Received: 16 June 2023 | Revised: 14 March 2024 | Accepted: 7 June 2024

DOI: 10.1002/sim.10152

RESEARCH ARTICLE

Statistics in Medicine WILEY

deepAFT: A nonlinear accelerated failure time model with artificial neural network

Patrick A. Norman[1] | **Wanlu Li**[2] | **Wenyu Jiang**[2] | **Bingshu E. Chen**[3]

[1]Kingston General Health Research Institute, Queen's University, Kingston, Ontario, Canada

[2]Department of Mathematics and Statistics, Queen's University, Kingston, Ontario, Canada

[3]Department of Public Health Sciences and Canadian Cancer Trials Group, Queen's University, Kingston, Ontario, Canada

Correspondence
Bingshu E. Chen, Department of Public Health Sciences and Canadian Cancer Trials Group, Queen's University, Kingston, Ontario, Canada.
Email: bechen@ctg.queensu.ca

Funding information
Natural Sciences and Engineering Research Council of Canada (NSERC)

The Cox regression model or accelerated failure time regression models are often used for describing the relationship between survival outcomes and potential explanatory variables. These models assume the studied covariates are connected to the survival time or its distribution or their transformations through a function of a linear regression form. In this article, we propose nonparametric, nonlinear algorithms (deepAFT methods) based on deep artificial neural networks to model survival outcome data in the broad distribution family of accelerated failure time models. The proposed methods predict survival outcomes directly and tackle the problem of censoring via an imputation algorithm as well as re-weighting and transformation techniques based on the inverse probabilities of censoring. Through extensive simulation studies, we confirm that the proposed deepAFT methods achieve accurate predictions. They outperform the existing regression models in prediction accuracy, while being flexible and robust in modeling covariate effects of various nonlinear forms. Their prediction performance is comparable to other established deep learning methods such as deepSurv and random survival forest methods. Even though the direct output is the expected survival time, the proposed AFT methods also provide predictions for distributional functions such as the cumulative hazard and survival functions without additional learning efforts. For situations where the popular Cox regression model may not be appropriate, the deepAFT methods provide useful and effective alternatives, as shown in simulations, and demonstrated in applications to a lymphoma clinical trial study.

KEYWORDS
accelerated failure time, clinical trials, deep neural network, nonlinear model, survival analysis

1 | INTRODUCTION

Recent developments in machine learning techniques, especially the rapid advancement in the field of deep learning, provide numerous potential research topics at the intersection of statistics and computer sciences.[1] Neural networks provide fast algorithms for tackling the challenge of nonlinear features and making effective predictions.[2,3] The artificial neural networks (ANN) methods typically deal with continuous or categorical outcomes that are completely

observed. Survival outcome data that are common in medical and health studies must be handled differently because of censoring.

Deep learning[4] methods are becoming popular and widely used in health and medical research for survival outcome data. Faraggi and Simon developed a single hidden layer ANN for the Cox proportional hazards (PH) model for predicting survival distribution.[5] Many ANN approaches including deep neural networks (DNN) have been developed for survival data. Some are also based on PH assumptions and Cox-type models,[6,7] some do not make model assumptions,[8-11] and some rely on specific distributional assumptions.[12] These methods often focus directly on predicting hazards, survival probabilities, or other distributional functions of survival time. Additional efforts or a round of deep learning are typically needed for predicting survival time.[11,13] For example, in the proposed method of Baek et al,[13] survival times are predicted through distribution function networks, utilizing the hazard function predicted in the first round of deep learning by deepSurv of Katzman et al.[7]

New techniques are developed in deep learning methods for survival data and are illustrated and compared in various contexts of medical applications such as cancer studies and genomics.[14-20] Chong et al established novel methods and theories combining DNN and clear statistical inference based on partial linear modeling in survival analysis; they also offered an extensive literature review of theoretical development, methodology and application in the related fields.[21] An up-to-date screening and literature review was conducted in Wiegrebe et al, which included 61 deep learning methods for survival data.[22] They categorized the reviewed methods based on a number of different attributes, for example, (1) various types of survival outcomes such as right-censored, interval-censored, and recurrent event data, (2) various bases of estimation models utilized in ANN methods, such as Cox-type, parametric estimation, and discrete time estimation. Comprehensive reviews and comparisons of machine learning methodologies for survival data can also be found in recent publications,[23-27] covering boosting, tree-based methods, DNN, etc., and/or machine learning methods for feature selections and high dimensional data. Even though an earlier study of Sargent showed that a simple single-layer ANN like that of Faraggi and Simon[5] may not outperform a standard Cox regression model,[28] these aforementioned more recent reviews and numerical evaluations have confirmed that DNN methods (multiple-layered ANN) such as the deepSurv have superior prediction performance for survival data.[7]

