Time-Dependent Survival Neural Network for Remaining Useful Life Prediction

Jianfei Zhang[1,2], Shengrui Wang[1,2(✉)], Lifei Chen[1], Gongde Guo[1], Rongbo Chen[2], and Alain Vanasse[3,4]

[1] College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China
{clfei,ggd}@fjnu.edu.cn
[2] Département d'Informatique, Université de Sherbrooke, Sherbrooke, Canada
{jianfei.zhang,shengrui.wang,rongbo.chen}@usherbrooke.ca
[3] Département de Médecine de Famille et de Médecine d'Urgence, Université de Sherbrooke, Sherbrooke, Canada
alain.vanasse@usherbrooke.ca
[4] Centre de Recherche du Centre Hospitalier Universitaire de Sherbrooke, Sherbrooke, Canada

Abstract. Remaining useful life (RUL) prediction has been a topic of practical interest in many fields involving preventive intervention, including manufacturing, medicine and healthcare. While most of the conventional approaches suffer from censored failures arising and statistically circumscribed assumptions, few attempts have been made to predict RUL by developing a survival learning machine that explores the underlying relationship between time-varying prognostic variables and failure-free survival probability. This requires a purely data-driven prediction approach, devoid of any a survival model and all statistical assumptions. To this end, we propose a time-dependent survival neural network that additively estimates a latent failure risk and performs multiple binary classifications to generate prognostics of RUL-specific probability. We train the neural network by a new survival learning criterion that minimizes the censoring Kullback-Leibler divergence and guarantees monotonicity of the resulting probability. Experiments on four datasets demonstrate the great promise of our approach in real applications.

Keywords: RUL prediction · Neural network · Survival learning · Failure risk · Time-varying data

1 Introduction

This paper is about RUL predictive analytics. Let's think about all the in-service machines we use daily, and organisms under pharmaceutical care, from an engine-propelled vehicle on the way to work or a lift going up and down, to an in-patient in the early stages of breast cancer. Imagine that one of these

Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11439, pp. 441–452, 2019.
https://doi.org/10.1007/978-3-030-16148-4_34

should fail (e.g., break down, worsen, die) every day from now on. What impact would that have? The truth is that some failures are just an inconvenience or a financial loss, while others could mean life or death. Therefore, preventive intervention (e.g., predictive maintenance and health care) to defer failures has recently been of great practical interest [23]. But how can we find the right moment for intervention? Providing an answer to this question is the aim of RUL prediction, which seeks to build models for accurate prognostics of RUL in engines, patients and other life entities in machine manufacturing, medicine, epidemiology, economics, etc.

Predicting RUL with great accuracy in the distant future is very challenging and indeed almost impossible in most practical situations. Rather, we turn our attention to the easier and more meaningful prognostic of *how long and how probably to remain failure-free (aka survival before failure)*. RUL prediction here thus refers exclusively to the prognostic of RUL-specific probability, i.e., failure-free survival probability at a specific time. By failure, we refer in particular to a non-recurring, single, adverse incident. With a prediction model in hand, decision makers can be provided with information about when a mechanical fault that can lead to whole system failure might take place. For instance, the probability of fault-free steering in a 15-year-old vehicle engine up to 50,000 km is 80% but for up to 80,000 km the probability drops to 10%; this knowledge allows for predictive maintenance (before 50,000 km) which may prolong engine usage and holds out the promise of considerable cost savings.

In this paper, we propose a time-dependent survival neural network (TSNN) which additively estimates a latent failure risk and performs multiple classifications to generate prognostics of RUL-specific probability. We provide a new censoring Kullback-Leibler divergence for evaluating the dissimilarity between the binary classification probabilities and the actual survival process. A generalized survival learning approach is developed to minimize such divergence, running under a constraint that guarantees monotonicity of the resulting probability. Experimental results on four real-world datasets from the fields of engineering, medicine and healthcare demonstrate the promise of TSNN in real applications. The paper makes the following primary contributions: (1) It develops a purely data-driven prediction approach free of any existing survival model and all statistical assumptions. (2) It transforms prediction into multiple classifications that potentially relate to each other. (3) It makes full use of time-varying prognostic variables by exploring latent failure risk in an additive manner. (4) It provides a learning criterion that allows automatic exploitation of data with censoring.

