not estimate the latent risk but instead predicts the output probabilities using Eq. 1 with the time-dependent input $\mathbf{x}^t$. To investigate the significance of accounting for censoring in neural networks, we use a Kaplan–Meier (KM) estimator [21] to fill in the survival probabilities for censored individuals in SNN, TSNN, and RSNN. The KM estimator for the survival probability at the specified survival time is a product of the same estimate up to the previous time and the observed survival rate for the specified time, given as

$$S(t) = \prod_{\forall \tau_k < t} \left(1 - \frac{|\{i | T_i = \tau_k\}|}{|\{i | T_i \geq \tau_k\}|}\right).$$

For our real-time survival networks, we set the hidden layer to $D = 4$ neurons. An output layer with $K = 20$ was used in analyses of the Engine and Battery datasets, and $K = 12$ for the Cancer and Aging datasets. The decay coefficient $\xi = 1.5$ was used in TSNN and KM-TSNN.

4.3 Evaluation metrics

Performance on the N_{te} test individuals $\mathbf{X}_{\text{te}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{\text{te}}})$ was evaluated in terms of three independent metrics: the failure-event AUC (FAUC), the concordance index (C-index), and the censoring Brier score (CBS), redefined as follows ($\mathbb{1}$ is the indicator function)

- FAUC provides a probability measure of classification ability at a pre-specified time snapshot (e.g., at τ_K in our case). It quantifies the model's ability to address the issue *Is i likely to remain event-free by time t?*.

$$\text{FAUC} = \frac{\sum_{i:\epsilon_i(\tau_K)=0} \sum_{j:\epsilon_j(\tau_K)=1} \mathbb{1}\left\{S_i(\tau_K | \mathbf{X}_{\text{te}}, \mathbf{W}^*) < S_j(\tau_K | \mathbf{X}_{\text{te}}, \mathbf{W}^*)\right\}}{\left|\{i : \epsilon_i(\tau_K) = 0\}\right| \times \left|\{j : \epsilon_j(\tau_K) = 1\}\right|}.$$

- C-index serves as a generalization of the FAUC, giving an estimate of how accurately the model can answer the question *Which of i and j is more likely to remain event-free?*.

$$\text{C-index} = \frac{\sum\limits_{i:\epsilon_i(\tau_K)=0} \sum\limits_{j:T_i<T_j} \mathbb{1}\left\{S_i\left(\tau_{\min\{\mathcal{K}_i^0\}} | \mathbf{X}_{\text{te}}, \mathbf{W}^*\right) < S_j\left(\tau_{\min\{\mathcal{K}_i^0\}} | \mathbf{X}_{\text{te}}, \mathbf{W}^*\right)\right\}}{\left|\{i : \epsilon_i(\tau_K) = 0\}\right| \times \left|\{j : T_i < T_j\}\right|}.$$

- CBS measures an ensemble prediction error across the test data, i.e., the power of a model to address the question *How accurate is the prediction that i will remain event-free?*.

$$\text{CBS} = \frac{1}{N_{\text{te}}} \sum_{i=1}^{N_{\text{te}}} \left(1 - \varepsilon_i(\tau_K) - S_i\left(\tau_K | \mathbf{X}_{\text{te}}, \mathbf{W}^*\right)\right)^2.$$

In some scenarios, one might be more concerned about the difference between the predicted survival time $\mathcal{T}_i$ and the true survival time T_i. For example, as the cost of hospital stays and medication scales linearly with the survival time, the error in the survival time could be relevant, defined as $\text{Æ}(\mathcal{T}_i, T_i) = \mathcal{T}_i - T_i$. Given any of the survival error estimates, we can make a point prediction of survival time for individual i using the survival probability S estimated by the models: e.g., for TSNN, we have

$$\mathcal{T}_i^* = \underset{\mathcal{T}_i \in \{\tau_1, \tau_2, \ldots, \tau_K\}}{\arg\min} \sum_{k=1}^{K} \sigma\left(\frac{\tau_K - \tau_1}{2} + \text{Æ}(\mathcal{T}_i, \tau_k)\right) S_i\left(\tau_k | \mathbf{X}_{\text{te}}, \mathbf{W}^*\right)$$

4.4 Results and discussion

From the tenfold cross-validation results on the test data, shown in Table 4, it is evident that TSNN outperforms all the other models except MTLR on the Cancer dataset when FAUC is measured. Either RSNN or SNN performs second-best, with the sole exception of FAUC on the Cancer dataset. TSNN and RSNN achieve average C-index improvements

Table 4 Comparison of the tenfold cross-validation FAUC, C-index, and CBS results on the test data, in the form of mean ± standard deviation

