Reliability Engineering and System Safety 217 (2022) 108033

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Attention-based deep survival model for time series data

Xingyu Li [a], Vasiliy Krivtsov [a,b,*], Karunesh Arora [a]

[a] *Consulting and Research Center, Ford Motor Company, Dearborn, United States*
[b] *Department of Mechanical Engineering, University of Maryland, United States*

ARTICLE INFO

Keywords:
Survival analysis
Internet of things
Deep learning
Sequence modeling
Prognostics

ABSTRACT

In the era of internet of things and Industry 4.0, smart products and manufacturing systems emit signals tracking their operating condition in real-time. Survival analysis shows its strength in modeling such signals to determine the condition of in-service equipment and products to yield critical operational decisions, i.e., maintenance and repair. One appealing aspect of survival analysis is the possibility to include subjects in the model which did not have their failure yet or when the exact failure time is unknown.

NN-based survival models, i.e., deep survival models, show superior performance in modeling the non-linear relationship between the reliability function and covariates. We propose a novel deep survival model, seq2surv, to incorporate the seq2seq structure and attention mechanism to enhance the ability to analyze a sequence of signals in the survival analysis. Similar to the seq2seq model which shows superior performance in machine translation, we designed the seq2surv model to translate from a sequence of signals to a sequence of survival probabilities and to update the reliability predictions along with real-time monitoring. Our results show that the seq2surv model outperforms existing deep survival approaches in terms of higher prediction accuracy and lower errors in the survival function estimation on both simulated and real-world datasets.

1. Introduction

Automobile manufacturers spend 2.5–3.0% of their sales revenue on fixing vehicle issues. Customer satisfaction and feedback are essential in marketing the product and preventing existing problems from recurring [1]. Manufacturers seek to address the challenges of effectively utilizing diverse databases including customer feedback, laboratory test, maintenance, and field-tracking to identify and resolve product defects at design and manufacturing phases [2]. With recent development in new communication systems and devices, cybersecurity standards, and deployable artificial intelligence, manufacturers have started to understand failures by data-driven prognosis techniques thus improving their profit margin by improving product design and preparing spare parts based on the failure prediction [3,4].

Data-driven prognosis analysis can leverage heterogeneous databases to derive actionable insights to enhance the development of innovative products [2] and improve the reliability of existing products [5]. Moreover, bridging between manufacturing data and field reliability makes root cause analysis of defects possible, even for defects that cannot be easily identified by technicians [6]. Manufacturers seek to understand field failures by modeling the distribution of failure times [7] and operational actions [8].

Survival analysis shows its strength in modeling such data to help determine the condition of in-service equipment and products for the entire lifeline, which can be classified into univariate, bivariate, and those containing covariates (a.k.a., explanatory variables). A comprehensive summary of these models is presented in [9]. In the univariate approach, the survival variable is typically the age of the product. It is widely discussed by many authors [10]. In the bivariate approach, the data are partitioned in a two-dimensional plane with one axis representing product age and the other axis representing product usage, e.g., mileage in the automotive industry [11]. There are studies that directly estimate the bivariate lifetime distribution of products [12]. In the covariates approach, models leverage exploratory variables describing the design, production and market-related information on the actual environment in which the product is used. Many procedures have been developed for collecting and analyzing warranty claim data [13,14].

Extensive research has been conducted in implementing the covariates approach in automotive case studies. Karim and Suzuki designed a Weibull regression model as a function of reliability-related covariates [14]. Attardi et al. [15] use a mixed-Weibull regression model for the analysis of automotive warranty claims data with engine type and car model used as covariates. Cox proportional hazards (CPH) model [16] is a semi-parametric model that estimates the effects of observed covariates on the hazard function. CPH model assumes that the risk can be computed by a linear combination of covariates and

[*] Corresponding author at: Consulting and Research Center, Ford Motor Company, Dearborn, United States.
E-mail address: vkrivtso@ford.com (V. Krivtsov).

https://doi.org/10.1016/j.ress.2021.108033
Received 18 May 2021; Received in revised form 3 September 2021; Accepted 5 September 2021
Available online 20 September 2021

links risks to the baseline hazard via the exponential function to enforce positivity. Krivtsov et al. uses a Cox regression model to understand the failure mechanism of tires [17].

It is interesting to note that binary classification models, one of the most common machine learning applications, are used in industrial settings, where survival methodology is applicable. Binary classifiers can provide predictions for a certain time slice but lose the interpretability and flexibility in modeling the distribution of an event as a continuous function of time. Moreover, in applications with a substantial amount of censoring, the use of binary classifiers tends to be problematic. For example, for the steering systems, the percentage of uncensored data, i.e., failure event occurs before the end of observation, is around 1% [18]. While binary classifiers typically ignore censored observations, one of the main objectives in survival analysis is to account for them [19].

Several challenges exist in implementing survival analysis in industrial practices. Classical statistics techniques for Cox regression rely on non-parametric or semi-parametric methods for the survival function estimation, primarily because they make working with censored data relatively straightforward [20]. However, non-parametric methods may suffer from the problem of dimensionality, when learning individual hazards, especially when the size of co-variate is large. Semi-parametric approaches usually depend on the prevailing assumption of constant risks over a lifetime, which is very likely to be unrealistic in many practical scenarios encountered in healthcare, predictive maintenance, econometrics, or operations research [8,20,21].

As such, a richer family of deep survival models have been developed to better fit survival data with nonlinear risk functions. Since neural networks (NNs) can learn highly complex and nonlinear functions, researchers have used NN-based to build survival models, which offer greater flexibility and accuracy in modeling the relationship between covariates and time-to-event, also known as deep survival model. Deep survival models combine the advantages of deep NNs to more accurately model complex functions with a better ability to handle the censored data.

Today, products and manufacturing devices are getting smarter. Sensors and smart chips are commonly used in machines and products to monitor use rate, system load, and various environmental variables in real-time. Next-generation reliability data are much richer with time-series features describing system operating and environmental information [22]. With the development of connected vehicle technology, vehicle speed, acceleration, temperature, pressure, and vast amounts of network data are available for traffic safety evaluation [23], remote vehicle prognostics and health management [24]. Connected manufacturing devices integrate industrial production and store machine-related data as large-scale time series from every aspect of production, transportation and after-sales [25,26].

Moreover, products with various usage rates, e.g., due to the heterogeneous customer behavior, lead to distinct failure processes. The excavators with a high usage rate are likely to experience more failures than those with either a moderate usage rate or low usage rate [27]. Another study also reveals that the reliability of the power system of electric vehicles depends on travel patterns and driver's behavior [28]. An increasing amount of time series data can be collected from the mounted sensors in vehicles and manufacturing systems. Vast data provide opportunities to understand the field failures but also challenge the existing deep survival models to effectively prognosticate failure [22], for example, high complexity and nonlinearity in the machine condition data [29].

Time series data often has periodic temporal features due to seasonality or complex patterns underlying the activities measured and noticeable noise from communication and measurement [30]. Accurate prognosis relies on effective feature extraction from the whole series to capture valuable information and discard irrelevant noise [31]. Unfortunately, all existing work, either statistical analysis or NN models, does not attain this goal well. To our best knowledge, no prognosis model is structured to take advantage of such time series data and grasp the emerging opportunity that has arisen from the breakthrough in information and communication technology.

In this study, we systematically review the existing deep survival models in multidisciplinary studies, spanning from disease management to automotive analysis and bridge the likelihood functions in survival analysis to the loss function in machine learning models. Machine learning models, which have been used in the parameterization of survival models, are summarized and compared according to their strengths and weakness, especially for the era of internet of things and Industry 4.0. We build on the previous study [32] and propose a novel deep survival model, seq2surv, that can effectively analyze the time-series data to address the emerging opportunities and challenges in Industry 4.0, for example, connected vehicles and smart manufacturing systems. Seq2surv model substantially improves the accuracy of predicting survivability of each individual among all existing deep survival models, indicating vast potentials in improving product designs and manufacturing processes. Our paper is structured as follows: after some preliminaries in Section 2, in Section 3, we review the fundamentals of survival analysis as well as the existing deep survival models. In Section 4, we address the challenges in analyzing the time-series features and making a prediction. In Section 5, we describe our seq2surv model, which predicts survival curves from correlated time series data. In Section 6, we implemented the proposed model on simulated and real-world time series datasets to show its effectiveness. In Section 7, we summarize our work and propose future research directions.

2. Preliminaries

In this study, we aim to model the distribution of failure time, denoted by T^*. In most cases, not all failure times are observable. We denote a right-censored observation as C^*, whereby the failure after C^* is censored. We assume that a failure is happened, once it is seen and recorded. Survival time T and failure indicator D are defined respectively as

$$T = \min\{T^*, C^*\}, \tag{1}$$

$$D = \mathbf{1}\{T^* \le C^*\}. \tag{2}$$

We denote $f(t)$ as the probability density function of the failure time, and $F(t)$ as the cumulative distribution function.

$$F(t) = \mathbb{P}(T^* \le t) = \int_0^t f(\tau)d\tau \tag{3}$$

The survival function $S(t)$ is the probability that the failure does not happen at time t.

$$S(t) = \mathbb{P}(T^* > t) = 1 - F(t) \tag{4}$$

Some survival models are often expressed in terms of the hazard function $h(t)$, which is the conditional probability that the failure is going to happen (in a small interval) given that the failure has not happened until that interval. Given a small time increment Δt, we have

$$h(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t}\mathbb{P}(t \le T^* < t + \Delta t | T^* \ge t) = 1 - F(t) = \frac{f(t)}{S(t)}. \tag{5}$$

The cumulative hazard function can be expressed as a function of $S(t)$, namely

$$H(t) = \int_0^t h(\tau)d\tau = \int_0^t \frac{\frac{d(1-S(\tau))}{dS(\tau)}}{S(\tau)} dS(\tau) = \int_0^t -\frac{1}{S(\tau)}\partial S(\tau) = -\log S(t). \tag{6}$$

where, $S(t) = \exp[-H(t)]$. Given survival time t and failure indicator $d \in \{0, 1\}$, we have the likelihood function for the right-censored survival data as follows

$$\mathbb{P}(T = t, D = d) = \mathbb{P}(T = t, C^* > t)^d \mathbb{P}(T^* > t, C^* = t)^{1-d}. \tag{7}$$

Now, assume that the T^* and C^* are independent, the likelihood function can be separated into two parts representing failure and censored times

$$\begin{aligned}\mathbb{P}(T=t, D=d) &= [\mathbb{P}(T^*=t)\mathbb{P}(C^*\geq t)]^d[\mathbb{P}(T^*>t)\mathbb{P}(C^*=t)]^{1-d}\\ &= [f(t)^d S(t)^{1-d}][f_{C^*}(t)^{1-d}(S_{C^*}(t)+f_{C^*}(t))^d]. \end{aligned} \tag{8}$$

Denote the survival time and failure result for individual i as t_i and d_i, respectively; then the likelihood of failure time distribution becomes

$$L_i = f(t_i)^{d_i} S(t_i)^{1-d_i}. \tag{9}$$

Assuming we have multiple individuals, the likelihood of failure time distribution becomes

$$L = \prod_i f(t_i|\mathbf{x}_i)^{d_i} S(t_i|\mathbf{x}_i)^{1-d_i} = \prod_i h(t_i|\mathbf{x}_i)^{d_i} S(t_i|\mathbf{x}_i), \tag{10}$$

where $\mathbf{x}_i$ is covariates of individual i. d_i is the indicator (failed or censored) of individual i. This function is referred to as the *full likelihood function*.