2 Motivation

To build a model for RUL prediction, training data should allow us to capture information regarding prognostic variables leading to failure. In the observational world, however, we need to know whether failure, dropout or study cutoff comes first. Thus, the outcome of interest in data is not only whether or not a failure occurred, but also when that failure occurred. Traditional regression methods

are not able to include both the failure and time aspects as the outcome in the model, though they are used to perform a time prediction on most time series data to answer the questions like *"How many days are left before failure?"*. In contrast, a considerable number of survival models have long been developed to utilize the partial information on each entity with censored failure and provide unbiased survival estimates. They incorporate data from multiple time points across entities for prediction of failure probability over time and thus can answer the question like *"How does the risk of failure change over time?"*.

These statements naturally lead one to consider using survival models for predicting RUL-specific probability. However, the three prominent survival modeling approaches developed primarily for retrospective cohort studies are characterized by their inherent disadvantages [20]. (1) Models utilizing the non-parametric approach, an analysis intended to generate unbiased descriptive statistics, cannot generally be used to assess the effect of multiple prognostic variables on failure. (2) The parametric approach suffers from an even more critical weakness, relying as it does on the assumption that the underlying failure distribution (i.e., how the probability of failure changes over time) has been correctly specified. (3) The semi-parametric approach requires an assumption on how the variables influence the risk of failure, which is often violated in practical use.

The increasing availability of complex lifetime data with time-varying prognostic variables poses more challenges to these approaches and is stimulating numerous research efforts that use data mining and machine learning methods in conjunction with survival models. Typical examples include multi-task learning [9,10,13,19], active learning [18], neural networks [4,7], transfer learning [11], Bayesian inference [16] and feature engineering [12,24] that extended to the semi-parametric Cox proportional-hazards model [3], as well as a random forest technique [6] that employed a non-parametric Nelson-Aalen estimator to predict the time to censored failures for establishing terminal nodes of forest. These approaches still suffer from the implausibility of the survival study hypothesis and prior knowledge and therefore cannot be selected as prediction models for our desired output. For example, although the feed-forward network proposed in [4] preserved most of the advantages of a typical Cox proportional-hazards hypothesis, it was still not the optimal way to model the baseline variations [7]. In addition, these time-to-failure prediction methods are not specifically designed to handle time-varying prognostic variables. The common approach employed is to predict the survival probability at a certain time (i.e., RUL-specific probability in this paper) using only the values of variables at that moment. The historical values are discarded in prediction but have been proven to latently affect the survival probability [15,25,26]. These arguments in turn demonstrate a need for a prediction model that releases priori statistical assumptions, explores latent risk and makes full use of time-varying prognostic variables.

3 Proposed Approach

Imagine a binary classification performed to predict failure of a machine in a given t-day time window; i.e., to answer the question *"Will a machine remain*

failure-free over the next t days?". This allows us to transform the original RUL prediction problem into a series of binary classification problems, as long as each has an RUL-specific output probability that the actual RUL, say T, is not earlier than t, denoted $\Pr(T > t)$. In this section, we provide a neural network that allows data to drive the survival learning inference, i.e., devoid of any a survival model and all statistical assumptions, to perform the binary classifications.

3.1 Time-Dependent Survival Neural Network

Survival Neural Network Classifier Architecture. We concentrate our attention on a one-hidden-layer neural network, i.e., three-layer networks with V input neurons, K output neurons and H 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 $h = 1, 2, \ldots, H$ computes a sum of all the inputs weighted by $\mathbf{w}_h^{\text{hide}} \in \mathbb{R}^V$, adds a bias b_h^{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 $\mathbf{w}_k^{\text{out}} \in \mathbb{R}^H$, adds a bias b_k^{out}, and then applies the activation function to obtain $S_k(\mathbf{x})$.