	FAUC	C-index	CBS	FAUC	C-index	CBS
	Engine			*Battery*		
TSNN	**.744** ± .017	**.753** ± .028	**.163** ± .018	**.810** ± .022	**.761** ± .014	**.212** ± .029
KM-TSNN	.731 ± .024	.678 ± .040	.248 ± .011	.695 ± .032	.733 ± .015	.261 ± .034
RSNN	.741 ± .013	.726 ± .018	.188 ± .014	.775 ± .020	.748 ± .023	.226 ± .015
KM-RSNN	.696 ± .017	.682 ± .011	.196 ± .029	.703 ± .026	.691 ± .027	.243 ± .028
SNN	.719 ± .038	.724 ± .026	.185 ± .023	.769 ± .015	.710 ± .029	.229 ± .038
KM-SNN	.676 ± .022	.639 ± .029	.283 ± .016	.674 ± .021	.656 ± .019	.255 ± .018
CoxNN	.686 ± .036	.613 ± .028	.404 ± .025	.664 ± .049	.718 ± .013	.332 ± .026
RNN-SURV	.623 ± .025	.529 ± .014	.336 ± .030	.582 ± .013	.475 ± .041	.373 ± .027
Cox	.694 ± .031	.587 ± .029	.276 ± .018	.651 ± .019	.629 ± .022	.301 ± .048
CoxAvg	.707 ± .025	.633 ± .026	.284 ± .021	.624 ± .017	.535 ± .023	.287 ± .032
AFT	.682 ± .014	.636 ± .053	.241 ± .042	.625 ± .030	.674 ± .020	.274 ± .022
EN-BJ	.736 ± .029	.688 ± .015	.339 ± .012	.718 ± .024	.654 ± .034	.237 ± .013
MTLR	.708 ± .051	.683 ± .023	.215 ± .043	.726 ± .020	.670 ± .015	.364 ± .019
RSF	.695 ± .019	.675 ± .031	.268 ± .031	.578 ± .029	.520 ± .041	.286 ± .031
	Cancer			*Aging*		
TSNN	.794 ± .013	**.782** ± .029	**.186** ± .017	**.787** ± .028	**.765** ± .031	**.151** ± .019
KM-TSNN	.694 ± .041	.681 ± .024	.226 ± .047	.730 ± .016	.736 ± .022	.166 ± .027
RSNN	.771 ± .024	.733 ± .035	.239 ± .022	.744 ± .023	.749 ± .028	.253 ± .021
KM-RSNN	.682 ± .015	.677 ± .027	.273 ± .025	.711 ± .012	.647 ± .033	.331 ± .028
SNN	.785 ± .034	.756 ± .017	.217 ± .008	.706 ± .020	.722 ± .018	.221 ± .015
KM-SNN	.663 ± .032	.639 ± .018	.322 ± .014	.707 ± .010	.645 ± .029	.224 ± .011
CoxNN	.733 ± .038	.674 ± .019	.235 ± .034	.721 ± .022	.717 ± .016	.301 ± .032
RNN-SURV	.572 ± .018	.540 ± .023	.301 ± .014	.568 ± .039	.521 ± .028	.274 ± .021
Cox	.699 ± .037	.620 ± .017	.263 ± .033	.572 ± .031	.553 ± .024	.236 ± .019
CoxAvg	.625 ± .027	.593 ± .022	.277 ± .031	.635 ± .021	.605 ± .017	.352 ± .023
AFT	.689 ± .034	.564 ± .028	.263 ± .036	.707 ± .037	.660 ± .024	.305 ± .026
EN-BJ	.767 ± .023	.745 ± .033	.279 ± .014	.742 ± .044	.720 ± .022	.235 ± .018
MTLR	**.818** ± .022	.739 ± .025	.243 ± .017	.716 ± .017	.734 ± .026	.324 ± .030
RSF	.732 ± .017	.673 ± .037	.272 ± .053	.722 ± .035	.684 ± .025	.336 ± .027

The best results are in bold, and the second-best performances are underlined

of 11% and 9%, respectively, over the prior state of the art. The superior performance of the survival networks (TSNN, RSNN, and SNN) relative to the KM-based alternatives (KM-TSNN, KM-RSNN, and KM-SNN) reveals that our survival learning approach to minimizing the censoring KL divergence we defined in Eq. 2 can effectively cope with censored data, compared with the conventional survival statistical estimator. Comparing TSNN with SNN and KM-TSNN with KM-SNN, we find that TSNN and KM-TSNN perform much better. This demonstrates the significance and effectiveness of estimating the latent failure risk. It can be seen from the high FAUC and C-index values achieved by RSNN that LSTM can be an effective approach for dealing with time-dependent covariates in longitudinal data, although RSNN may not generate a CBS as low as that of TSNN or SNN. CoxNN yields