The classical methods for survival data often rely on the Cox regression model, which is originally based on the PH model assumption.[29,30] The Cox model is semi-parametric for modeling hazard function while assuming linear covariate effects on the log relative risk. The PH assumption, however, may not be satisfied in some situations. In this article, we will consider an alternative model for survival outcome data, the accelerated failure time (AFT) model. It models the survival time (or, more often, log survival time) directly, rather than modeling the hazard function. This allows for a straightforward interpretation of the covariate effects. Parametric methods for AFT regression models can be restrictive for making specific model assumptions for the error distributions.[31] Buckley and James proposed a semi-parametric linear regression method with iterative imputations for the censored survival time.[32,33] Lin proposed a semi-parametric linear regression model for medical cost data utilizing the inverse probability of censoring weights (IPCW).[34] The idea of IPCW dates back to Robins and Rotnitzky and some useful discussions and modifications can be found in Fan and Gijbels and Robins et al for recovering the loss of information due to censoring.[35-37] Rank-based estimation provides another type of semi-parametric method for AFT regression.[38]

Steingrimsson and Morrison proposed deep learning methods for survival data by modifying the loss functions of full data DNN for censoring, through doubly robust adjustments.[11,39] Their methods are nonparametric with no model assumptions for survival time,[11] and their proposed constructions of unbiased outcomes and loss functions are robust and generally applicable to data with censoring. However, the method or loss function used is dependent on the output, which could be either the survival probability at a given time point or the restricted mean survival time; predictions for both output types would require two separate implementations of the DNN.[11]

In this article, we propose DNN methods for the AFT model (deepAFT). Like Buckley and James[32] and Lin,[34] we will only assume an AFT model form without specifying the error distribution. We develop a multiple-hidden-layer feed-forward ANN for modeling the nonlinear effects of the input variables on survival outcomes, and propose an imputation algorithm and two techniques based on IPCW for handling the challenge of censoring in survival data.[32,34,36] The proposed methods can predict different types of output of practical interests, such as the survival time, restricted mean survival time, or survival probability function over the entire range of time. Once a proposed DNN method for the AFT model is implemented with the mean survival time and baseline survival function outputs, these other output types or measures are obtainable from direct algebraic calculations. The proposed methods are evaluated in simulations and demonstrated through application to a lymphoma clinical trial study. They provide more accurate predictions than the

existing regression methods in the survival analysis literature such as the linear Cox regression and parametric AFT regression models. The advantages of the deepAFT methods are eminent compared to the popular regression models when the PH model assumption is not satisfied and when flexible modeling is crucial to capture the complicated nonlinear covariate impacts on survival outcomes. The proposed deepAFT methods turn out to have similar prediction performance as their counterparts, deepSurv and random survival forest.[7,14] These studies confirm that the proposed methods are useful and effective as DNN methods. Their main strengths include offering new perspectives and interpretations based on the AFT model structure, and easily producing predictions for survival time and other related outcome measures and distributional functions, all from one DNN implementation.

2 | DEEP LEARNING METHODS FOR THE AFT MODEL

Let $\tilde{T}$ be a random variable describing the survival time of an individual in a study. It is usually the time to the occurrence of a specific event, such as death from a disease under study or cancer recurrence after remission. $\tilde{T}$ may not be completely observed due to right-censoring. Let C be a random variable describing the (potential) censoring time of the individual. We denote the observed survival time by $T = \min(\tilde{T}, C)$, and introduce a censoring indicator $\Delta = I(\tilde{T} \leq C)$. A p-dimensional covariate vector $\boldsymbol{z}$ is observed for the individual. For individuals $i = 1, 2, \cdots, n$, we assume that $\tilde{T}_i$ and C_i are independent given the covariate $\boldsymbol{z}_i$. Denote the observed data for individual i by $(t_i, \delta_i, \boldsymbol{z}_i)$.