3. Literature review

Given a parametric assumption for a distribution of survival times, a variaty of survival models and parameter estimation methods have been built in the framework of (deep) survival analysis. In this section, we summarize the existing (deep) survival models by sorting them as time-invariant and time-dependent survival models.

3.1. Time-invariant survival models

Cox proportional hazards model (CPH) is one of the most popular models in survival analysis, which provides a semi-parametric specification of the hazard rate in continuous time [16]. The hazard function is separated into two parts $h_0(t)$ and $g(\mathbf{x})$, representing the contribution of the baseline hazard and the covariates (features), respectively:

$$h(t|\mathbf{x}) = h_0(t)\exp[g(\mathbf{x})], \qquad g(\mathbf{x}) = \beta^T\mathbf{x}, \tag{11}$$

where $h_0(t)$ is the non-parametric baseline hazard, which can be distribution-free or parametrized, and $\exp[g(\mathbf{x})]$ is the relative risk function, which is time-invariant in this formulation. The parametric part is fitted by maximizing the Cox partial likelihood

$$\begin{aligned} L_i &= \mathbb{P}(D_i=1, t=T_i|\sum_j D_j = 1, t=T_i)\\ &= \frac{h_0(t)\exp[g(\mathbf{x}_i)]}{\sum_{T_j\geq T_i} h_0(t)\exp[g(\mathbf{x}_j)]} = \frac{\exp[g(\mathbf{x}_i)]}{\sum_{T_j\geq T_i}\exp[g(\mathbf{x}_j)]}. \end{aligned} \tag{12}$$

By considering all individuals, the partial likelihood is estimated as

$$L = \prod_i (\frac{\exp[g(\mathbf{x}_i)]}{\sum_{j\in\sigma_i}\exp[g(\mathbf{x}_j)]})^{D_i}. \tag{13}$$

Note that no term of the baseline hazard is included in the likelihood expression due to the time-invariant risk assumption. This is a *partial likelihood function* as the covariates can be estimated without knowing baseline hazard over time. According to the partial likelihood, we can optimize the model parameters by using negative partial log-likelihood as the loss function for training a machine learning model.

$$\text{loss} = -\sum_i D_i[g(\mathbf{x_i}) - \log(\sum_{j\in\sigma_i}\exp[g(\mathbf{x}_j)])] \tag{14}$$

The parameterization of the relative risk function $\exp[g(\mathbf{x})]$ is rather straightforward by using a NN to replace linear function $\beta^T\mathbf{x}$. It is theoretically justified since the likelihood function can be used as a loss function in machine learning models and naturally incorporates non-proportional hazards. Faraggi and Simon proposed an approach to modeling censored survival data using the simple feed-forward NN as a basis for the non-proportional hazards model [33]. In 1994, Liestbl et al. [34] firstly proposed the idea to train the NN-based COX model by backpropagation, an efficient algorithm in machine learning to find model weights. However, because the practice of NNs was not as developed as it is today, they failed to demonstrate improvements beyond the CPH model [35,36].

In 2004, Franco et al. proposed a survival model based on NNs for the prognosis of outcome in patients with primary breast cancer [37]. Improvement of prognosis is highlighted and a summary of successful applications of NNs is included together with an analysis of their advantages over the standard statistical tools by using backpropagation. Katzman et al. extended from previous work by proposing multi-layer perceptrons, a NN with a deeper structure to parameterize the risk function, named as DeepSurv [21]. They found that the proposed network outperformed the CPH model in predictive ability on survival data with linear and nonlinear risk functions based on breast cancer and heart attacker datasets. Ching et al. [38] developed a Cox-nnet, a single layer NN model, for prognosis prediction on the cancer-related datasets.

3.2. Time-variant survival models

Parameterization of the relative risk function by a NN enables the modeling of the complex relationship between risks and covariates. However, the assumption for time-invariant hazard remains, which implies that hazards of any pair of individuals remain proportional over time, as shown in Eq. (15). This assumption is not often realistic, especially when covariate change over time [39] or risk function is time-dependent [40].

$$\frac{h(t|\mathbf{x}_1)}{h(t|\mathbf{x}_2)} = \frac{\exp[g(\mathbf{x}_1)]}{\exp[g(\mathbf{x}_2)]} = \text{const} \tag{15}$$

Continuing with the semi-parametric form of the Cox model, Kvamme et al. [19] parameterized the relative risk function by discretizing the function and parameterizing the function by both covariates and time, $\exp[g(t,\mathbf{x})]$, named as *CoxTime*, which changes the hazard function and loss function as,

$$h(t|\mathbf{x}) = h_0(t)\exp[g(t,\mathbf{x})], \tag{16}$$

$$\text{loss} = -\sum_i D_i[g(T_i,\mathbf{x_i}) - \log(\sum_{j\in\sigma_i}\exp[g(T_j,\mathbf{x}_j)])]. \tag{17}$$

An alternative approach is to use a fully parametric survival model with a discretized hazard or survival function [41]. Assume that time span $\mathbb{T}$ is a set of positive time point τ_j's, $\mathbb{T}=\{\tau_1,\ldots,\tau_m\}$, we now use a probability mass function to represent the probability that the failure happens at time τ_j (time period j).

$$f(\tau_j) = \mathbb{P}(T^* = \tau_j) \tag{18}$$

The survival function and the hazard rate then become

$$S(\tau_j) = \mathbb{P}(T^*>\tau_j) = \sum_{k\geq j} f(\tau_k), \tag{19}$$

$$h(\tau_j) = \mathbb{P}(T^*=\tau_j|T^*>\tau_{j-1}) = \frac{\mathbb{P}(T^*=\tau_j)}{\mathbb{P}(T^*>\tau_{j-1})}. \tag{20}$$

By substituting Eqs. (18) and (19) and let $S(\tau_0)=1$, the survival function can be expressed as a function of the hazard rate,

$$f(\tau_j) = h(\tau_j)S(\tau_{j-1}) \tag{21}$$

$$S(\tau_j) = [1-h(\tau_j)]S(\tau_{j-1}) = \prod_{k=1}^{j}[1-h(\tau_k)]. \tag{22}$$

We denote the discretized hazard term as $\tilde{h}(k)$. The survival function in Eq. (6) is discretized as follows

$$S(t_j|\mathbf{x}) = \exp\{-\sum_{k=1}^{j}\tilde{h}(k)\}, \tag{23}$$

where, the hazard rate $h_j = 1-e^{-\tilde{h}_j}$. The likelihood function in Eq. (10) can be converted to a loss function that minimizes the mean negative log-likelihood, namely:

$$\text{loss} = -\frac{1}{n}\sum_{i=1}^{n}(d_i \log[f(t_i|\mathbf{x}_i)] + (1-d_i)\log[S(t_i|\mathbf{x}_i)]), \tag{24}$$

$$= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\kappa(t_i)}(y_{ij}\log[h(\tau_j|\mathbf{x}_i)] + (1-y_{ij})\log[1-h(\tau_j|\mathbf{x}_i)]). \tag{25}$$

Here, $y_{ij} = 1\{t_i = \tau_j, d_i = 1\}$ and $\kappa(t)$ is the index of the discrete time t. To transform into a deep survival model, the idea is to use a NN to estimate parameters for hazard function/survival function of each time τ_j according to feature $\mathbf{x}$. The NN has n-dimensional output (n is the length of lifetime) representing a hazard rate for each time period.

Parameterization of the hazard function

The sigmoid function is often used in parameterizing the hazard function as its value is in the range of $[0,1]$. Denote ϕ_j as the jth output of the NN, the hazard function becomes

$$h(\tau_j|\mathbf{x}) = \frac{1}{1+\exp[-\phi_j(\mathbf{x})]}. \tag{26}$$

In this approach, Brown et al. [41] proposed a discrete-time survival model using NNs. This model can easily be trained with SGD and used to predict the survival function given a set of fixed time periods. In 1998, Biganzoli et al. [42] used the feedforward NN to parameterize the hazard rate modeled in discrete time, also referred to as the time-encoded method by [21]. Recently, researchers seek to use deep networks to analyze the complex relationship between different co-variates. Gensheimer and Narasimhan proposed a *nnet-survival* [43], a discrete-time survival model based on parameterization of hazard rate, and using mini-batch stochastic gradient descent (SGD) in model training. The model is well-performed on both simulated and real-world datasets. They experimented with image datasets by incorporating convolutional units. Deep convolutional neural network has been well recognized for improving the prognostics accuracy [44]. Zhang et al. [45] proposed a CNN-based Survival (CNN-Survival) model with a modified loss function. The proposed model outperforms conventional radiomics and CPH-based prognosis models given the medical imaging datasets.

Parameterization of the PMF

In this case, we denote $\Phi(\mathbf{x}) = \sum_{k=1}^{m}\exp[\phi_k(\mathbf{x})]$ and parameterize the PMF as follows

$$f(\tau_j|\mathbf{x}) = \frac{\exp[\phi_j(\mathbf{x})]}{1+\Phi(\mathbf{x})}, \tag{27}$$

where, $\sum_{k=1}^{m} f(\tau_k|\mathbf{x}) + S(\tau_m|\mathbf{x}) = 1.$ and $\phi_{m+1}(\mathbf{x}) = 0$. Given the parameterization of the PMF, the survival function are formalized as follows

$$S(\tau_j|\mathbf{x}) = \sum_{k=j+1}^{m} f(\tau_k|\mathbf{x}) + S(\tau_m|\mathbf{x}) = \frac{\sum_{i=j+1}^{m+1}\exp[\phi_i(\mathbf{x})]}{1+\Phi(\mathbf{x})}, \tag{28}$$

$$\frac{f(\tau_j|\mathbf{x})}{S(\tau_m|\mathbf{x})} = \exp[\phi_j(\mathbf{x})]. \tag{29}$$

Deephit is a deep survival model following this parameterization approach, which also accounts for the competing risks [46]. Authors considered the ranking loss and combined it with negative log-likelihood by weighted sum. Multi-task logistic regression (MTLR) model was proposed by Yu et al. [47]. They reformed the likelihood function as the probability of predicting a correct sequence of binary labels indicating the event outcome of a series of time periods.

Giunchiglia et al. [48] proposed a recurrent NN-based survival model, named RNN-SURV, which achieves the highest C-index on two large and three small publicly available datasets. Lombardo et al. [49] implemented a 2D-CNN and a 3D-CNN with the logistic hazard function of distant metastasis occurrence in head and neck cancer patients. Results show that image-based deep learning models are robust for analysis of metastasis occurrence and could be used for treatment personalization.