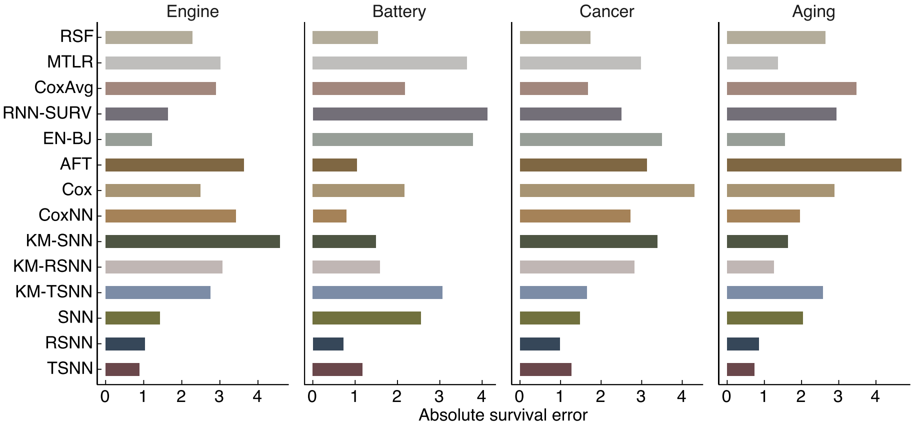


Fig. 5 Comparison on average absolute survival error |Æ| over all individuals having a failure event

even lower accuracies in comparison with Cox and CoxAvg, demonstrating that use of the risk nonlinearity property alone does not enhance the Cox model [22]. Although CoxAvg takes the historical data into account, it does not perform better than the Cox model, mainly because using a single mean value as the representative of all the repeated measures cannot recognize the underlying relations between time-dependent covariates and the survival probability and therefore would not enhance the model's predictive ability. As a recurrent neural network, RNN-SURV achieves much lower prediction accuracies in terms of FAUC and C-index in comparison with RSNN. Note that the real-time survival networks take into account potential relationships between the classifications and therefore achieve a significant performance gain over the regression method MTLR, which performs each prediction task independently [31]. Note also the extremely low CBS achieved by TSNN on the four datasets, indicating high accuracy in predicting the absolute survival probability and high confidence in forecasting failure.

An important indicator of the effectiveness of the censoring-KL-divergence-based survival learning approach is whether it enables our real-time survival networks to recommend the right moment for preventive intervention in the form of maintenance or treatment. To investigate this, we compare the average absolute survival error (|Æ|) of predicting the survival times for event individuals. As can be seen from Fig. 5, TSNN and RSNN achieve a very low error less than τ, while the competing models output an over 2τ deviation in predicting survival time. This in turn demonstrates that our survival learning approaches are able to provide accurate prognostic information about the time of failure events, especially in the long-term longitudinal aging study.

We compare the survival probability curves yielded by competing models by means of a case study on the Engine dataset. All of the engines that failed at any time were divided into 6 groups according to their times to failure. In each sub-figure of Fig. 6, we plotted a survival curve according to the average failure-free survival probability predicted by each model on the corresponding group of engine failures. It can be seen from the respective gray areas that either TSNN or RSNN (plotted by the dashed curve) yields a significantly lower average probability over all data (i.e., all engine failures) in comparison with other models, mainly because latent risk estimation can help in amending the relationship between latent risk and failure-free survival probability. This means that, using our real-time survival networks, the