We are interested in studying the relationship between $\tilde{T}$ and $\boldsymbol{z}$. Instead of the Cox regression model, we consider the accelerated failure time model and assume that the log-failure time $\tilde{Y} = \log(\tilde{T})$ takes the form

$$\tilde{Y} = \mu(\boldsymbol{z}) + e, \tag{1}$$

where $\mu(\boldsymbol{z})$ is a function of the covariate vector $\boldsymbol{z}$ and e is an error term from a distribution with a constant mean and a common variance. We propose deep learning methods for estimating $\mu(\boldsymbol{z})$ in (1) based on ANN. The methods aim to estimate the log survival time, which provides an estimation or prediction of the survival time upon a simple transformation.

A parametric method in survival analysis typically assumes that $e = \sigma\epsilon$ in model (1), where ϵ is a standard random variable of a family of location-scale distributions, while referring to μ as the location parameter and σ as the scale parameter. A common choice is to assume $\mu(\boldsymbol{z}) = \beta^T z$ for a vector of regression coefficients β, which leads to the parametric AFT regression model in the literature, usually analyzed by the method of maximum likelihood estimation.

2.1 | ANN for AFT model for data without censoring

Unlike the AFT regression models, the deep learning methods we propose do not assume a specific form for the location function $\mu(\boldsymbol{z})$ and do not assume a specific distribution for the error term in (1). We do need to assume that the e_i terms are independent and identically distributed for individuals $i = 1, \ldots, n$. If there is no censoring, we can run an ANN directly with the covariate vector $\boldsymbol{z}$ as the inputs and the completely observed log survival time y as the response, and obtain the outputs, that is, one single prediction of log survival time for each individual. For data without censoring, we recommend an ANN with m hidden layers, using a rectifier linear unit (ReLU) activation function, $\phi(x) = \max(0, x)$,[40] and the mean squared error (MSE) loss function $\frac{1}{n}\sum_{i=1}^{n}\{y_i - \mu(\boldsymbol{z}_i)\}^2$. With $m = 2$, for example, the feed-forward output of the network can be expressed as

$$\mu(\boldsymbol{z}) = b^{(3)} + w^{(3)}\phi(b^{(2)} + w^{(2)}\phi(b^{(1)} + w^{(1)}\boldsymbol{z})),$$

where $w^{(1)}, w^{(2)}, w^{(3)}$ are the weight matrices (slope parameters) of $p_1 \times p$, $p_2 \times p_1$, $1 \times p_2$ dimensions, and $b^{(1)}, b^{(2)}, b^{(3)}$ are the bias term vectors (intercept parameters) of p_1, p_2 and 1 dimensions, for the first and second inner layers with p_1 and p_2 nodes respectively. With the MSE loss function, the ANN easily obtains the gradients of the output with respect to the weight parameters, and completes in turn the back-propagation.

Other choices of activation functions such as the sigmoid function and hyperbolic tangent function, and loss functions such as the cross-entropy function and Kullback–Leibler divergence loss function, are possible and useful in building the ANN method in various application scenarios.[1] The main advantage of the recommended ANN with ReLU activation and

MSE loss is that it is very efficient in computation, involving only basic mathematical operations and simple derivatives. The method is suitable for training a DNN on a large and complex dataset.

2.2 | Deep learning methods for data with censoring

To handle typical survival data subject to right censoring, we develop three methods built upon the above ANN method. The first way to incorporate censoring is by imputation conditioning on the covariates, similar to the idea introduced by Buckley and James, based on the AFT model assumption.[32] Without loss of generality, assume the data arise from the model (1) with $\mu(\boldsymbol{z}) = 0$. If the observed t is a censoring time ($\delta = 0$), we impute the failure time by $t^* = E(\tilde{T}|\tilde{T} > t)$. With $\mu(\boldsymbol{z}) = 0$, the failure time random variables for individuals $i = 1, \ldots, n$ are independent, identically distributed (i.i.d), we calculate the imputed data by

$$t_i^* = \begin{cases} \sum_{j:t_j>t_i} \frac{t_j d\hat{S}(t_j)}{\hat{S}(t_i)} & \text{for } \delta_i = 0, \\ t_i & \text{for } \delta_i = 1, \end{cases} \tag{2}$$

where $\hat{S}(t)$ is the Kaplan–Meier estimate of the survival function, and $d\hat{S}(t_j) = \hat{S}(t_j-) - \hat{S}(t_j)$ is the change in the survival function at time t.[41] For data with $\mu(\boldsymbol{z}) \neq 0$, we simply apply this imputation method after removing the current estimates $\hat{\mu}(\boldsymbol{z})$ of the location parameter from the corresponding responses, according to the form of model (1).