Although the loss functions of both parameterization methods are equivalent, see. Eqs. (24) and (25), different parameterization approaches may result in different performance. Condition of Eqs. (27) and (28) in parameterization on survival function ensures the probability of failure sum up to 1, which indirectly helps the model to learn the reasonable solutions. As a summary, we compared the common deep survival models and the associated parameterization approaches in Table 1.

Despite the successful implementation of NN-based models in survival analysis, there is a lack of a comprehensive model with the ability to analyze long-sequence time-series co-variates. Time-series signals require the model to extract representative features based on the temporal correlations of every single co-variate and the cross-correlation between co-variates. More importantly, the ability to memorize long-sequence of co-variates are critical, and how to bridge the long-sequence co-variates and long-sequence of survival prediction remains an unsolved problem. In this study, we propose seq2surv as the first attention-based deep survival model to enhances the model's ability to extract features from long time series and effectively translate features into reliability estimates, which shows its superior performance in various reliability evaluation criteria and datasets.

4. Problem statement

Given the time-series features (covariates) of each individual, our goal is to forecast another time-series output representing the reliability performance at each future time period. Extensive research has been made in time-series modeling. Although the well-known models, including the auto-regressive moving average (ARMA) model, kernel methods, and ensemble methods, have shown their effectiveness in many real-world applications, most of these approaches employ a pre-defined nonlinear form, which struggles to capture the true underlying patterns and to estimate survival outcomes appropriately [50]. Furthermore, as the reliability performance for the entire lifetime is often desired, it requires the model capable of making multi-step predictions, which becomes another challenge of using time-series forecasting models due to the error accumulation [51].

Sequence-to-sequence (seq2seq) is one of the popular models that directly maps an input of sequence to an output of sequence, i.e., generate multi-step predictions. The main advantage of seq2seq models over time-series models is the ability to bridge two different sequences in terms of size and type. Seq2seq has proven to be very successful in many disciplines and has largely replaced traditional models in machine translation [52]. We make an analogy between machine translation and survival analysis of time-series features. Instead of learning how to translate a sentence from one language to another (as in machine translation), our survival model tends to translate a 'language' of operating status into another 'language' of reliability performance.

Until recently, most of the deep learning models for sequences were built on recurrent neural networks (RNNs) [53] and encoder–decoder networks with RNN units [54]. However, the performance of RNN-based networks deteriorates with the increasing length of input sequence [50], which limits the model's ability to analyze the correlations and identify the representative features. In the analysis of smart manufacturing systems and connected vehicles, this could be a concern, since we usually expect to make predictions based upon relatively long sensor inputs, e.g., weeks of machine operation or years of driver behavior. We address this problem by leveraging the attention mechanism and seek to improve the performance of survival analysis.

Table 1
A summary of deep survival models.

Model	Likelihood function	Param. model	Covar. function	Loss function
Prop.-Cox	$\mathbb{P}(t \le T^* < t + \Delta t, D = d)$	$h(t\mid\mathbf{x}) = h_0(t)\exp[\theta^T\mathbf{x}]$	$\theta^T(\mathbf{x})$	$-\sum_i D_i[\theta^T\mathbf{x_i} - \log\sum_{j\in\sigma_i}\exp[\theta^T\mathbf{x}_j]]$
Deepsurv/ Cox-nnet	$\mathbb{P}(t \le T^* < t + \Delta t, D = d)$	$h(t\mid\mathbf{x}) = h_0(t)\exp[g(\mathbf{x})]$	$g(\mathbf{x})$	$-\sum_i D_i[g(\mathbf{x_i}) - \log\sum_{j\in\sigma_i}\exp[g(\mathbf{x}_j)]]$
Cox-time	$\mathbb{P}(t \le T^* < t + \Delta t, D = d)$	$h(t\mid\mathbf{x}) = h_0(t)\exp[g(\mathbf{x})]$	$g(\mathbf{x}, \mathbf{t})$	$-\sum_i D_i[g(T_i, \mathbf{x_i})-$ $\log(\sum_{j\in\sigma_i}\exp[g(T_j, \mathbf{x}_j)])]$
Nnet-Survival	$\mathbb{P}(T = t, D = d)$	$h(\tau_j\mid\mathbf{x}) = \frac{1}{1+\exp[-\phi_j(\mathbf{x})]}$	$\phi_j(\mathbf{x})$	$-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\kappa(t_i)}(y_{ij}\log[h(\tau_j\mid\mathbf{x}_i)]$ $+(1-y_{ij})\log[1-h(\tau_j\mid\mathbf{x}_i)]])$
CNN-Survival	$\mathbb{P}(T = t, D = d)$	$h(\tau_j\mid\mathbf{x}) = \frac{1}{1+\exp[-\phi_j(\mathbf{x})]}$	$\phi_j(\mathbf{x})$ (CNN)	$-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\kappa(t_i)}(y_{ij}\log[h(\tau_j\mid\mathbf{x}_i)]$ $+(1-y_{ij})\log[1-h(\tau_j\mid\mathbf{x}_i)]])$
PMF	$\mathbb{P}(T = t, D = d)$	$\frac{S(\tau_j\mid\mathbf{x})}{S(\tau_m\mid\mathbf{x})} = \exp[\phi_j(\mathbf{x})]$	$\phi_j(\mathbf{x})$	$-\frac{1}{n}\sum_{i=1}^{n}[d_i\log f(t_i\mid\mathbf{x}_i)$ $+(1-d_i)\log S(t_i\mid\mathbf{x}_i)]$
Deep Hit	$\mathbb{P}(T = t, D = d)$	$\frac{S(\tau_j\mid\mathbf{x})}{S(\tau_m\mid\mathbf{x})} = \exp[\phi_j(\mathbf{x})]$	$\phi_j(\mathbf{x})$	$-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\kappa(t_i)}(y_{ij}\log[h(\tau_j\mid\mathbf{x}_i)]$ $+(1-y_{ij})\log[1-h(\tau_j\mid\mathbf{x}_i)])$ $+\sum_{i,j}D_i 1_{T_i<T_j}\exp(\frac{\bar{S}(T_i\mid\mathbf{x}_i)-\bar{S}(T_j\mid\mathbf{x}_j)}{\sigma}))]$
MTLR	$PY = (y_1, \dots, y_m)$	$\frac{S(\tau_j\mid\mathbf{x})}{S(\tau_m\mid\mathbf{x})} = \exp\sum_{k=j}^{m}\psi_k(\mathbf{x})$	$\psi_k(\mathbf{x})$	$-\frac{1}{n}\sum_{i=1}^{n}[d_i\log f(t_i\mid\mathbf{x}_i) + (1-$ $d_i)\log S(t_i\mid\mathbf{x}_i)]$
RNN-SURV	$\mathbb{P}(T = t, D = d)$	$\frac{S(\tau_j\mid\mathbf{x})}{S(\tau_m\mid\mathbf{x})} = \exp[\phi_j(\mathbf{x})]$	$\phi_j(\mathbf{x})$ (LSTM)	$-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\kappa(t_i)}(y_{ij}\log[h(\tau_j\mid\mathbf{x}_i)]$ $+(1-y_{ij})\log[1-h(\tau_j\mid\mathbf{x}_i)])$ $+\sum_{i,j}D_i 1_{T_i<T_j}[1+\frac{\log\sigma(\bar{S}(T_j\mid\mathbf{x}_j)-\bar{S}(T_i\mid\mathbf{x}_i))}{\log 2}]])$

5. Seq2surv model

In this section, we propose a seq2surv model with an attention mechanism to learn complex patterns in the time-series signals and to generate multi-step survival predictions with the ability to filter out irrelevant noise by encoder–decoder structure. The model is designed to bridge sequences of time-series features to a sequence of the survival probability estimates for the entire lifetime. Specifically, the architecture of seq2seq model is leveraged to parameterize the survival function with a new loss function designed for time-series features.

We assume the W time periods of the most recent features are most representative of the reliability performance. So the time-series feature of type k in a selected period of times are $\mathbf{x}_k(\tau_j) = [x_k(\tau_{k,j-W}), \dots, x_k(\tau_{k,j})]$, where $x_k(\tau_{k,j})$ is the value of feature k at time $\tau_{k,j}$. Note that, different features do not need to be in the same time scales. Considering K features, we denote the time-series inputs as $\mathbf{X}_{\tau_j} = [\mathbf{x}_1(\tau_j), \dots, \mathbf{x}_K(\tau_j)]$.

Then, we extend from the previous parameterization, Eqs. (27) and (28), to time-series features. The probability mass function and the survival function are as follows

$$f(\tau_j|\mathbf{X}_{\tau_j}) = \frac{\exp[\phi_j(\mathbf{X}_{\tau_j})]}{1+\Phi(\mathbf{X}_{\tau_j})}, \tag{30}$$

$$S(\tau_j|\mathbf{X}_{\tau_j}) = \frac{\sum_{i=j+1}^{m+1}\exp[\phi_i(\mathbf{X}_{\tau_j})]+1}{1+\Phi(\mathbf{X}_{\tau_j})}, \tag{31}$$

where $\phi_{m+1}(\mathbf{X}_{\tau_j}) = 0$ and $\Phi(\mathbf{X}_{\tau_j}) = \sum_{k=1}^{m}\exp[\phi_k(\mathbf{X}_{\tau_j})]$. Note that condition $\sum_{k=1}^{m} f(\tau_k|\mathbf{X}_{\tau_j}) + S(\tau_m|\mathbf{X}_{\tau_j}) = 1$ still reserves for the parameterization. To ensure features have the same length, we pad the features by adding zeros as placeholders to the shorter length vectors. Each individual is described by the recent behaviors rather than the static features used in other models. Hence, the shape of the survival function evolves with real-time updates in features. The loss function of the model becomes

$$\text{loss} = -\frac{1}{n}\sum_{i=1}^{n}[d_i\log f(T_i|\mathbf{X}_{T_i}) + (1-d_i)\log S(t_i|\mathbf{X}_{T_i})]. \tag{32}$$

Given the loss function, the high-level learning scheme contains two aspects including feature extraction and survival function prediction. Depending on the learning scheme, feature extraction is classified as 1) local approach — the free parameters of the model are learned individually for each series, and 2) global approach — the parameters are learned jointly on all-time series. Due to interconnections between the covariates, we target to develop our learning model in a global approach.

Given extracted features, the survival function prediction can be recurrent or concurrent. In recurrent structures, each prediction of the future horizon is based on the predicted values of the previous time period. One downside of this prediction is that the error may accumulate from a series of predictions, which often relies on the teacher forcing method to recursively supply the true values as inputs during training. Models with concurrent structures directly map learned features to a complete sequence of future values for a range of steps. The evaluation time is much shorter, especially for long-sequence prediction.