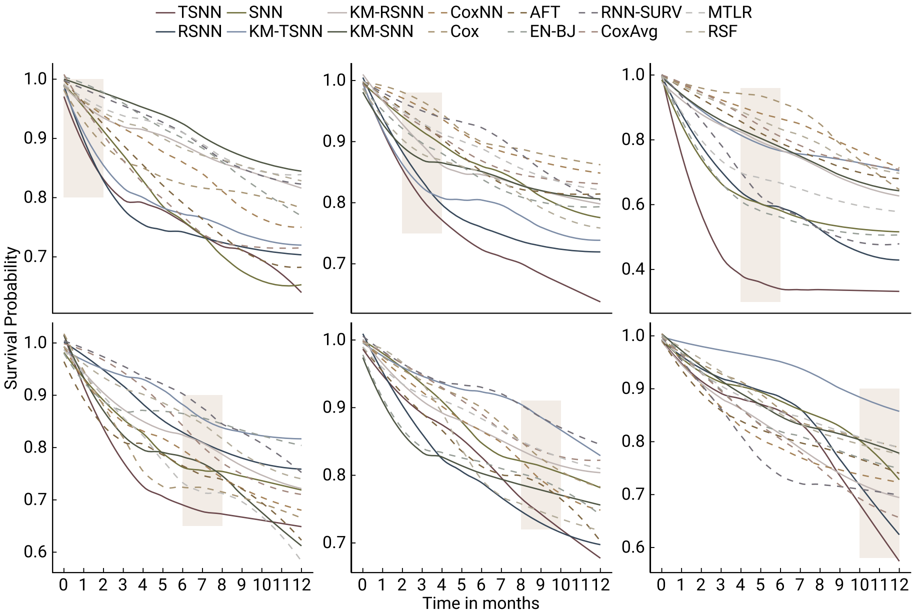


Fig. 6 Change in predicted survival probability curve for engines. The 6 sub-figures are plotted for the engines that failed, at 2-month intervals (see individual rectangles) from the 1st month to the 12th month. Each curve in the sub-figures is the average predicted probability for a group of engines

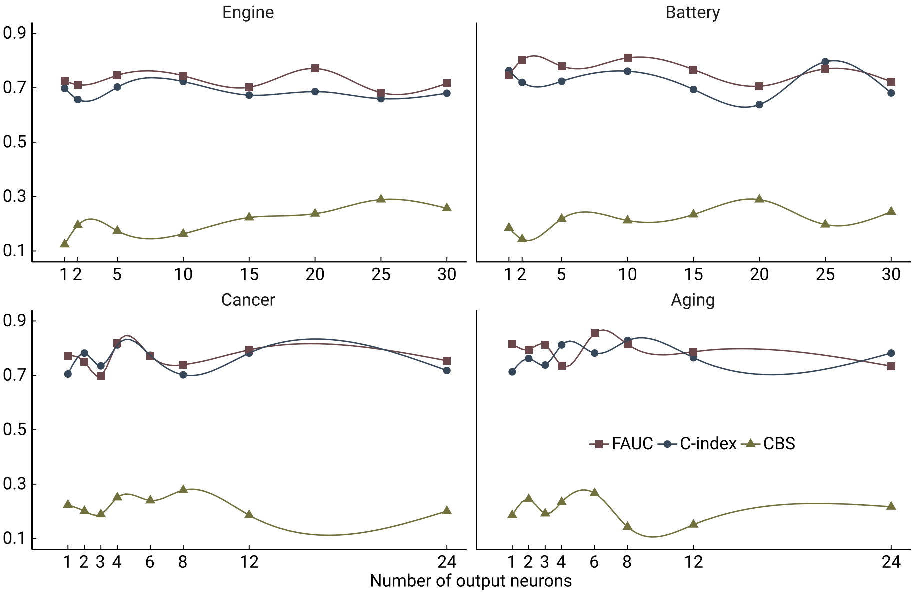


Fig. 7 Change in TSNN performance with varying K

equipment crew could be issued a warning much earlier than in the other models, and offered advice on maintenance intervention in time to stave off potential failure.