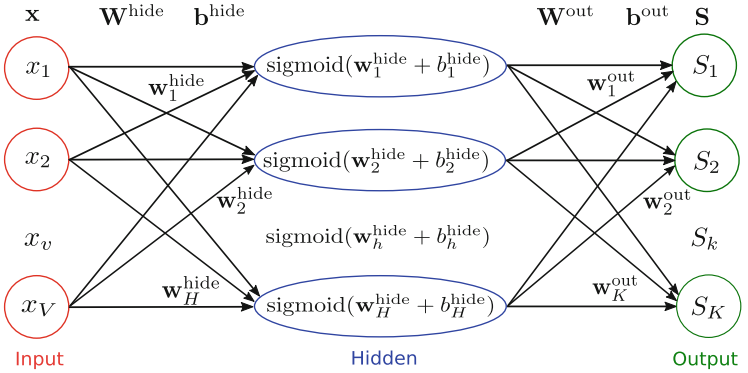

Fig. 1. A survival neural network

Survival neural network, in principle, is a combination of multiple classifiers, each performing a binary classification on every entity that is or is not still failure-free. Hence, we interpret $S_k(\mathbf{x}) \in (0, 1)$ as the classification probability that the entity with variables $\mathbf{x}$ remains failure-free by τ_k. In doing so, with the K classification outputs over disjoint time snapshots $\tau_1 < \tau_2 < \cdots < \tau_K$ in hand, we are able to estimate an RUL-specific probability curve which depicts *how long and how probably the entity will remain failure-free*. Hence, given the weights $\mathbf{W}^{\text{hide}} \in \mathbb{R}^{H \times V}$, $\mathbf{W}^{\text{out}} \in \mathbb{R}^{K \times H}$ and the biases $\mathbf{b}^{\text{hide}} \in \mathbb{R}^H$, $\mathbf{b}^{\text{out}} \in \mathbb{R}^K$ for computing the hidden and output layers, respectively, we scale the outputs $\mathbf{S}(\mathbf{x}) \in \mathbb{R}^K$ to the range of the logistic sigmoid function that is applied

component-wise to the vector, i.e.,

$$\mathbf{S}(\mathbf{x}) = \texttt{sigmoid}\left(\mathbf{W}^{\text{out}} \cdot \texttt{sigmoid}\left(\mathbf{W}^{\text{hide}} \cdot \mathbf{x} + \mathbf{b}^{\text{hide}}\right) + \mathbf{b}^{\text{out}}\right). \tag{1}$$

For the time-varying variables, the output is yielded with input values observed at corresponding time snapshots, that is, $S_k(\mathbf{x}) = S(\mathbf{x}(\tau_k))$, where $\mathbf{x}(\tau_k)$ consists of V observational values at τ_k. Nevertheless, this approach does not take the historical variable values into account in estimating the failure risk.

TSNN with Latent Failure Risk Estimation. Note that the exponential component in Eq. 1 can serve as the failure risk, like the conventional cumulative risk in the Cox [3] and accelerated failure time (AFT) models [21]. Obviously, the risk is not dependent on any historical values at all. To address this issue, we propose the form $\gamma(*, t)$ to stand for the decay ratio of the failure risk. By such decay, we can model the amount of the latent risk produced by the values at time $*$ remaining at time $t(\geq *)$. This can be an exponential function of time in the form $\gamma(*, t) = \exp\{G(* - t)\}$. Simply, we make the decay coefficient G take a positive 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 all historical values observed at time points $j \in R(t)$ before t, we estimate the failure risk in an additive manner and compute the TSNN output at τ_k as follows:

$$S_k(\mathbf{x}) = \left(1 + \frac{1}{|R(\tau_k)|} \sum_{j \in R(\tau_k)} \exp\left\{G(j-t)\right\} \exp\left\{-\mathbf{W}^{\text{out}} \phi(\mathbf{x}(j)) - \mathbf{b}^{\text{out}}\right\}\right)^{-1}$$

$$\phi(\mathbf{x}(j)) = \left(1 + \frac{1}{|R(j)|} \sum_{u \in R(j)} \exp\left\{G(u-t)\right\} \exp\left\{-\mathbf{W}^{\text{hide}} \mathbf{x}(u) - \mathbf{b}^{\text{hide}}\right\}\right)^{-1}.$$