Following this idea, we design a novel seq2seq-based parameterization function ϕ_j to predict the future sequence of the survival probability based on the jointly learned features. The resulted seq2surv model is shown in Fig. 1. The general architecture of our model follows an encoder–decoder structure: the encoder maps the input sequence to a latent state vector and the decoder generates the output sequence from the latent state vector. Encoders and decoders can, in principle, be any NN architecture and are often recognized by their superior ability in denoising and feature reduction [55]. We also incorporate the attention mechanism to address the limitations of the long-sequence feature extraction of RNNs.

5.1. Encoder

The encoder is built by a bidirectional RNN, which goes over the time-series features in two directions. A forward RNN scans over time series from left to right (shown in dark blue), and a backward RNN takes in the time series from right to left (shown in light blue). Bidirectional RNNs have both look-ahead and look-behind ability, which helps to extract complex features, including trend, periodicity and patterns, from collections of high dimension irregular time series [51]. We use a gated recurrent unit (GRU) to take the sequential input $x(t)$ as follows

$$z_t = \sigma(W_z[h_t - 1, x(t)]), \tag{33}$$

$$r_t = \sigma(W_r[h_t - 1, x(t)]), \tag{34}$$

$$\tilde{h}_t = \tanh(W_h[r_t * h_{t-1}, x(t)]), \tag{35}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, \tag{36}$$

where W_z, W_r and W_h are weights to be learned during the training of the network. h_t is the output of the hidden state. z_t and r_t

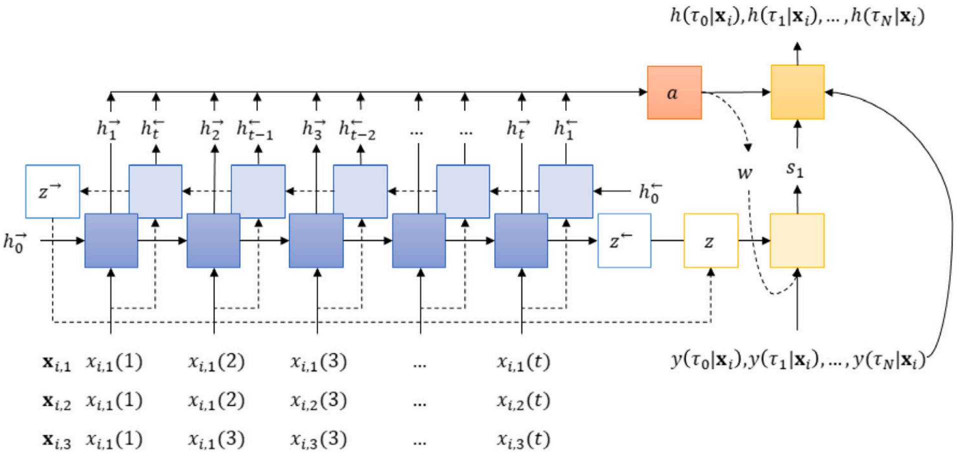
$h(\tau_0|\mathbf{x}_i), h(\tau_1|\mathbf{x}_i), \dots, h(\tau_N|\mathbf{x}_i)$
a
$h_1^{\rightarrow}$ $h_t^{\leftarrow}$ $h_2^{\rightarrow}$ $h_{t-1}^{\leftarrow}$ $h_3^{\rightarrow}$ $h_{t-2}^{\leftarrow}$ ... ... $h_t^{\rightarrow}$ $h_1^{\leftarrow}$
w
s_1
$z^{\rightarrow}$
$h_0^{\leftarrow}$
$h_0^{\rightarrow}$
$z^{\leftarrow}$
z
$\mathbf{x}_{i,1}$ $x_{i,1}(1)$ $x_{i,1}(2)$ $x_{i,1}(3)$ ... $x_{i,1}(t)$
$\mathbf{x}_{i,2}$ $x_{i,1}(1)$ $x_{i,1}(2)$ $x_{i,2}(3)$ ... $x_{i,2}(t)$
$\mathbf{x}_{i,3}$ $x_{i,1}(1)$ $x_{i,1}(3)$ $x_{i,3}(3)$ ... $x_{i,3}(t)$
$y(\tau_0|\mathbf{x}_i), y(\tau_1|\mathbf{x}_i), \dots, y(\tau_N|\mathbf{x}_i)$

Fig. 1. The network architecture of the seq2surv model.

are update gate vector and reset gate vector. We use ← and → to annotate the state from forward GRU and backward GRU. So the outputs from all hidden layers becomes $\mathbf{h}_t^{\leftarrow} = [h_T^{\leftarrow}, \dots, h_1^{\leftarrow}]$ and $\mathbf{h}_t^{\rightarrow} = [h_1^{\rightarrow}, \dots, h_T^{\rightarrow}]$, we firstly created concatenated the hidden states together as $\mathbf{h}_t = [(h_T^{\leftarrow}, h_1^{\rightarrow}), \dots, (h_1^{\leftarrow}, h_T^{\rightarrow})]$. All hidden states of the encoder are then aggregated as $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T]$.

We used the context vector z to hold information for the current state which is passed down to the decoder, by implementing a linear layer g_1 and a tanh activation function as follows

$$z = \tanh[g_1(h_T^{\leftarrow}, h_T^{\rightarrow})]. \tag{37}$$

5.2. Attention mechanism

In a regular seq2seq network, only context vector z is used for the decoder to learn and predict the survival outcomes. To address the challenges in memorizing long-sequence inputs, the attention mechanism has been gaining importance especially in seq2seq learning [56]. In the attention mechanism, the main idea is to weigh all outputs of hidden states to dynamically highlight relevant features of the input data. We incorporate the attention mechanism to make deep survival models focus differently on features at each period of lifetime.

The attention mechanism is a linear layer followed by a softmax function. In the seq2surv model, the attention layer takes all hidden states of decoder, s_{t-1}, and stacked forward and backward hidden states from the encoder $\mathbf{H}_t$ as inputs, and outputs an attention vector a_t. a_t has the same length of the time-series features and its value measures the feature importance. Denoting a linear layer as g_2, the attention mechanism is formalized as

$$\tilde{a}_t = \tanh(g_2(\mathbf{s}_{t-1}, \mathbf{H}_t)), \tag{38}$$

$$a_t = \phi(\tilde{a}_t), \tag{39}$$

where ϕ is the softmax function to ensure values of all outputs are between 0 and 1. The weights of the linear layer g_2 are trained along with encoder and decoder.

5.3. Decoder

The decoder takes a weighted sum of the hidden states of the encoder to predict survival probabilities of all time periods. The bi-directional structure of RNN endows each hidden state of the encoder with context information, i.e., how does feature value change as compared to values in the past and future. Inputs to decoder $\mathbf{w}_t$ is a weighted sum of encoder hidden states $\mathbf{H}_t$ by computed attention a_t, i.e.

$$\mathbf{w}_t = a_t \mathbf{H}_t. \tag{40}$$

The weighted outputs of hidden states, $\mathbf{w}_t$ and previous decoder hidden state, s_{t-1} are then passed together into decoder RNN to compute the current hidden state s_t, namely

$$s_t = \mathrm{GRU}([\mathbf{w}_t, s_{t-1}]). \tag{41}$$

We then pass $\mathbf{w}_t$ and s_t through the linear layer, g_3, to make predictions of survival probabilities for the entire life span $f(t|\mathbf{X}_T)$, namely

$$f(t|\mathbf{X}_T) = g_3(\mathbf{w}_t, s_t). \tag{42}$$

Survival predictions are evaluated by the loss function in Eq. (32) to refine weights in the RNNs and attention mechanism by using the backpropagation algorithm.

6. Results

In this section, we compared our seq2surv model with the aforementioned deep survival models in the simulated and NASA turbofan engine datasets. Evaluation metrics, including SEP, Concordance index, and Brier score, are adopted for model comparison (detailed information can be found in Appendix B).

6.1. Simulated dataset

To empirically evaluate our proposed methodology, we simulated a collection of time series, describing the status of a mechanical part, with failure time generated in 100-time periods. These experiments are designed to verify that the proposed model behaves as expected. Each part, i.e., individual, has attributes of failure time, failure result, and time-series features. Each feature is a simulated 200-time-step and 10-type time series. Good parts and defective parts are differentiated by abnormal trends or patterns. All parts are separated into 4 risk strata based on the survival time. For risk strata of good parts $\mathbf{G}$, time series are generated from a uniform distribution, namely,

$$x_{i,k}(t) = U(-1, 1), \forall k, t, i \in \mathbf{G}. \tag{43}$$

Good individuals are censored, i.e., parts remain healthy (not failed) by the end of the observation period. Three failure modes of defective parts are simulated. We define the risk strata of these different failure modes as $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$.

In the first failure mode, abnormal patterns are created in feature 1 and repeated -1 and 1 occurring in a period of 5. The survival time is fixed as 24.

$$x_{i,1}(5n + a_1) = -1, \quad \forall 5n + a_1 \in T, i \in \mathbf{P}_1, \tag{44}$$

$$x_{i,1}(5n + a_1 + 2) = 1, \quad \forall 5n + a_1 + 2 \in T, i \in \mathbf{P}_1, \tag{45}$$

Table 2
Comparison of model performance in terms of C-index and IBS.

	seq2surv	AE-net	NN-net	PMF	Deephit	MTLR	Linear
C-index	1.0	0.938	0.957	0.940	0.961	0.957	0.782
IBS	0.009	0.019	0.012	0.017	0.019	0.012	0.084

where, a_1 is phase shift, a random integer between 1 and 5. The first failure mode is attributed to periodic signals in the first feature. The random phase shift makes failure independent on time but dependent on the pattern.

In the second failure mode, the abnormal patterns existing in both feature 1 and feature 2, and occurring in a period of 8. The survival time is fixed as 48.

$$x_{i,1}(8n + a_2) = -1, \quad \forall 8n + a_2 \in T, i \in \mathbf{P}_2, \tag{46}$$

$$x_{i,2}(8n + a_2) = 1, \quad \forall 8n + a_2 \in T, i \in \mathbf{P}_2, \tag{47}$$

where, a_2 is phase shift, a random integer between 1 and 8. The second failure mode is attributed to patterns co-existing in features 1 and 2. Note that -1 in feature 1 may originate from both first and second failure modes. In order to make an accurate prediction, the model needs to distinguish different '-1's from analyzing the correlations among different feature types.