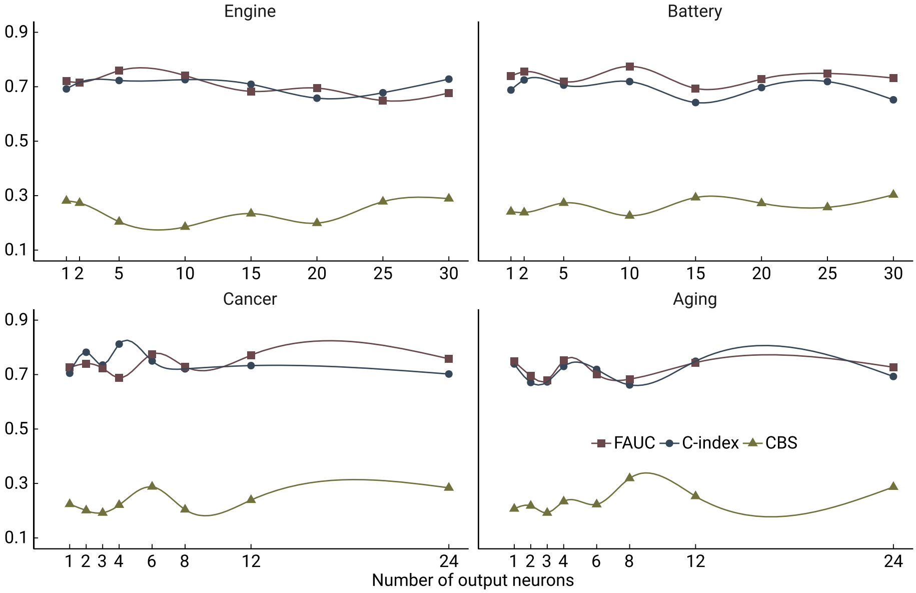


Fig. 8 Change in RSNN performance with varying K

To gain a deeper insight into the functionality of TSNN and RSNN, we set a varying K value of 1, 2, 5, 10, 15, 20, 25, and 30 when they run on Engine (in the 300-cycle follow-up) and Battery (in the 300,000s observational period), and of 1, 2, 3, 4, 6, 8, 12, and 24 when they run on the Cancer and Aging datasets (in the 2-year study period), with the output time interval becoming 24, 12, 8, 6, 4, 3, 2, and 1 month(s), respectively. (Please keep in mind that K is a user-defined value and the time interval is not required to be equal.) The FAUC, C-index, and CBS results shown in Fig. 7 change by less than 4%, 6% and 13% on Engine; 9%, 13%, and 14% on Battery; 12%, 11%, and 9% on Cancer; and 8%, 11%, and 9% on Aging, respectively. In Fig. 8, we can see that RSNN's FAUC, C-index, and CBS results change by less than 7%, 6%, and 10% on Engine; 6%, 8%, and 8% on Battery; 9%, 10%, and 9% on Cancer; and 7%, 7%, and 12% on Aging, respectively. This demonstrates that users can count on TSNN and RSNN as reliable, as they will not fluctuate enormously with change in the output layer of neural networks. Figure 9 shows the average results (for the four datasets) yielded by TSNN with a varying decay ratio ξ, which might lead to an inaccurate risk estimate and therefore a poor predictive ability when ξ becomes extremely large or small. It can be seen clearly that TSNN achieves high FAUC and C-index results and maintains a low CBS when the ratio takes a value in the range [1,2].

5 Conclusion

In this paper, we have provided two dynamic-data-driven survival neural network models for time-to-event prediction in longitudinal studies. TSNN performs an additive latent failure risk estimation and multiple binary classifications for predicting survival probabilities. RSNN employs a network of long short term memory units to analyze time-dependent covariates and generates the survival probabilities within multiple time intervals. The new survival learning

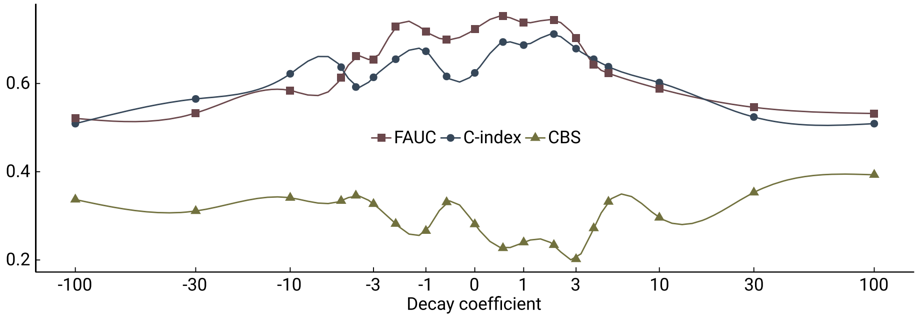


Fig. 9 Change in TSNN performance (average on the four datasets) with varying decay ratio ξ

approach optimizes the neural networks by minimizing the censoring KL divergence between the resulting probabilities and the actual survival process. In addition, the learning criterion constrains the survival probability to decrease as time elapses. The AUC, C-index, Brier score, and survival error (based on survival time estimate) are redefined as the evaluation metrics. Experimental results on four time-to-event datasets confirm that our models outperform several state-of-the-art models and are therefore good candidates for developing a decision-making assistance system to help with early prediction and preventive intervention in long-term follow-up studies.

Acknowledgements This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant 396097-2015, the National Natural Science Foundation of China (NSFC) under Grants U1805263 and 61672157, the Canadian Institutes of Health Research (CIHR) under Grant 391051, the Fonds de Recherche du Québec-Santé, and the Département de médecine de famille et de médecine d'urgence at the Université de Sherbrooke. Part of this work was done while Jianfei Zhang was doing research in CWRU with Yanfang Ye. Jianfei Zhang and Yanfang Ye's work is partially supported by the National Science Foundation (NSF) under Grants IIS-1951504, CNS-1940859, CNS-1946327, CNS-1814825, and OAC-1940855, the Department of Justice/ National Institute of Justice (DoJ/NIJ) under Grant NIJ 2018-75-CX-0032, and the Institute for Smart, Secure and Connected Systems (ISSACS) at CWRU and Cleveland Foundation under Grant 292767.