We propose the first deep learning method (deepAFT) for model (1) as an iterative method.

Initial step

Outer loop: Apply the imputation (2) to the original data (t_i, δ_i), then define $y_i^* = \log(t_i^*)$, $i = 1, \ldots, n$. This means we initially take $\mu(\boldsymbol{z})$ to be 0.

Inner loop: Train the ANN based on the imputed data $(y_i^*, \boldsymbol{z}_i)$, producing an updated estimate of the location parameter, $\hat{\mu}^{(1)}(\boldsymbol{z})$. According to model (1), define $v_i^{(1)} = t_i^* / \exp\{\hat{\mu}^{(1)}(\boldsymbol{z}_i)\}$ for all individuals under study.

Step k

Outer loop: Impute for $(v_i^{(k)}, \delta_i)$, $i = 1, \ldots, n$, with the same idea as (2),

$$v_i^{*(k)} = \begin{cases} \sum_{j:v_j^{(k)}>v_i^{(k)}} \frac{v_j^{(k)} d\hat{S}(v_j^{(k)})}{\hat{S}(v_i^{(k)})} & \text{for } \delta_i = 0, \\ v_i^{(k)} & \text{for } \delta_i = 1. \end{cases} \tag{3}$$

Update the responses by $y_i^{*(k)} = \hat{\mu}^{(k)}(\boldsymbol{z}_i) + \log v_i^{*(k)}$, that is, update the imputed failure time by $t_i^{*(k)} = v_i^{*(k)} \exp\{\hat{\mu}^{(k)}(\boldsymbol{z}_i)\}$.

Inner loop: Train the ANN based on the imputed data $(y_i^{*(k)}, \boldsymbol{z}_i)$ using the regular mean squared error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left\{y_i^{*(k)} - \mu(\boldsymbol{z}_i)\right\}^2,$$

as the loss function, and obtain an updated estimate $\hat{\mu}^{(k+1)}(\boldsymbol{z})$. Define $v_i^{(k+1)} = t_i^{*(k)} / \exp\{\hat{\mu}^{(k+1)}(\boldsymbol{z}_i)\}$ for all individuals under study.

Stopping criteria

Step k with an outer loop of imputation and an inner loop of training the ANN will repeat for $k = 1, 2, \ldots$, until either $|\hat{\mu}^{(k)}(\boldsymbol{z}) - \hat{\mu}^{(k-1)}(\boldsymbol{z})| \leq d$, or until M iterations have been completed. The threshold value d and maximum iteration number M are both prespecified.

The stochastic gradient descent method is used to train the ANN model.[42] The optimal hyper parameters such as the learning rate, the coefficient for the momentum gradient descent and the L_2 regularization parameters are determined using a random search method proposed by Bergstra and Bengio.[43] If the underlying distribution of the data is an AFT

model, the estimates $\hat{\mu}^{(k)}(\boldsymbol{z})$ should eventually converge to the true $\mu(\boldsymbol{z})$. In this way, the deepAFT method predicts the survival time despite censoring in the observed data.

Another idea to incorporate the right censoring in training ANN for survival data is to adjust the loss function by the inverse probability of censoring weights. The IPCW technique is widely used in developing a consistent estimation method or defining a cost function, for information recovery in various types of censoring or missing data situations.[34,35,37,44] For the observed data $(t_i, \delta_i, \boldsymbol{z}_i)$ and $y_i = \log\ t_i$, we define the loss function as the adjusted mean squared error

$$\text{AMSE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\delta_i}{\hat{G}(t_i)}\{y_i - \mu(\boldsymbol{z}_i)\}^2, \tag{4}$$

where $\hat{G}(t_i)$ is the Kaplan–Meier estimate for the survival function of the censoring time C. The estimate $\hat{G}(t_i)$ is obtained for the data $(t_i, 1-\delta_i)$, describing the censoring time distribution. An individual that is censored at t_i (with $\delta_i = 0$) does not have a completely observed response, and does not contribute to the loss function. For an individual that has failed at t_i, its true censoring time C is greater than t_i; it has $\delta_i = 1$ and $G(t_i) = P(C > t_i)$ is a suitable weight in the loss function to compensate for censoring. The AMSE reduces to the regular MSE for data with no censoring.