The third failure mode is an increasing trend pattern in the third feature, repeated in a period of 12. The survival time is fixed as 72.

$$x_{i,1}(12n + a_3) = -1, \quad \forall 12n + a_3 \in T, i \in \mathbf{P}_3, \tag{48}$$

$$x_{i,2}(12n + a_3 + 1) = -0.5, \quad \forall 12n + a_3 + 1 \in T, i \in \mathbf{P}_3, \tag{49}$$

$$x_{i,3}(12n + a_3 + 2) = 0.5, \quad \forall 12n + a_3 + 2 \in T, i \in \mathbf{P}_3, \tag{50}$$

$$x_{i,4}(12n + a_3 + 3) = 1, \quad \forall 12n + a_3 + 3 \in T, i \in \mathbf{P}_3, \tag{51}$$

where, a_3 is phase shift, a random integer between 1 and 12. This failure mode is designed to evaluate the ability in memorizing the patterns existing in a relatively long sequence, which is an increasing trend. By this experimental design, all risk strata have the same mean value of 0. The standard deviations of different risk strata are similar, where $\sigma_G = 0.577, \sigma_{\mathbf{P}_1} = 0.599, \sigma_{\mathbf{P}_2} = 0.591, \sigma_{\mathbf{P}_3} = 0.585$. Examples of features 1, 2 and 3 from each risk strata are shown in Fig. 2.

Features from all risk strata turn out to be noise most of the time with comparable values. The design of this experiment makes the correlation between feature statistics (i.e., the mean and the variance) and failure results very weak. To make accurate survival predictions, models have to differentiate patterns in different time series. Redundant features (7 out of 10 is irrelevant to failure) are designed to test the model's ability to focus on the most failure-matter features. We randomly generated 14,000 parts, with 10,000 for training and 2,000 for both validation and testing. We compare among 7 discrete-time models with time-variant risks, including seq2surv, Nn-net, PMF, Deephit, MTLR, we also designed an auto-encoder (AE), which has an encoder–decoder structure but without attention mechanism. We use the AE-based survival model to highlight the advantage of the attention mechanism.

We evaluate model performance by C-index and Brier score defined in Eqs. (B.1) and (B.2). C-index measures the ranking accuracy, i.e., the fraction of predicted survival times that are ranked correctly. Integrated Brier score (IBS) measures the integrated error in the predicted survival probability. The performance of survival models is compared in Table 2.

The proposed seq2surv model outperforms other models in terms of highest C-index, best accuracy in ranking, and lowest IBS, the most accurate survival probability estimate. The linear model performs the worst among all models due to its simple structure, which lacks the ability to capture complex patterns. Most of the failure modes in time series data cannot be determined by the value of a single feature or a certain time. The dependency on time and other features are nonlinear, which makes NN-based deep survival models become more successful.

After seq2surv, Deephit has the second highest C-index but also the worst IBS. As the ranking errors are included in the loss function, Deephit tends to adjust model weights to account more for the ranking performance than other deep survival models do. In comparison, MTLR and N-net have relatively low IBS but a poor ranking performance, as compared to Deephit. A trade-off between ranking performance and prediction accuracy of survival probability is observed in the existing deep survival models, which highlights their limitation to capture complex relationships in time series data. To highlight the bottlenecks in predictions, we visualize the outputs of three models, including linear model, vanilla NN-based model, and the proposed seq2surv model, in Fig. 3 and use the averaged risks to measure SEP of parts in four risk strata.

Each point is a part with an index marked on the x-axis and averaged risks on the y-axis. The color of the point is used to distinguish the parts from different risk strata. In the linear model, there is no clear separation between parts, i.e., all the dots are mixed up, indicating the inability of the linear model to differentiate time series using complex features. For the NN-based model, the separation between good parts and defective parts is clear. However, the separation between parts from different failure modes is vague. For the seq2surv model, the good parts and bad parts are well separated, i.e., a clear difference in the predicted risks. Separation among different failure modes is notable, where parts in failure mode 3 have the lowest risk and parts in failure mode 1 are ranked highest, which are aligned with their survival times.

6.2. NASA turbofan engine dataset

The second dataset C-MAPSS is from the NASA data repository composing of run-to-failure sensor measurements from degrading turbofan engines, which has been widely used to evaluate models for prognosis and RUL predictions [29,57]. C-MAPSS simulates a turbofan engine model at varying altitudes ranging, Mach numbers, and sea-level temperatures. Operational settings and sensor signals are recorded as time-series features. We used data record *FD001 - FD004.txt* to train and test the models. A sample of C-MAPSS dataset used in this study is shown in Table A.4.

In preparation of datasets for deep survival analysis, we need to translate many long run-to-failure signals into survival times and time-series features. As all the engines are failed, so all the samples from data are uncensored with the failure index of 1. By presuming an observation time t_i of an engine, a training sample i can be created by using a moving window with length W as follows

- Survival time: $T_i = t_f - t_i$;
- Failure result: $D_i = 1$;
- Time-series feature k: $\mathbf{x}_k(t_i) = [x_k(t_i - W), \dots, x_k(t_i)] \forall t_i \in [W, t_f]$.

A generalized procedure to prepare the training sets is shown in Fig. 4.

Although all data in this example are uncensored, i.e., failure happens before the end of observation, the data preparation process can be easily extended for the censored data, i.e., when failure is not observed. Given an engine i with censored data, the survival time becomes $T_i = t_o - t_i$ and failure index is $D_i = 0$. The rest of the preparation procedure remains the same.

With a window size of 100, we sampled 71567 samples from 709 NASA turbo engines following the proposed data preparation procedure. Compared to using all run-to-failure data of each engine as a sample, our samples are enlarged more than 100 times and the size of time-series features of each sample are the same. The proposed data preparation procedure can help the model better understand the changes of signals and enriches insights from limited and expensive experiments. We split all samples into the training set, validation set

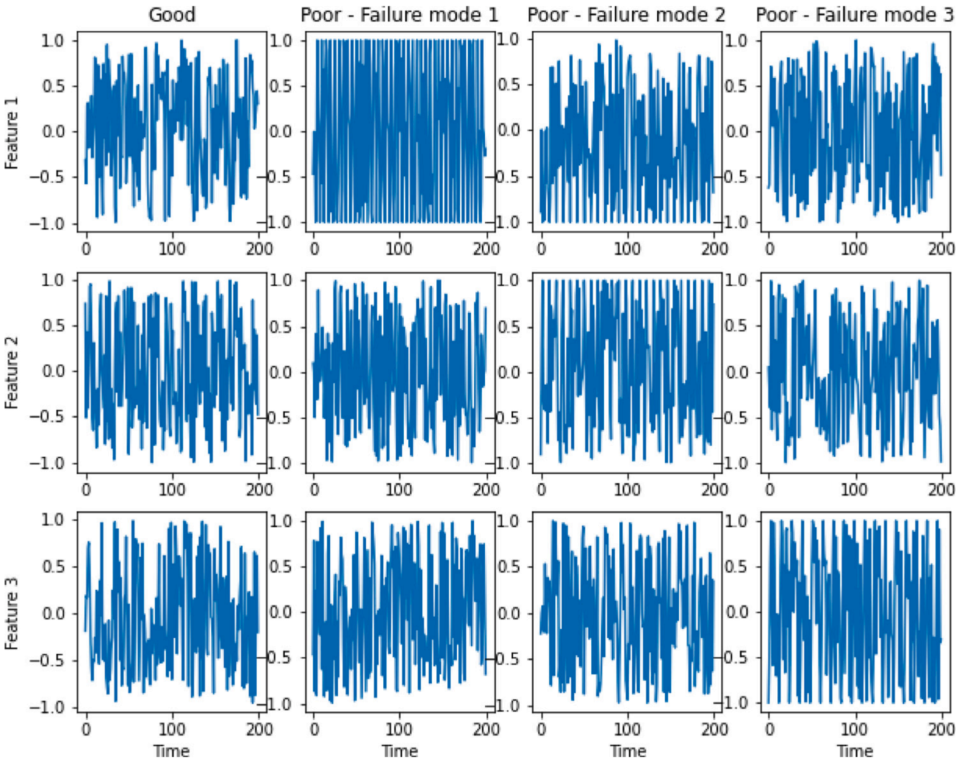

Fig. 2. Example for good parts and different defected parts.

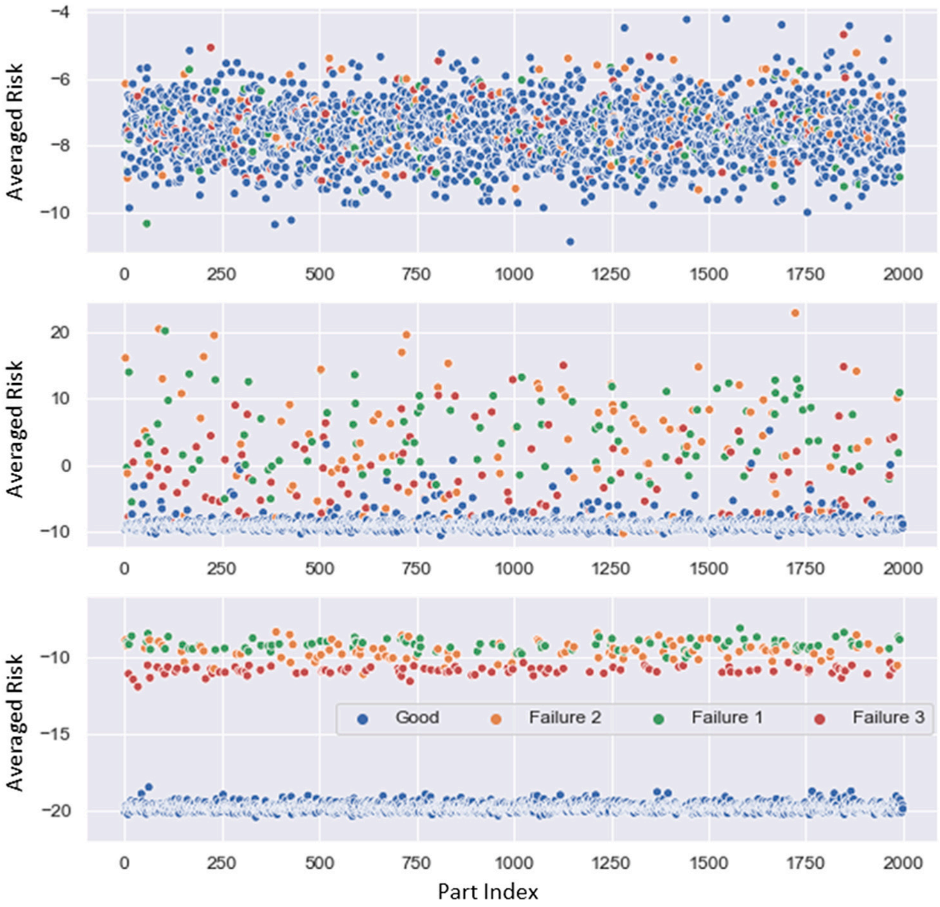

Fig. 3. Comparison of separations learned by different model: a) Linear model, b) Vanilla NN-based model, c) Seq2surv model.