References

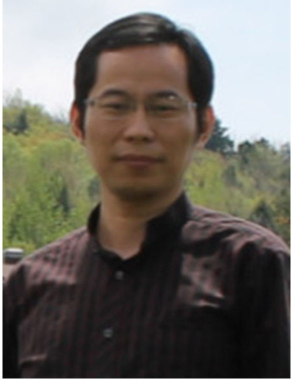
1. Aalen O (1978) Nonparametric estimation of partial transition probabilities in multiple decrement models. Ann Stat 6:534–545
2. Alaa AM, van der Schaar M (2017) Deep multi-task Gaussian processes for survival analysis with competing risks. In: Proceedings of the annual conference on neural information processing systems (NIPS), pp 2326–2334
3. Bellot A, van der Schaar M (2018) Multitask boosting for survival analysis with competing risks. In: Proceedings of the annual conference on neural information processing systems (NIPS), pp 1397–1406
4. Biganzoli E, Boracchi P, Mariani L, Marubini E (1998) Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med 17(10):1169–1186
5. Binder H, Schumacher M (2008) Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. BMC Bioinform 9(1):14
6. Buckley J, James I (1979) Linear regression with censored data. Biometrika 66(3):429–436
7. Caruana EJ, Roman M, Hernández-Sánchez J, Solli P (2015) Longitudinal studies. J Thorac Dis 7(11):E537
8. Chen L, Wang S (2013) Central clustering of categorical data with automated feature weighting. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 1260–1266
9. Chen Q, May RC, Ibrahim JG, Chu H, Cole SR (2014) Joint modeling of longitudinal and survival data with missing and left-censored time-varying covariates. Stat Med 33(26):4560–4576

10. Cox DR (1972) Regression models and life tables. J R Stat Soc Ser B (Stat Methodol) 34:187–220
11. Cutler SJ, Ederer F (1958) Maximum utilization of the life table method in analyzing survival. J Chronic Dis 8(6):699
12. Fan J, Zhang W (2008) Statistical methods with varying coefficient models. Stat Interface 1(1):179
13. Faraggi D, Simon R (1995) A neural network model for survival data. Stat Med 14(1):73–82
14. Fernández T, Rivera N, Teh YW (2016) Gaussian processes for survival analysis. In: Proceedings of the annual conference on neural information processing systems (NIPS), pp 5021–5029
15. Fisher LD, Lin DY (1999) Time-dependent covariates in the Cox proportional-hazards regression model. Annu Rev Public Health 20(1):145–157
16. Giunchiglia E, Nemchenko A, van der Schaar M (2018) RNN-SURV: a deep recurrent model for survival analysis. In: International conference on artificial neural networks (ICANN), pp 23–32. Springer, Berlin
17. Grob GL, Cardoso Â, Liu CB, Little DA, Chamberlain BP (2018) A recurrent neural network survival model: predicting web user return time. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), pp 152–168. Springer, Berlin
18. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
19. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS (2008) Random survival forests. Ann Appl Stat 2:841–860
20. Jenkins SP (2005) Survival analysis. Unpublished Manuscript, Institute for Social and Economic Research, Chapter 3, University of Essex, Colchester, UK
21. Kaplan EL, Meier P (1958) Nonparametric estimation from incomplete observations. J Am Stat Assoc (JASA) 53(282):457–481
22. Katzman J, Shaham U, Bates J, Cloninger A, Jiang T, Kluger Y (2018) DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Med Res Methodol 18:24
23. Kim H, Golub GH, Park H (2004) Imputation of missing values in DNA microarray gene expression data. In: CSB, pp 572–573
24. Kim M, Pavlovic V (2018) Variational inference for Gaussian process models for survival analysis. In: Proceedings of the annual conference on uncertainty in artificial intelligence (UAI), pp 435–445
25. Lee C, Zame WR, Yoon J, van der Schaar M (2018) Deephit: a deep learning approach to survival analysis with competing risks. In: Proceedings of the AAAI national conference on artificial intelligence (AAAI), pp 2314–2321
26. Li H, Ge Y, Zhu H, Xiong H, Zhao H (2017a) Prospecting the career development of talents: a survival analysis perspective. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 917–925
27. Li Y, Wang J, Ye J, Reddy CK (2016a) A multi-task learning formulation for survival analysis. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 1715–1724
28. Li Y, Wang L, Wang J, Ye J, Reddy CK (2017b) Transfer learning for survival analysis via efficient l2,1-norm regularized Cox regression. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 231–240
29. Li Y, Xu KS, Reddy CK (2016b) Regularized parametric regression for high-dimensional survival analysis. In: Proceedings of the SIAM international conference on data mining (SDM), pp 765–773
30. Liestbl K, Andersen PK, Andersen U (1994) Survival analysis and neural nets. Stat Med 13(12):1189–1200
31. Lin H-C, Baracos V, Greiner R, Chun-nam JY (2011) Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In: Proceedings of the annual conference on neural information processing systems (NIPS), pp 1845–1853
32. Liu M, Lu W, Shore RE, Zeleniuch-Jacquotte A (2010) Cox regression model with time-varying coefficients in nested case–control studies. Biostatistics 11(4):693–706
33. Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. Evolut Comput 4(1):1–32
34. Moghaddass R, Rudin C (2014) The latent state hazard model, with application to wind turbine reliability. Ann Appl Stat 9(4):1823–1863
35. Rumelhart DE, Hinton GE, Williams RJ (1988) Neurocomputing: foundations of research. Chapter Learning representations by back-propagating errors. MIT Press, Cambridge, pp 696–699
36. Street WN (1998) A neural network model for prognostic prediction. In: Proceedings of the annual international conference on machine learning (ICML), pp 540–546
37. Sun Y, Sundaram R, Zhao Y (2009) Empirical likelihood inference for the Cox model with time-dependent coefficients via local partial likelihood. Scand J Stat 36(3):444–462