We now propose the second deep learning method for model (1) and name it the deepAFT-IPCW method, which is to train an ANN as described in Section 2.1 for the observed data $y_i = (\log t_i, \delta_i, \boldsymbol{z}_i)$, while taking the AMSE in (4) as the loss function. This method trains an ANN only once for the observed data, no iterations are required.

When the proportion of censoring is large, the IPCW method may cause large variations when predicting the survival time. Fan and Gijbels suggest using the following unbiased transformation of the censored survival time

$$\check{T}_i = \delta_i\phi_1(T_i) + (1-\delta_i)\phi_2(T_i) = \delta_i\{\phi_1(T_i) - \phi_2(T_i)\} + \phi_2(T_i),$$

where

$$\begin{cases} \phi_1(t) &= (1+\alpha)\int_0^t \hat{G}(s)^{-1}ds - \alpha t\hat{G}(t)^{-1} \\ \phi_2(t) &= (1+\alpha)\int_0^t \hat{G}(s)^{-1}ds \end{cases},$$

with a tuning parameter α.[36] This transformation ensures that $E(\check{T}|\boldsymbol{Z} = \boldsymbol{z}) = E(\tilde{T}|\boldsymbol{Z} = \boldsymbol{z})$. When $\alpha = -1$, this reduces to the traditional IPCW method. When $\alpha = 0$, this is the Leurgans' unbiased synthetic data $\check{T}_i = \int_0^{T_i}\hat{G}^{-1}(s)ds$, which has the potential to reduce the variance of the predicted survival time.[45] Following the suggestion of Fan and Gijbels, we choose the tuning parameter in the above transformation as

$$\hat{\alpha} = \min_{i,\delta_i=1}\frac{\int_0^{t_i}\hat{G}(s)^{-1}ds - t_i}{t_i\hat{G}(t_i)^{-1} - \int_0^{t_i}\hat{G}(s)^{-1}ds}.$$

We then propose the third method, the deepAFT-Transform method for model (1), which is to train the ANN model as in Section 2.1 using regular mean squared error (MSE) as the loss function, for the transformed data $(\log(\check{T}_i), \boldsymbol{z}_i)$ with $\check{T}_i$ defined above, $i = 1, \ldots, n$.

The proposed deepAFT methods only assume that log survival time has a general AFT model structure (1). They do not assum specific error distributions, not rely on parametric estimation, and hence widely applicable.

2.3 | Further estimations for the AFT model

After fitting model (1) by one of the proposed deep learning methods, we can easily make further personalized estimations and obtain prognostic measures such as the survival probability at a landmark time point, restricted mean survival time, median survival time, and predictive responses to treatments in clinical trials. These have great implications in the area of modern personalized medicine. It is worth noting that all these measures are directly obtainable from simple algebraic calculations upon the completion of a deepAFT method.

Let $T_0 = \tilde{T}_i\exp(-\mu_i)$ be the survival time of subjects with covariate values $\boldsymbol{z} = \boldsymbol{0}$, and $S_{T_0}(t)$ be the corresponding baseline survival function. We can estimate $S_{T_0}(t)$ by applying the Kaplan–Meier method to the scaled survival data

$(t_i \exp(-\hat{\mu}_i), \delta_i)$, where $\hat{\mu}_i = \hat{\mu}(\boldsymbol{z}_i)$ is the predicted score obtained from the deep learning method for subject i. By properties of the AFT model, the predicted survival function for subject i is then

$$\hat{S}(t|\boldsymbol{z}_i) = \hat{S}_{T_0}(te^{-\hat{\mu}_i}). \tag{5}$$

It allows us to predict the survival probability at a given time, say 1 or 5 years.

The restricted mean survival time $E\{\min(\tilde{T}, \tau)\}$ up to a prespecified time τ can then be estimated by

$$\hat{E}\{\min(\tilde{T}, \tau)|\boldsymbol{z}_i\} = \int_0^\tau \hat{S}(t|\boldsymbol{z}_i)dt,$$

once the survival function is estimated from (5).

Suppose m_0 is the median survival time of T_0. By model (1), the median of subject i can be estimated by $\hat{m}_i = m_0 \exp(\hat{\mu}_i)$. A predicted score $\hat{\mu}_i > 0$ implies that subject i will have a longer median survival than those with baseline covariates $\boldsymbol{z} = \boldsymbol{0}$.