Our approach can be thought of as a generalization of multi-task classification, which enables flexible modeling of RUL-specific probability in parallel. Each task executes on all training entities but has an individual variable input. As was discussed in [10], 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.

3.2 RUL-Specific Probability Evaluation

Survival Process. Given N_{tr} training entities, the actual survival process for entity i can be modulated 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 failure occurs by time τ_k, taking a value of 1 up to τ_k, and 0 thereafter, and -1 for unknown cases. Once $\varepsilon_i(t)$ becomes "0" it will not turnover to"1", there are thus $K + 1$ possible legal sequences of the form $(1, 1, \ldots, 0, 0, \ldots)$, including the sequences 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 lifetime are strictly greater than those after failure, i.e., $\varepsilon_i(\tau_k) > \varepsilon_i(\tau_{k'})$, $\forall k \in \mathcal{K}_i^1$ and $\forall k' \in \mathcal{K}_i^0$.

Censoring Kullback-Leibler Divergence. The TSNN cannot be an effective prediction model unless it achieves the objective that *the predicted RUL-specific probabilities approach the actual survival process*. In order to qualify such approachability, we define the censoring Kullback-Leibler (KL) divergence, an alternative to the relative error [17], between the distributions of the RUL-specific probability $S_k \in (0, 1)$ and the survival status $\varepsilon(\tau_k) \in \{0, 1\}$, as follows:

$$D_i(k) = \varepsilon_i(\tau_k) \ln \frac{\varepsilon_i(\tau_k)}{S_k(\mathbf{x}_i)} + (1 - \varepsilon_i(\tau_k)) \ln \frac{1 - \varepsilon_i(\tau_k)}{1 - S_k(\mathbf{x}_i)}.$$

The optimal weights make S_k 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 is then to minimize $D_i(k)$ over time snapshots $\mathcal{K}_i^{\{1,0\}} = \mathcal{K}_i^0 \cup \mathcal{K}_i^1$ at which survival statuses are known, for all N_{tr} training entities.

3.3 TSNN Learning

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 the monotonicity:

$$\Delta_i(k, k+1) = S_k(\mathbf{x}_i) - S_{k+1}(\mathbf{x}_i) > 0, \forall k = 1, 2 \ldots, K-1, \forall i = 1, 2, \ldots, N_{\text{tr}}.$$

The proven penalty method converts the constrained optimization problem into a series of unconstrained optimization problems. Accordingly, we utilize the static penalty [14] that along with its parameter λ incurred for violating the inequality constraints and minimize the average error computed by

$$E = \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \left(\left| \mathcal{K}_i^{\{1,0\}} \right|^{-1} \sum_{k \in \mathcal{K}_i^{\{1,0\}}} D_i(k) + \frac{\lambda}{K-1} \sum_{k=1}^{K-1} \left(\min \left\{ 0, \Delta_i(k, k+1) \right\} \right)^2 \right).$$

We train the neural network using the forward-only Levenberg-Marquardt algorithm presented in [22], which inherits the speed advantage of the Gauss-Newton algorithm and the stability of the steepest descent method.