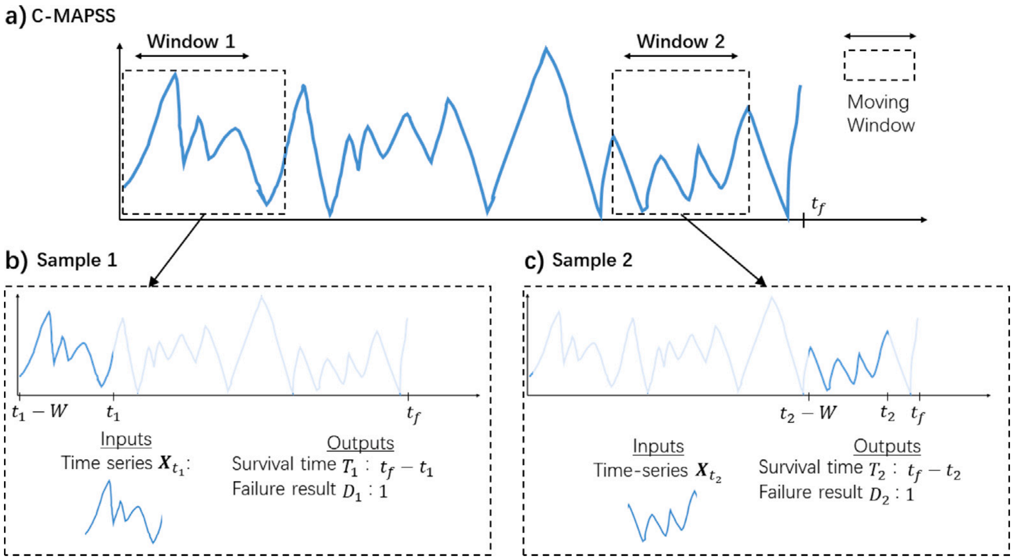

Fig. 4. A diagram that shows the feature preparation from run-to-failure sensor measurements. a) one of the sensor measurements and failure time in C-MAPSS datasets, b) sample 1 created by window 1 with calculated survival time and time-series features, c) sample 2 created by window 2 with calculated survival time and time-series features.

Table 3
Comparison of model performance in terms of C-index and IBS.

	seq2surv	AE-net	NN-net	PMF	Deephit	MTLR	Linear
C-index	0.972	0.853	0.891	0.899	0.895	0.864	0.674
IBS	0.017	0.075	0.061	0.054	0.103	0.072	0.151

and testing set with the size of 57253, 7157 and 7157, respectively. The performance of 7 discrete-time models on the C-MAPSS datasets is compared in Table 3.

As evident from the table, seq2surv model significantly outperforms existing deep survival models in the C-MAPSS datasets. Besides the seq2surv model, the model with the highest C-index is PMF, which can rank almost 90% of engines in the correct order. By using the seq2surv model, additional 7.3% engines can be ranked correctly resulting C-index = 0.972. Moreover, more than 350% of improvement in the IBS is observed by comparing the seq2surv model (0.017) and the best existing model NN-net (0.061). Fig. 5 compares Brier scores per period to confirm the improvement of performance.

Brier scores are shown as a bell shape over time. At the start and the end of the experiment, the survival probability is usually close to 0 or 1, respectively, indicating engines are either healthy or failed respectively. In such periods, models can make the correct predictions easily and keep the errors (i.e., Brier scores) low. Brier scores are the highest between the period 25 and 100, indicating the most challenging periods for models to distinguish complex time series with noise. In seq2surv model, Brier scores are always below 3 percent, indicating the superior performance of seq2surv model in feature denoising and extraction.

Time-series signals require the model to analyze not only the magnitude of features but also the context of features at each time, including how a feature changes compared to the before/after times and compared to other features. Seq2surv model is designed in such a way to take the time series data as sequential input and learn the contextual information by the bi-directional GRU, which enables each hidden state to carry the look-ahead and look-behind information. Compared to LSTM, GRU exposes the complete memory and hidden state outputs, which estimate attention accurately. Given the attention mechanism, the information from hidden states is prioritized and translated into a dense form to represent the advanced features, e.g., trends, correlations, and periodicity, in the time series. The better ability of feature extraction and memory can bridge time series and failures more effectively, thus making the survival probability prediction more accurate.

The seq2surv model has substantial potential for many practical applications. Often, there arises a need to predict not only the expected cumulative number of defective products as a function of time in service but also the changes in claims as a function of calendar time, which is important to prepare spare part inventory for repair. Following previous study [10], seq2surv model can be used to estimate the number of failed products for any calendar time period $[\tau_j, \tau_k]$ as follows

$$n(\tau_j, \tau_k) = \sum_i \sum_{t=j}^{k} n_{i,j}(S_i(\tau_t) - S_i(\tau_{t+1}))(\tau_{t+1} - \tau_t), \tag{52}$$

where $n_{i,j}$ is the indicator that a product i is in service at calendar time j ($n_{i,j} = 1$) or not ($n_{i,j} = 0$).

Accurate prediction of remaining useful life (RUL) of products and components helps businesses to identify the product defects earlier to avoid catastrophic failure. Given the outputs of seq2surv model, i.e., predicted survival probabilities $S_i(t)$, RUL of an individual i can also be approximated by

$$T_i = \arg\min_t \{S_i(t) < \bar{S}\}, \tag{53}$$

where $\bar{S}$ is a threshold survival probability defining a risky operating situation. As signals are updated in real-time, the predicted survival probability also evolves over time so as the predicted RUL. Following this idea, we selected four engines in C-MAPSS datasets to illustrate how to transform the real-time signals into the RULs in Fig. 6.

The predicted RULs converge to the actual time in all four engines, i.e., the predicted RUL at actual failure time is 0. Specifically, Fig. 6a and b show engines with constant degradation, where RULs are decreasing almost linear with the observation time. In Fig. 6c, the fluctuating RULs are calculated from the real-time signals, indicating varying engine conditions. Fig. 6d shows an engine with the best initial condition (longest RUL), but its condition deteriorates quickly in the first 20 periods and failed around time 80. By comparing the changes in RULs, one may find clues of when an engine starts going wrong. Like the fuel meter of a vehicle (measuring the remaining fuel level), seq2surv model can translate sensor signals to reliability performance in real-time for every monitored component - a ’life meter’.

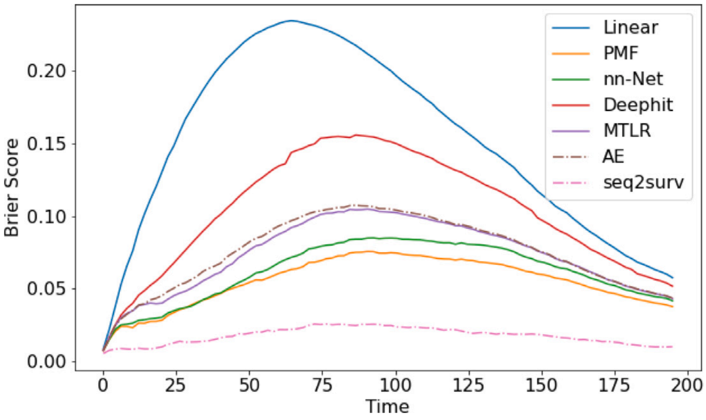

Fig. 5. Comparison of Brier scores in the C-MAPSS test set between the seq2surv model and other deep survival models in the discrete-time category.

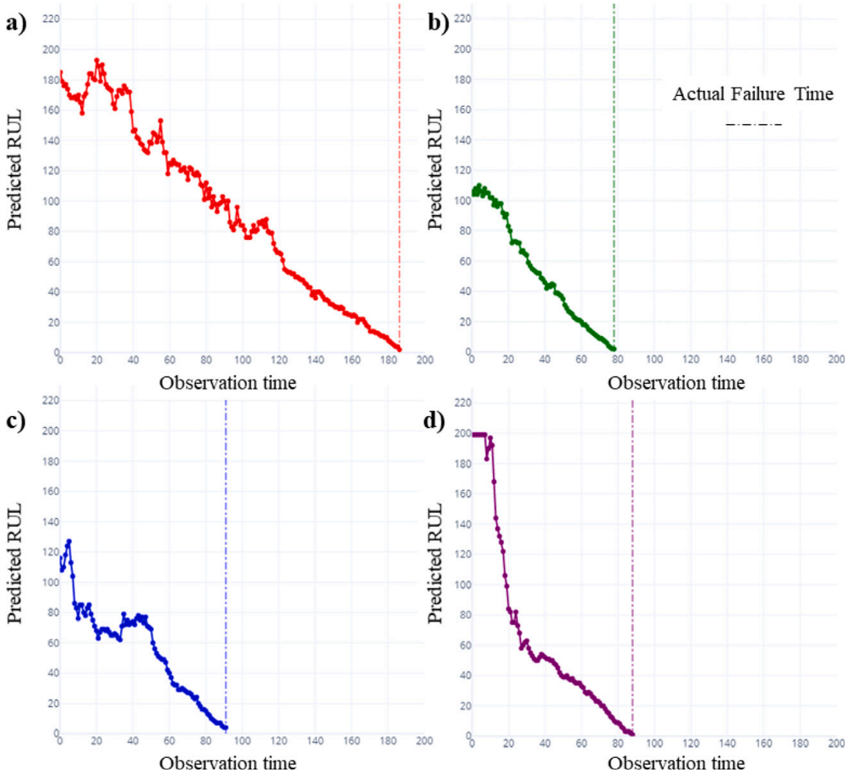

Fig. 6. Predicted RULs for four different engines in the C-MAPSS datasets.

7. Conclusions and future work

This paper proposes a novel system prognostic tool, seq2surv, to analyze the relationship between lifetime reliability performance and complex real-time signals, which are widely collected in Industry 4.0 and the Internet of Things. Seq2surv model is designed for the reliability analysis of time-series signals, which commonly exist in smart manufacturing systems and autonomous/connected vehicles. Seq2surv leverages the merits from the seq2seq model (commonly used for machine translation) and from the classic survival model to improve the ability to extract features from long time series and prognosticating the system conditions for the entire lifeline.

Analogously to machine translation, seq2surv can translate a sequence of input features, describing the system operating and environmental information, into another sequence of outputs, describing the system reliability performance. Thanks to the attention mechanism, the encoder is able to analyze and store a long sequence of signals as dense features. The decoder translates the extracted features into reliability estimates using a concurrent structure. Given the simulated time series with similar statistics, the seq2surv model can effectively separate the time series originated from different risk strata by leveraging the trend, seasonality, and complex relationship among different features. Our model shows the best performance in both C-index and IBS as compared to all the existing deep survival models in both simulated and NASA turbo engine datasets, indicating a superior ability in ranking an individual's survivability and in predicting the survival probabilities.

Promising future research directions include deriving the explanations of the system prognosis results. Often multiple failure modes exist in the same product. Identifying the failure mode along with its expected failure time helps to avoid or resolve failures. Moreover, identifying factors and patterns in the signals that contribute to the failure benefits the root cause analysis and improves product design and manufacturing processes.

Table A.4
Sample of C-MPASS data. o is short for the operational settings and s is short for the sensor measurement.