In clinical trials, it is important to predict the survival benefit of a certain treatment while taking into account complex clinical, biochemical and genetic factors of individual patients. Describe the treatment assignment by the first covariate z_1, such that $z_1 = 1$ (or $z_1 = 0$) for those assigned the new (or standard) treatment. We can easily obtain the predicted scores $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(0)}$ for the two different treatment groups of patients with covariate vectors $\boldsymbol{z} = (1, \boldsymbol{x})$ and $\boldsymbol{z} = (0, \boldsymbol{x})$ respectively, where $\boldsymbol{x}$ contains all other covariates under consideration. If $\hat{\mu}^{(1)} > \hat{\mu}^{(0)}$, the patients receiving the new treatment will have $\{\exp(\hat{\mu}^{(1)} - \hat{\mu}^{(0)}) - 1\} \times 100\%$ longer expected survival time than if they had the standard treatment. We can also provide personalized Kaplan–Meier survival curves based on (5), assuming the patient received either the new treatment or the standard treatment, accounting for all other relevant covariate information of this particular patient.

3 | SIMULATION STUDIES

We examine the performance of the proposed methods in three simulation scenarios. In the first scenario, we generate time to event data from the AFT model (1) with location parameter

$$\mu(\boldsymbol{z}) = 1.2 + 1.5z_1 + 0.8\sin(3\pi z_2) + 1.7z_3^2 - 1.2z_2z_4 + 0.2|z_2|^{z_4} +$$

$$1.2I\{(4z_3^2 + (z_4 - 1)^2) > 0.5\}, \tag{6}$$

and a specified value for the scale parameter σ, where $I\{\}$ is an indicator function. The covariates are randomly generated, with z_1 from Bernoulli with $P(Z_1 = 1) = 0.5$, z_2 from uniform distribution between $(-1, 1)$, z_3 from normal distribution with mean 0 and standard deviation 0.5, and z_4 uniform distribution between $(0, 2)$. The error in (1) is generated from normal distribution.

As the second simulation scenario, to investigate the robustness of the proposed method, we generate data from the proportional hazards (PH) model with hazard function

$$\lambda(t, \boldsymbol{z}) = \lambda_0(t)\exp\{\eta(\boldsymbol{z})\}, \tag{7}$$

where the baseline hazard function $\lambda_0(t) = \exp(-5 + 2.5\cos(\pi t) + 0.5t^{0.3})$, and $\eta(\boldsymbol{z}) = \mu(\boldsymbol{z}) - 1.2$, with $\mu(\boldsymbol{z})$ defined in (6).

In the first and second simulation scenarios, in each simulation setting, the censoring time follows an exponential distribution, whose rate parameter is selected to ensure that the censoring rate is around 30% among all subjects. For each simulation setting considered, we generate $n = 1000$ subjects in each simulation run. To evaluate the prediction performance of different methods, 75% of the n subjects generated in each simulation run are used to build the prediction models, and the remaining 25% of the generated subjects are reserved as test data.

In the third simulation scenario, we further simulate sparse data from the exact same simulation model initially considered by Katzman et al, in section 4.2.2.[7] The datasets are generated from the following simulation model specified by

the hazard function

$$\lambda(t|z_1, \ldots, z_{10}) = \log(5)\exp\{-0.5(z_1^2 + z_2^2)\}, \tag{8}$$

where $z_1, \ldots, z_{10}$ follow an uniform distribution within the interval $[-1.0, 1.0]$. We also follow the specified censoring mechanism of Katzman et al, which is to keep a fixed censoring rate of 10%, by setting a fixed value C_0 and censoring all subjects with survival time $> C_0$. In each round of simulation, a total sample of $n = 5000$ subjects are generated, among which 4000 are taken as training data and the remaining 1000 subjects as testing data. All simulation setups in this scenario follow those in Katzman et al, to ensure fair comparisons between the proposed deepAFT methods and their deepSurv method.[7] Each setting in each of the three simulation scenarios is repeated with $R = 500$ simulation runs.