4 Experiments

4.1 Data and Pre-processing

Four lifetime datasets were drawn from the prognostics data repository provided by the PCoE of NASA Ames, the Surveillance, Epidemiology, and End Results

(SEER) statistics database, and 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 pressure compressor and fan degradation. The randomized Battery usage dataset employed the first 20 cells, each with 42 galvanostatic voltage curves. Failure was censored for 45.8% of batteries and 10 prognostic variables were extracted from the time series of temperature and current (mA) every 30 s. For the breast Cancer dataset, RULs were computed by subtracting the date of diagnosis from the date of last contact (the study cutoff). The healthy Aging data were acquired directly between Dec 2008 and Nov 2009 from respondents in a survey, which focused on the health of Canadians aged 45 and over by examining the various factors that impact healthy aging. A total of 3,390 valid interviews covering the population living in the ten provinces were used. Table 1 summarizes the statistics, including data size N, dimensionality V, censoring rate C, missing-value percentage M and failure of interest. Categorical variables were transformed into numerical values by means of the probabilistic frequency estimator presented in [2]. Afterwards, missing values were filled in via a linear regression provided by [8]. In order to reduce data redundancy and improve data integrity, all values were normalized.

Table 1. Statistics of the four lifetime datasets

Dataset (source)	N	V	C	M	Failure of interest
Engine (NASA)	1,416	21	27.4%	11.3%	Compressor and fan degradation
Battery (NASA)	842	10	45.8%	5.9%	30% fade in rated battery capacity
Cancer (SEER)	3,390	18	19.3%	15.7%	Breast cancer caused death
Aging (CCHS)	7,611	35	34.5%	26.2%	Retirement and disability

4.2 Competitors

We compared TSNN against several state-of-the-art methods. CoxNN [4] replaces the linear exponent of the Cox hazard by a nonlinear artificial neural networks output; TD-Cox [5] extends the Cox model to time-varying variables; AFT [21] assumes a Weibull RUL distribution in our experiments; EN-BJ [1] extends the least squares estimator to the semi-parametric linear regression model in which the error distribution is completely unspecified; MTLR [13] models RUL distribution by combining multi-task logistic regression in a dependent manner, with the regularization parameter chosen via an additional 10-fold cross validation (10CV); RSF [6] estimates conditional cumulative failure hazard by aggregating tree-based Nelson-Aalen estimators.

We also studied TSNN with simplified configurations, yielding three models as follows. SNN does not estimate the latent risk. Rather, it predicts the output probabilities using Eq. 1 with the time-varying input $\mathbf{x}(t)$; KM-TSNN uses a Kaplan-Meier (KM) estimator to fill in the RULs for censored cases, according

to the method introduced in [17]; KM-SNN uses a KM estimator to fill in the RULs for censored cases in SNN. The parameters for competitors were those used in the original papers. For TSNN, KM-TSNN, SNN and KM-SNN, we set the hidden layer to $H = 4$ neurons. An output layer with $K = 20$ was used in analyses of the Engine and Battery datasets, and $K = 12$ in the Cancer and Aging datasets. The penalty parameter λ was chosen through an independent 10CV on the training data. The decay coefficient $G = 1.5$ was used in TSNN and KM-TSNN.

4.3 Evaluation Metrics

Performance on the N_{te} test entities was evaluated in terms of three independent metrics: the failure AUC (FAUC), the concordance index (C-index) and the censoring Brier score (CBS), redefined as follows ($\mathbb{1}$ is the indicator function)

$$\text{FAUC} = \frac{\sum_{i:\epsilon_i(\tau_K)=0} \sum_{j:\epsilon_j(\tau_K)=1} \mathbb{1}\left\{S_K(\mathbf{x}_i) < S_K(\mathbf{x}_j)\right\}}{|\{i : \epsilon_i(\tau_K) = 0\}| \times |\{j : \epsilon_j(\tau_K) = 1\}|}$$

$$\text{C-index} = \frac{\sum_{i:\epsilon_i(\tau_K)=0} \sum_{j:T_i<T_j} \mathbb{1}\left\{S_{\min\{\mathcal{K}_i^0\}}(\mathbf{x}_i) < S_{\min\{\mathcal{K}_i^0\}}(\mathbf{x}_j)\right\}}{|\{i : \epsilon_i(\tau_K) = 0\}| \times |\{j : T_i < T_j\}|}$$

$$\text{CBS} = \frac{1}{N_{\text{te}}} \sum_{i=1}^{N_{\text{te}}} \left(1 - \varepsilon_i(\tau_K) - S_K(\mathbf{x}_i)\right)^2.$$