38. Thomas L, Reyes EM (2014) Tutorial: survival estimation for Cox regression models with time-varying coefficients using SAS and R. J Stat Softw 61(c1):1–23
39. Tian L, Zucker D, Wei L (2005) On the Cox model with time-varying regression coefficients. J Am Stat Assoc (JASA) 100(469):172–183
40. Tibshirani R et al (1997) The LASSO method for variable selection in the Cox model. Stat Med 16(4):385–395
41. Vinzamuri B, Li Y, Reddy CK (2014) Active learning based survival regression for censored data. In: Proceedings of the ACM international conference on information and knowledge management (CIKM), pp 241–250
42. Vinzamuri B, Reddy CK (2013) Cox regression with correlation based regularization for electronic health records. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 757–766
43. Wang L, Li Y, Zhou J, Zhu D, Ye J (2017) Multi-task survival analysis. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 485–494
44. Wang W (2004) Proportional hazards regression models with unknown link function and time-dependent covariates. Stat Sin 14(3):885–906
45. Wei L-J (1992) The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. Stat Med 11(14–15):1871–1879
46. Wilamowski BM, Yu H (2010) Neural network learning without backpropagation. IEEE Trans Neural Netw (TNN) 21(11):1793–1803
47. Wu Y, Yuan M, Dong S, Lin L, Liu Y (2018) Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 275:167–179
48. Yamashita H, Tanabe T (2010) A primal-dual exterior point method for nonlinear optimization. SIAM J Optim (SIOPT) 20(6):3335–3363
49. Yang G, Cai Y, Reddy CK (2018) Spatio-temporal check-in time prediction with recurrent neural network based survival analysis. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 2976–2983
50. Yu S, Fung G, Rosales R, Krishnan S, Rao RB, Dehing-Oberije C, Lambin P (2008) Privacy-preserving Cox regression for survival analysis. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 1034–1042
51. Zhang D (2008) Analysis of survival data (chapter 10: time dependent covariates). https://www.coursehero.com/file/18367916/chap10/
52. Zhang J, Chen L, Vanasse A, Courteau J, Wang S (2016a) Survival prediction by an integrated learning criterion on intermittently varying healthcare data. In: Proceedings of the AAAI national conference on artificial intelligence (AAAI), pp 72–78
53. Zhang J, Wang S, Chen L, Guo G, Chen R, Vanasse A (2019) Time-dependent survival neural network for remaining useful life prediction. In: Proceedings of the Pacific-Asia conference on knowledge discovery and data mining (PAKDD), pp 441–452. Springer, Berlin
54. Zhang J, Wang S, Courteau J, Chen L, Bach A, Vanasse A (2016b) Predicting COPD failure by modeling hazard in longitudinal clinical data. In: Proceedings of the IEEE international conference on data mining (ICDM), pp 639–648
55. Zhang Z, Reinikainen J, Adeleke KA, Pieterse ME, Groothuis-Oudshoorn CG (2018) Time-varying covariates and coefficients in Cox regression models. Ann Transl Med 6(7):121
56. Zhou M (2001) Understanding the Cox regression models with time-change covariates. Am Stat 55(2):153–155
57. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B (Stat Methodol) 67(2):301–320