Index	o_1	o_2	o_3	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}	s_{21}
1	0.0019	-0.0003	100	518.67	642.15	1591.82	1403.14	14.62	21.61	553.75	2388.04	9044.07	1.3	47.49	522.28	2388.07	8131.49	8.43	0.03	392	2388	100	39	23.42
2	-0.0043	0.0003	100	518.67	642.35	1587.99	1404.2	14.62	21.61	554.26	2388.08	9052.94	1.3	47.27	522.42	2388.03	8133.23	8.42	0.03	390	2388	100	38.95	23.34
3	0.0007	0	100	518.67	642.35	1582.79	1401.87	14.62	21.61	554.45	2388.11	9049.48	1.3	47.13	522.86	2388.08	8133.83	8.37	0.03	392	2388	100	38.88	23.37
4	-0.0019	-0.0002	100	518.67	642.37	1582.85	1406.22	14.62	21.61	554	2388.06	9055.15	1.3	47.28	522.19	2388.04	8133.8	8.43	0.03	393	2388	100	38.9	23.4
5	-0.0043	-0.0001	100	518.67	642.1	1584.47	1398.37	14.62	21.61	554.67	2388.02	9049.68	1.3	47.16	521.68	2388.03	8132.85	8.41	0.03	391	2388	100	38.98	23.37
6	0.001	0.0001	100	518.67	642.48	1592.32	1397.77	14.62	21.61	554.34	2388.02	9059.13	1.3	47.36	522.32	2388.03	8132.32	8.4	0.03	392	2388	100	39.1	23.38
7	-0.0034	0.0003	100	518.67	642.56	1582.96	1400.97	14.62	21.61	553.85	2388	9040.8	1.3	47.24	522.47	2388.03	8131.07	8.41	0.03	391	2388	100	38.97	23.31
8	0.0008	0.0001	100	518.67	642.12	1590.98	1394.8	14.62	21.61	553.69	2388.05	9046.46	1.3	47.29	521.79	2388.05	8125.69	8.37	0.03	392	2388	100	39.05	23.41
9	-0.0033	0.0001	100	518.67	641.71	1591.24	1400.46	14.62	21.61	553.59	2388.05	9051.7	1.3	47.03	521.79	2388.06	8129.38	8.43	0.03	393	2388	100	38.95	23.47
10	0.0018	-0.0003	100	518.67	642.28	1581.75	1400.64	14.62	21.61	554.54	2388.05	9049.61	1.3	47.15	521.4	2388.01	8140.58	8.43	0.03	392	2388	100	38.94	23.48
11	0.0016	0.0002	100	518.67	642.06	1583.41	1400.15	14.62	21.61	554.52	2388.09	9049.37	1.3	47.18	521.8	2388.02	8134.25	8.39	0.03	391	2388	100	39.06	23.37
12	-0.0019	0.0004	100	518.67	643.07	1582.19	1400.83	14.62	21.61	553.44	2388.12	9046.82	1.3	47.38	521.85	2388.08	8128.1	8.42	0.03	393	2388	100	38.93	23.28
13	0.0009	0	100	518.67	642.35	1592.95	1399.16	14.62	21.61	554.48	2388.09	9047.37	1.3	47.44	521.67	2388	8134.43	8.4	0.03	393	2388	100	39.18	23.38
14	-0.0018	-0.0003	100	518.67	642.43	1583.82	1402.13	14.62	21.61	553.64	2388.11	9052.22	1.3	47.3	522.5	2388.08	8127.56	8.42	0.03	391	2388	100	38.99	23.35
15	0.0006	0.0005	100	518.67	642.13	1587.98	1404.5	14.62	21.61	553.94	2388.05	9049.34	1.3	47.24	521.49	2388.07	8136.11	8.39	0.03	392	2388	100	38.97	23.46
16	0.0002	0.0002	100	518.67	642.58	1584.96	1399.95	14.62	21.61	553.8	2388.06	9054.92	1.3	47.12	521.89	2388.04	8137.27	8.45	0.03	392	2388	100	38.81	23.33
17	-0.0031	-0.0001	100	518.67	642.62	1591.04	1396.12	14.62	21.61	554.2	2388.05	9049.55	1.3	47.21	521.76	2388.07	8132.73	8.4	0.03	392	2388	100	38.89	23.4
18	0.0032	-0.0003	100	518.67	641.79	1587.56	1400.35	14.62	21.61	554.18	2388.04	9053.99	1.3	47.4	521.89	2388.03	8129.13	8.43	0.03	391	2388	100	38.8	23.35
19	-0.0037	0.0001	100	518.67	643.04	1581.11	1405.23	14.62	21.61	554.81	2388.05	9045.9	1.3	47.22	522.07	2388.02	8129.71	8.42	0.03	392	2388	100	39.03	23.42
20	-0.0012	0.0001	100	518.67	642.37	1586.07	1398.13	14.62	21.61	554.08	2388.11	9048.15	1.3	47.15	522.42	2388.08	8134.02	8.4	0.03	392	2388	100	39.09	23.31
21	0.0002	0	100	518.67	642.77	1592.93	1400.57	14.62	21.61	553.63	2388.04	9061.21	1.3	47.24	522	2388.03	8130.41	8.4	0.03	392	2388	100	38.92	23.38
22	0.0034	-0.0003	100	518.67	642.14	1588.19	1394.75	14.62	21.61	553.98	2388.05	9046.28	1.3	47.25	521.52	2388.05	8127.9	8.42	0.03	392	2388	100	38.94	23.46
23	-0.001	0.0003	100	518.67	642.38	1590.83	1398.81	14.62	21.61	553.49	2388.12	9043.76	1.3	47.44	522.13	2388.03	8133.88	8.39	0.03	392	2388	100	39	23.37
24	0.0023	-0.0004	100	518.67	642.77	1594.1	1399.39	14.62	21.61	554	2388.02	9054.16	1.3	47.36	522.56	2388.02	8136.61	8.39	0.03	393	2388	100	38.95	23.43
25	0	0.0002	100	518.67	642.16	1589.08	1396.07	14.62	21.61	554.11	2388.07	9047.11	1.3	47.26	522.28	2388.06	8131.15	8.43	0.03	394	2388	100	38.86	23.41
26	-0.0012	-0.0004	100	518.67	642.44	1590.47	1401.84	14.62	21.61	554.07	2388.02	9047.96	1.3	47.37	522	2388.13	8134.6	8.4	0.03	393	2388	100	38.99	23.45
27	-0.0024	0.0005	100	518.67	642.35	1582.84	1399.13	14.62	21.61	554.68	2388.12	9049.84	1.3	47.41	522.57	2388.08	8127.3	8.43	0.03	390	2388	100	39.01	23.28
28	0.0012	-0.0001	100	518.67	641.91	1584.83	1400.99	14.62	21.61	554.25	2388.05	9050.47	1.3	47.24	522.41	2388.06	8133.06	8.42	0.03	393	2388	100	38.93	23.36
29	-0.0022	0	100	518.67	642.2	1593.52	1396.08	14.62	21.61	554.37	2388.07	9045.62	1.3	47.4	522.19	2388	8137.86	8.41	0.03	390	2388	100	39.05	23.41
30	0.0014	0.0005	100	518.67	642.02	1584.18	1396.9	14.62	21.61	554.13	2388.08	9058.78	1.3	47.41	521.95	2388.06	8125.28	8.41	0.03	392	2388	100	38.94	23.34
31	0.0005	-0.0003	100	518.67	642.33	1591.38	1400.36	14.62	21.61	554.96	2388.04	9050.97	1.3	47.25	521.92	2388.07	8129.7	8.41	0.03	392	2388	100	39.02	23.5
32	-0.0042	-0.0004	100	518.67	642.71	1588.4	1402.43	14.62	21.61	554.61	2388.04	9047.13	1.3	47.28	521.91	2388.08	8134.42	8.38	0.03	392	2388	100	38.83	23.35
33	0.0015	-0.0001	100	518.67	642.54	1581.47	1400.48	14.62	21.61	554.3	2388.03	9046.46	1.3	47.47	521.7	2388.09	8126.61	8.43	0.03	392	2388	100	38.81	23.31
34	0.0003	0.0002	100	518.67	642.44	1590	1403	14.62	21.61	554.3	2388.04	9045.76	1.3	47.31	521.75	2388.01	8140.17	8.44	0.03	391	2388	100	39.13	23.42
35	-0.0004	-0.0002	100	518.67	642.54	1581.72	1405.54	14.62	21.61	554.53	2388.01	9044.56	1.3	47.25	522.32	2388.04	8132.48	8.42	0.03	392	2388	100	38.86	23.43
36	-0.0004	0	100	518.67	641.99	1579.11	1398.9	14.62	21.61	554.63	2388.07	9055.44	1.3	47.29	522.29	2388.04	8131.95	8.41	0.03	392	2388	100	38.99	23.44
37	-0.0018	-0.0002	100	518.67	641.93	1589.6	1399.5	14.62	21.61	554.7	2388.08	9055.27	1.3	47.27	522.42	2388.08	8136.61	8.42	0.03	392	2388	100	39.06	23.48
38	-0.003	-0.0001	100	518.67	642.01	1583.21	1400.69	14.62	21.61	553.97	2388.02	9051.76	1.3	47.12	522.23	2388.05	8131.81	8.41	0.03	392	2388	100	38.7	23.37
39	0	-0.0004	100	518.67	642.24	1582.08	1401.77	14.62	21.61	554.57	2388.06	9052.98	1.3	47.33	521.84	2388.01	8131.03	8.38	0.03	392	2388	100	38.84	23.38
40	0.0033	0.0004	100	518.67	642.4	1591.31	1403.21	14.62	21.61	553.72	2388.06	9048.99	1.3	47.09	522.07	2388.03	8138.08	8.44	0.03	392	2388	100	39.13	23.44

CRediT authorship contribution statement

Xingyu Li: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Vasiliy Krivtsov:** Conceptualization, Formal analysis, Writing – original draft, Project administration. **Karunesh Arora:** Conceptualization, Methodology, Software, Writing – original draft, Visualization.

Appendix A. Sample of C-MPASS data

The size of the C-MPASS data set is significant. We selected the first 40 time periods of the first turbo-fan to demonstrate the features used in training. Features used are the same for all time and turbo-fans (see Table A.4).

Appendix B. Evaluation criteria

We summarize metrics used in this study for evaluating time-to-failure prediction that accounts for censoring.

SEP Graf et al. [58] proposed SEP as the weighted geometric mean of absolute relative risks. This is the measure to evaluate the ability of the Cox model in distinguishing the reliability performance of individuals from different risk strata. For deep survival models, we leverage this idea to compare the difference in model outputs for different risk strata.

Concordance Index Concordance Index or C-index estimates the accuracy that predicted survival times of the two individuals order correctly. C-index was originally designed for time-invariant survival models. In order to generalize the C-index to the time-variant model, we follow previous studies [32,59], where the survival probabilities are compared at one individual's failure time.

$$\text{C-Index} = \frac{1}{n_p} \sum_{T_i<T_j, T_i^*<C_i^*} \mathbf{1}_{\bar{S}(T_i|\mathbf{x}_i)<\bar{S}(T_i|\mathbf{x}_j)} \tag{B.1}$$

where n_p represents the number of pairs of individuals in the dataset.