FAUC provides a probability measure of classification ability at a pre-specified time snapshot (e.g., at τ_K in our case). It qualifies the model's ability to address the issue *"Is i likely to remain failure-free by time t?"* C-index serves as a generalization of the FAUC, giving an estimate of how accurately to answer the question *"Which of i and j is more likely to remain failure-free?"* CBS measures an ensemble prediction error across the test data, i.e., the power of a model to address the issue *"How accurate is the prediction that i will remain failure-free?"*.

4.4 Results and Discussion

From the 10CV results on the test data, shown in Table 2, it is evident that TSNN outperforms all the other models but FAUC yielded by MTLR on the Cancer dataset. The alternatives SNN and KM-TSNN perform second-best, with the sole exception of FAUC on the Cancer dataset (second-best results yielded by TD-Cox) and FAUC on the Aging dataset (by EN-BJ). The superior performance of TSNN relative to KM-TSNN, and of SNN relative to KM-SNN, reveal that our survival learning approach to minimize the censoring KL divergence can effectively cope with censored data in comparison to the conventional survival 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. CoxNN yields even lower accuracies in comparison to TD-Cox, demonstrating that use

of the risk nonlinearity property alone does not enhance the Cox model [7]. Note that TSNN and SNN 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 [13]. Note also the extremely low CBS achieved by TSNN on the four datasets indicates high accuracy in predicting the absolute RUL-specific probability and high confidence in forecasting failure.

Table 2. Comparison of the 10CV FAUC, C-index and CBS results on the test data, in the form of mean (standard deviation). The best results are in bold and the second-best performances are underlined.

	FAUC	C-index	CBS	FAUC	C-index	CBS
	Engine			Battery		
TSNN	**.744**(.017)	**.753**(.028)	**.163**(.018)	**.810**(.022)	**.761**(.014)	**.212**(.029)
SNN	.719(.038)	.724(.026)	.185(.023)	.769(.015)	.710(.029)	.229(.038)
KM-TSNN	.731(.024)	.678(.040)	.248(.011)	.695(.032)	.733(.015)	.261(.034)
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)
TD-Cox	.740(.047)	.587(.029)	.276(.018)	.754(.028)	.686(.017)	.301(.048)
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)
SNN	.785(.034)	.756(.017)	.217(.008)	.706(.020)	.722(.018)	.221(.015)
KM-TSNN	.694(.041)	.681(.024)	.226(.047)	.730(.016)	.736(.022)	.166(.027)
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)
TD-Cox	.753(.019)	.642(.025)	.297(.018)	.652(.045)	.628(.038)	.359(.007)
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 censoring KL divergence based survival learning may enable TSNN (and SNN) to recommend the right moment for preventive intervention. For this investigation, we performed a case study on the Engine dataset. The engines that experienced failure were divided into 6 groups according to their times to failure. In each sub-figure of Fig. 2, we plotted an RUL curve according to the average RUL-specific probability predicted by each model on the corresponding group of engine failures. It can be seen from the respective gray areas that

TSNN (plotted by the salmon dashed curve) yields a significantly lower average probability over all data (i.e., all engine failures) in comparison to other models, mainly because latent risk estimation can help in amending the relationship between latent risk and RUL-specific probability. This means that, using our TSNN, the 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.