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

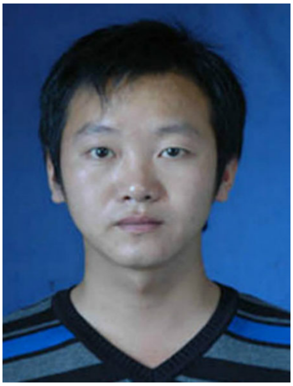
Jianfei Zhang is currently a postdoctoral scholar at Department of Computer and Data Sciences, Case Western Reserve University, Cleveland, USA. He received an M.Sc. from Fujian Normal University, China, in 2013, and a Ph.D. from Université de Sherbrooke, Sherbrooke, Québec, Canada, in 2019. His research interests include health intelligence, computational medicine, biostatistics, machine learning, and data mining. During his Ph.D. studies, he has been specifically working on analytics of electronic health records and COPD data, in cooperation with the research centre of Centre Hospitalier Universitaire de Sherbrooke (CHUS) in Sherbrooke, Québec.



Lifei Chen is currently a professor at Fujian Normal University, China, and an adjunct professor at Université de Sherbrooke, Canada. After receiving a B.Sc. degree from University of Electronic Science and Technology of China and an M.Sc. degree from Tsinghua University, China, he earned his Ph.D. from Xiamen University, China, in 2008. His research interests include data mining, pattern recognition, statistics, and business intelligence.

Yanfang Ye is currently an associate professor at Case Western Reserve University, Cleveland, USA. She was previously an associate professor at West Virginia University (2013–2019). She received her Ph.D. from Xiamen University, China, in 2010, and was the Principal Scientist in Comodo Security Solutions Inc (2010–2013), after holding the position of R&D Deputy Director at Kingsoft Internet Security Corporation (2008–2010). Her primary research areas include cybersecurity, data mining, machine learning, and health intelligence. Her developed algorithms and systems have been incorporated into popular commercial products, including Comodo Internet Security and Kingsoft Antivirus.

Gongde Guo is currently a professor in department of computer science and the Dean of the College of Mathematics and Informatics, Fujian Normal University, China. After receiving his B.Sc. and M.Sc. from Zhejiang University, China, in 1985 and 1987, respectively, he earned a Ph.D. from Ulster University, UK, in 2004. His research interests include data mining, artificial intelligence, machine learning, and cybersecurity.



Rongbo Chen is Ph.D. student at Université de Sherbrooke, Canada. He received his M.Sc. from Shantou University in 2013. His research interests include data mining, machine learning, and bioinformatics.

Alain Vanasse is currently a professor and clinical researcher in Faculté de médecine de famille et de la santé, Université de Sherbrooke, holding a Licentiate of the Medical Council of Canada. He is also the scientific director of the Unité de soutien SRAP du Québec and the director of PRIMUS research group. He received a Ph.D. in Medicine from Université de Sherbrooke, Canada, in 1977, and a Ph.D. in Public Health from Université Catholique de Louvain, France, in 2000. He was awarded the Étienne-Le-Bel Prize in 2019.

Shengrui Wang is currently a professor in Département d'informatique, and the Vice-Dean of the Faculté des sciences at Université de Sherbrooke, Canada. He received his M.A.S from Université Joseph Fourier in 1986 and Ph.D. from Institut National Polytechnique de Grenoble, France, in 1989. His research interests include bioinformatics, data mining, artificial intelligence, information retrieval, health intelligence, and financial machine learning. He chaired the Research Tools and Instruments committee of the Natural Sciences and Engineering Research Council of Canada (NSERC), for Computer, Mathematical and Statistical Sciences (2015–2016 and 2017–2018).

Affiliations

Jianfei Zhang[1,3] · Lifei Chen[1,2] · Yanfang Ye[3] · Gongde Guo[2] · Rongbo Chen[1] · Alain Vanasse[4,5] · Shengrui Wang[1,2]

✉ Shengrui Wang
shengrui.wang@usherbrooke.ca

Jianfei Zhang
jianfei.zhang@usherbrooke.ca

Lifei Chen
clfei@fjnu.edu.cn

Yanfang Ye
yanfang.ye@case.edu

Gongde Guo
ggd@fjnu.edu.cn

Rongbo Chen
rongbo.chen@usherbrooke.ca

Alain Vanasse
alain.vanasse@usherbrooke.ca

[1] Département d'informatique, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada

[2] College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350108, China

[3] Department of Computer and Data Sciences, Case Western Reserve University, Cleveland, OH 44106, USA

[4] Département de médecine de famille et de médecine d'urgence, Université de Sherbrooke, Sherbrooke, QC J1H 5N4, Canada

[5] Centre Hospitalier Universitaire de Sherbrooke (CHUS), Sherbrooke QC J1H 5N4, Canada