Brier Score Brier score estimates the mean squared errors of the survival probability estimate at each time period t to the binary outcomes from time-to-failure data. To balance the contribution of censored and uncensored data, the scores are rebalanced to reduce the weight of the censored data [58].

$$BS(t) = \frac{1}{N} \sum_{i=1}^{N} [\frac{\mathbf{1}_{\{T_i>t\}}(1 - \bar{S}(t|\mathbf{x}_i))^2}{\bar{S_{C*}}(t)} + \frac{\mathbf{1}_{\{T_i\le t, D_i=1\}}\bar{S}(t|\mathbf{x}_i)^2}{\bar{S_{C*}}(T_i)}] \tag{B.2}$$

References

[1] Zhao X, He S, Xie M. Utilizing experimental degradation data for warranty cost optimization under imperfect repair. Reliab Eng Syst Saf 2018;177:108–19.

[2] Alkahtani M, Choudhary A, De A, Harding JA. A decision support system based on ontology and data mining to improve design using warranty data. Comput Ind Eng 2019;128:1027–39.

[3] Li X, Epureanu BI. AI-based competition of autonomous vehicle fleets with application to fleet modularity. European J Oper Res 2020;287(3):856–74.

[4] Xu F, Yang F, Fei Z, Huang Z, Tsui K-L. Life prediction of lithium-ion batteries based on stacked denoising autoencoders. Reliab Eng Syst Saf 2021;208:107396.

[5] Kang S, Kim E, Shim J, Cho S, Chang W, Kim J. Mining the relationship between production and customer service data for failure analysis of industrial products. Comput Ind Eng 2017;106:137–46.

[6] Zou Z, Chen J, Pang X. Task space-based dynamic trajectory planning for digging process of a hydraulic excavator with the integration of soil–bucket interaction. Proc Inst Mech Eng K: J Multi-Body Dyn 2019;233(3):598–616.

[7] Oh Y, Bai DS. Field data analyses with additional after-warranty failure data. Reliab Eng Syst Saf 2001;72(1):1–8.

[8] Li X, Nassehi A, Epureanu BI. Degradation-aware decision making in reconfigurable manufacturing systems. CIRP Annal 2019;68(1):431–4.

[9] Krivtsov VV. Field data analysis & statistical warranty forecasting. IEEE Catalog No CFP11RAM-CDR 2011.

[10] Modarres M, Kaminskiy MP, Krivtsov V. Reliability engineering and risk analysis: A practical Guide. CRC Press; 2016.

[11] Lawless JF, Crowder M, Lee K-A. Analysis of reliability and warranty claims in products with age and usage scales. Technometrics 2009;51(1):14–24.

[12] Huang Y-S, Gau W-Y, Ho J-W. Cost analysis of two-dimensional warranty for products with periodic preventive maintenance. Reliab Eng Syst Saf 2015;134:51–8.

[13] Krivtsov V, Frankstein M. Nonparametric estimation of marginal failure distributions from dually censored automotive data. In: Annual symposium reliability and maintainability, 2004-RAMS. IEEE; 2004, p. 86–9.

[14] Karim M, Suzuki K. Analysis of warranty data with covariates. Proc Inst Mech Eng O: J Risk Reliab 2007;221(4):249–55.

[15] Attardi L, Guida M, Pulcini G. A mixed-Weibull regression model for the analysis of automotive warranty data. Reliab Eng Syst Saf 2005;87(2):265–73.

[16] Cox DR. The analysis of multivariate binary data. Appl Stat 1972;113–20.

[17] Krivtsov VV, Tananko DE, Davis TP. Regression approach to tire reliability analysis. Reliab Eng Syst Saf 2002;78(3):267–73.

[18] Vinta S. Analysis of data to predict warranty cost for various regions. In: 2009 annual reliability and maintainability symposium. IEEE; 2009, p. 78–82.

[19] Kvamme H, Borgan Ø, Scheel I. Time-to-event prediction with neural networks and Cox regression. 2019, ArXiv preprint arXiv:1907.00825.

[20] Nagpal C, Li XR, Dubrawski A. Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. IEEE J Biomed Health Inf 2021.

[21] Katzman JL, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y. Deep survival: A deep cox proportional hazards network. Stat 2016;1050(2).

[22] Meeker WQ, Hong Y. Reliability meets big data: opportunities and challenges. Qual Eng 2014;26(1):102–16.

[23] Zhou Y, Bridgelall R. Review of usage of real-world connected vehicle data. Transp Res Rec 2020;2674(10):939–50.

[24] Zhang Y, Gantt GW, Rychlinski MJ, Edwards RM, Correia JJ, Wolf CE. Connected vehicle diagnostics and prognostics, concept, and initial practice. IEEE Trans Reliab 2009;58(2):286–94.

[25] Ren L, Liu Y, Wang X, Lü J, Deen MJ. Cloud-edge based lightweight temporal convolutional networks for remaining useful life prediction in iiot. IEEE Internet Things J 2020.

[26] Tao F, Qi Q, Liu A, Kusiak A. Data-driven smart manufacturing. J Manuf Syst 2018;48:157–69.

[27] Yang D, He Z, He S. Warranty claims forecasting based on a general imperfect repair model considering usage rate. Reliab Eng Syst Saf 2016;145:147–54.

[28] Irshad UB, Rafique S, Town G. Reliability assessment considering intermittent usage of electric vehicles in parking lots. In: 2020 IEEE 29th international symposium on industrial electronics. IEEE; 2020, p. 965–70.

[29] Li X, Zhang W, Ding Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Reliab Eng Syst Saf 2019;182:208–18.

[30] Cinar YG, Mirisaee H, Goswami P, Gaussier E, Aït-Bachir A, Strijov V. Position-based content attention for time series forecasting with sequence-to-sequence rnns. In: International conference on neural information processing. Springer; 2017, p. 533–44.

[31] Tian Q, Liu J, Wang D, Tang A. Time series prediction with interpretable data reconstruction. In: Proceedings of the 28th ACM international conference on information and knowledge management; 2019. p. 2133–6.

[32] Kvamme H, Borgan Ø. Continuous and discrete-time survival prediction with neural networks. 2019, ArXiv preprint arXiv:1910.06724.

[33] Faraggi D, Simon R. A neural network model for survival data. Stat Med 1995;14(1):73–82.

[34] Liestbl K, Andersen PK, Andersen U. Survival analysis and neural nets. Stat Med 1994;13(12):1189–200.

[35] Xiang A, Lapuerta P, Ryutov A, Buckley J, Azen S. Comparison of the performance of neural network methods and cox regression for censored survival data. Comput Statist Data Anal 2000;34(2):243–57.

[36] Mariani L, Coradini D, Biganzoli E, Boracchi P, Marubini E, Pilotti S, Salvadori B, Silvestrini R, Veronesi U, Zucali R, et al. Prognostic factors for metachronous contralateral breast cancer: a comparison of the linear cox regression model and its artificial neural network extension. Breast Cancer Res Treat 1997;44(2):167–78.

[37] Franco L, Jerez JM, Alba E, et al. Artificial neural networks and prognosis in medicine. Survival analysis in breast cancer patients. In: ESANN; 2005. p. 91–102.

[38] Ching T, Zhu X, Garmire LX. Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. PLoS Comput Biol 2018;14(4):e1006076.

[39] Bellera CA, MacGrogan G, Debled M, de Lara CT, Brouste V, Mathoulin-Pélissier S. Variables with time-varying effects and the Cox model: some statistical concepts illustrated with a prognostic factor study in breast cancer. BMC Med Res Methodol 2010;10(1):1–12.

[40] Tian L, Zucker D, Wei L. On the Cox model with time-varying regression coefficients. J Amer Statist Assoc 2005;100(469):172–83.

[41] Brown SF, Branford AJ, Moran W. On the use of artificial neural networks for the analysis of survival data. IEEE Trans Neural Netw 1997;8(5):1071–7.

[42] Biganzoli E, Boracchi P, Mariani L, Marubini E. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med 1998;17(10):1169–86.

[43] Gensheimer MF, Narasimhan B. A scalable discrete-time survival model for neural networks. PeerJ 2019;7:e6257.

[44] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 2018;172:1–11.

[45] Zhang Y, Lobo-Mueller EM, Karanicolas P, Gallinger S, Haider MA, Khalvati F. Cnn-based survival model for pancreatic ductal adenocarcinoma in medical imaging. BMC Med Imag 2020;20(1):1–8.

[46] Lee C, Zame W, Yoon J, van der Schaar M. Deephit: A deep learning approach to survival analysis with competing risks. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32(1); 2018.

[47] Yu C-N, Greiner R, Lin H-C, Baracos V. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. Adv Neural Inf Process Syst 2011;24:1845–53.

[48] Giunchiglia E, Nemchenko A, van der Schaar M. Rnn-SURV: A deep recurrent model for survival analysis. In: International Conference on Artificial Neural Networks. Springer; 2018, p. 23–32.

[49] Lombardo E, Kurz C, Marschner S, Avanzo M, Gagliardi V, Fanetti G, Franchin G, Stancanello J, Corradini S, Niyazi M, et al. Distant metastasis time to event analysis with CNNs in independent head and neck cancer cohorts. Sci Rep 2021;11(1):1–12.

[50] Qin Y, Song D, Chen H, Cheng W, Jiang G, Cottrell G. A dual-stage attention-based recurrent neural network for time series prediction. 2017, ArXiv preprint arXiv:1704.02971.

[51] Benidis K, Rangapuram SS, Flunkert V, Wang B, Maddix D, Turkmen C, Gasthaus J, Bohlke-Schneider M, Salinas D, Stella L, et al. Neural forecasting: Introduction and literature overview. 2020, ArXiv preprint arXiv:2004.10240.

[52] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014, ArXiv preprint arXiv:1406.1078.

[53] Kalchbrenner N, Blunsom P. Recurrent continuous translation models. In: Proceedings of the 2013 conference on empirical methods in natural language processing; 2013. p. 1700–9.

[54] Wang J, Zhang C. Software reliability prediction using a deep learning model based on the RNN encoder–decoder. Reliab Eng Syst Saf 2018;170:73–82.

[55] Cho K. Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images. In: International conference on machine learning. PMLR; 2013, p. 432–40.

[56] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. 2014, ArXiv preprint arXiv:1409.3215.

[57] Ellefsen AL, Bjørlykhaug E, Æsøy V, Ushakov S, Zhang H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab Eng Syst Saf 2019;183:240–51.

[58] Graf E, Schmoor C, Sauerbrei W, Schumacher M. Assessment and comparison of prognostic classification schemes for survival data. Stat Med 1999;18(17–18):2529–45.

[59] Antolini L, Boracchi P, Biganzoli E. A time-dependent discrimination index for survival data. Stat Med 2005;24(24):3927–44.