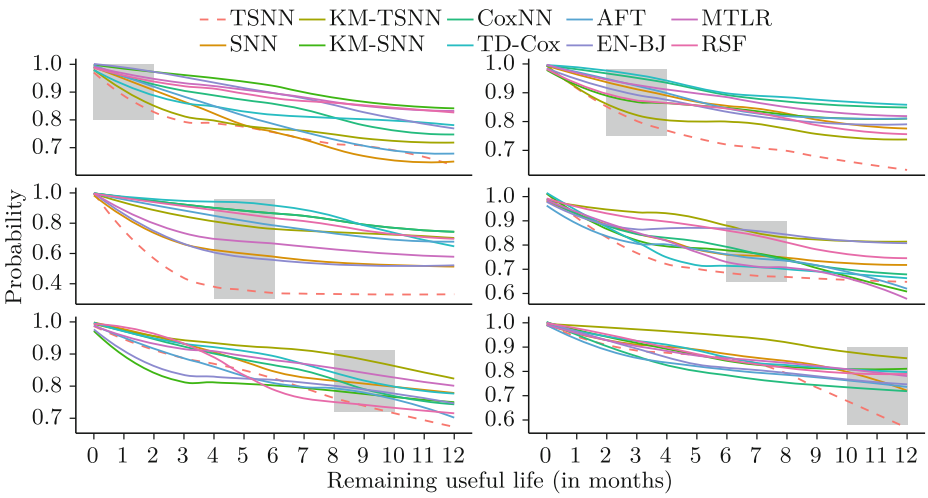

Fig. 2. Change in predicted RUL-specific probability curve for engines. The 6 sub-figures are plotted for the engines that failed at intervals of 2 month (see each gray rectangle), from the 1st month to the 12th month. Every curve in each sub-figure is the average predicted probability of the engines.

In order to provide a deeper insight into the functionality of TSNN, we set a varying K value of 1, 2, 3, 4, 6, 8, 12 and 24 when it runs on the Aging dataset (in 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. 3 change less than 8%, 11% and 9%, respectively; this demonstrates that users can count on TSNN as reliable, as it won't fluctuate enormously with change in the output layer of neural networks.

Figure 4 shows the average results of TSNN with a varying decay coefficient G, which might lead to an inaccurate risk estimate and therefore a poor predictive ability when it 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 it takes a value in the range [1,2].

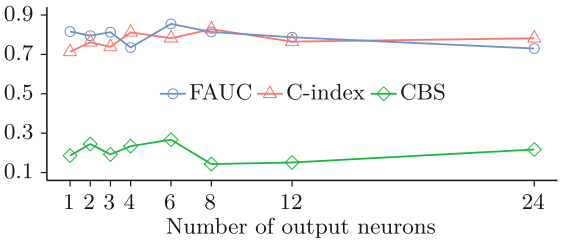

Fig. 3. Change in TSNN performance on the aging dataset with varying K

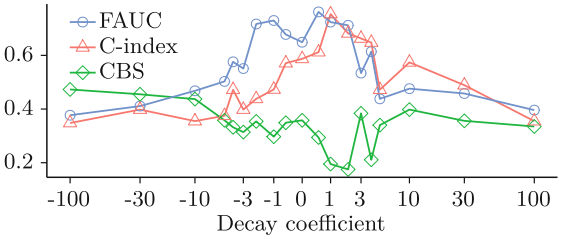

Fig. 4. Change in TSNN performance on the aging dataset with varying G

5 Conclusions

In this paper, we proposed a data-driven TSNN model for RUL prediction. TSNN performs an additive latent failure risk estimation and multiple binary classifications for predicting RUL-specific probabilities. The new survival learning approach optimizes a neural network by minimizing the censoring KL divergence between the resulting probabilities and the actual survival process. In addition, the learning criterion constrains the RUL-specific probability to decrease as time elapses. Experimental results on four lifetime datasets confirm that our model outperforms several state-of-the-art models and is therefore a good candidate for developing a decision-making assistance system to help with preventive intervention.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61672157, the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. 396097-2010, the program PAFI of Centre de Recherche du CHUS.

References

1. Buckley, J., James, I.: Linear regression with censored data. Biometrika **66**(3), 429–436 (1979)
2. Chen, L., Wang, S.: Central clustering of categorical data with automated feature weighting. In: IJCAI, pp. 1260–1266 (2013)
3. Cox, D.R.: Regression models and life tables. J. R. Stat. Soc. Ser. B. Stat. Methodol. **34**, 187–220 (1972)

4. Faraggi, D., Simon, R.: A neural network model for survival data. Stat. Med. **14**(1), 73–82 (1995)
5. Fisher, L.D., Lin, D.Y.: Time-dependent covariates in the Cox proportional-hazards regression model. Annu. Rev. Public Health **20**(1), 145–157 (1999)
6. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S.: Random survival forests. Ann. Appl. Stat. **2**, 841–860 (2008)
7. Katzman, J., Shaham, U., Bates, J., Cloninger, A., Jiang, T., Kluger, Y.: DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Med. Res. Methodol. **18**, 18–24 (2018)
8. Kim, H., Golub, G.H., Park, H.: Imputation of missing values in DNA microarray gene expression data. In: CSB, pp. 572–573 (2004)
9. Li, H., Ge, Y., Zhu, H., Xiong, H., Zhao, H.: Prospecting the career development of talents: a survival analysis perspective. In: KDD, pp. 917–925 (2017)
10. Li, Y., Wang, J., Ye, J., Reddy, C.K.: A multi-task learning formulation for survival analysis. In: KDD, pp. 1715–1724 (2016)
11. Li, Y., Wang, L., Wang, J., Ye, J., Reddy, C.K.: Transfer learning for survival analysis via efficient L2, 1-norm regularized Cox regression. In: ICDM, pp. 231–240 (2017)
12. Li, Y., Xu, K.S., Reddy, C.K.: Regularized parametric regression for high-dimensional survival analysis. In: SDM, pp. 765–773 (2016)
13. Lin, H.C., Baracos, V., Greiner, R., Chun-Nam, J.Y.: Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In: NIPS, pp. 1845–1853 (2011)
14. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evol. Comput. **4**(1), 1–32 (1996)
15. Moghaddass, R., Rudin, C.: The latent state hazard model, with application to wind turbine reliability. Ann. Appl. Stat. **9**(4), 1823–1863 (2014)
16. Sinha, D., Ibrahim, J.G., Chen, M.: A Bayesian justification of Cox's partial likelihood. Biometrika **90**(3), 629–641 (2003)
17. Street, W.N.: A neural network model for prognostic prediction. In: ICML, pp. 540–546 (1998)
18. Vinzamuri, B., Li, Y., Reddy, C.K.: Active learning based survival regression for censored data. In: CIKM, pp. 241–250 (2014)
19. Wang, L., Li, Y., Zhou, J., Zhu, D., Ye, J.: Multi-task survival analysis. In: ICDM, pp. 485–494 (2017)
20. Wang, P., Li, Y., Reddy, C.K.: Machine learning for survival analysis: a survey. ACM Comput. Surv. **51**(6), 1–36 (2019)
21. Wei, L.J.: The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. Stat. Med. **11**(14–15), 1871–1879 (1992)
22. Wilamowski, B.M., Yu, H.: Neural network learning without backpropagation. IEEE Trans. Neural Netw. **21**(11), 1793–1803 (2010)
23. Wu, Y., Yuan, M., Dong, S., Lin, L., Liu, Y.: Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neural Comput. **275**, 167–179 (2018)
24. Yu, S., et al.: Privacy-preserving Cox regression for survival analysis. In: KDD, pp. 1034–1042 (2008)
25. Zhang, J., Chen, L., Vanasse, A., Courteau, J., Wang, S.: Survival prediction by an integrated learning criterion on intermittently varying healthcare data. In: AAAI, pp. 72–78 (2016)
26. Zhang, J., Wang, S., Courteau, J., Chen, L., Bach, A., Vanasse, A.: Predicting COPD failure by modeling hazard in longitudinal clinical data. In: ICDM, pp. 639–648 (2016)