For comparisons to the three proposed deepAFT methods, we investigate prediction methods based on a log normal parametric regression model, a semi-parametric model, that is, the Cox regression model, the deepSurv method with the underlying PH model assumption,[7] and the nonlinear random survival forest (RSF) model with no distributional assumptions.[14] The regular regression models are developed with a linear predictor $\boldsymbol{\beta}^T\boldsymbol{z}$, with all covariates in the simulated dataset.

For fair comparisons, we use the same ANN structure for both the proposed deepAFT methods and deepSurv methods. For the simulation in Table 1, all the ANN methods consist of an input layer, three hidden layers and an output layer. The number of neurons in the hidden layers are 7, 11, and 7, respectively. For the activation functions, we use ReLU for the first, second and third layers and an identity function for the output layer, respectively. We develop an R function *hyperTuning()* specifically to determine the optimal hyper parameters such as the learning rate, the coefficient for the momentum gradient descent and the L_2 regularization parameters that maximize the cross-validation C-index.

Methods are evaluated based on the measures of concordance index (C-index)[46] and mean squared error. The C-index is the probability $P(\hat{\mu}_q > \hat{\mu}_r|\tilde{T}_q > \tilde{T}_r)$ for random individuals q and r, where in the context of our study $\hat{\mu}_q$ and $\hat{\mu}_r$ are the respective predicted (mean) survival times. The empirical C-index for a prediction model can be evaluated on a test set as the number of concordant pairs divided by the total possible number of comparable pairs of observations. It is the most commonly used discriminatory index in survival analysis. A C-index of exactly 1 indicates that the model's predictions are perfectly ordered, 0.5 is the average output of a totally random model, and a C-index of 0.6 to 0.7 is typical of a predictive model. The C-index is popular because it is important for evaluating the prediction models in terms of their discriminatory performance. It is also popular because of the difficulty to obtain the mean squared error of a Cox model, which does not directly predict (mean) survival times. Likewise, the deepSurv (which is also Cox-based) and the random forest methods predict the hazard function and the cumulative hazard function respectively, rather than predicting the (mean) survival time directly; the evaluations based on mean squared errors are then not straightforward without substantial efforts in further learning or survival time generations/predictions. When finding the C-index for a Cox-based model and a random forest method, we simply determine the order of $\hat{\mu}_q$ and $\hat{\mu}_r$ by the predicted survival probabilities or risk scores of individuals q and r. For each prediction method considered (except the Cox model, deepSurv and random forest methods), we also evaluate the performance based on the mean squared error when the data are generated from the first simulation scenario, the AFT model (1) and (6). More specifically in this scenario, in a simulation replication, for each prediction model built on a training set, we calculate the mean squared error on the corresponding test set with adjustment for censoring, that is, the AMSE as in (4).

Table 1 reports the C-indices for comparing the three proposed deepAFT methods, the deepSurv, and random forest methods (RSF), the Cox regression model (Cox PH), and the log normal AFT regression model (Log Normal). It reports the average of the empirical C-indices, and the corresponding Monte-Carlo standard deviation (sd) and 2.5 and 97.5 percentiles of the C-indices across the simulation replications. For all the simulation scenarios considered, the proposed deepAFT methods and the other two deep learning methods (deepSurv and RSF methods) achieve substantially higher C-indices than the regular regression models (Cox PH and Log Normal).

For the simulation scenarios with data from the AFT model (1) and PH model (7), the RSF method has the highest numerical C-index value, while the first deepAFT achieves the second highest, the deepSurv method the third highest C-index values; these achieved C-indices are fairly close in values and their corresponding 2.5 percentile to 97.5 percentile intervals overlap substantially for these deep learning methods. For the simulation scenario of the sparse data model considered by Katzman et al, the censoring time is fixed at C_0, which is a degenerate probability distribution; the deepAFT-IPCW and deepAFT-transform methods are then not applicable. For this simulation scenario, both the Cox PH and Log Normal AFT regression models provide little prediction values with the C-indices around 0.50; the deepAFT,

TABLE 1 Performance of Cox PH regression model, Log Normal AFT regression model, deepAFT, deepSurv and random survival forest (RSF) methods based on empirical C-index, averaged across $R = 500$ replications, for data simulated from AFT model (1), PH model (7), and sparse data from PH model (8).

Method	C-index	sd	(2.5, 97.5) percentiles
AFT model (1) with $\mu(\boldsymbol{Z})$ given in (6) and $\sigma = 0.5$			
Cox PH	0.7643	0.0172	(0.730, 0.798)
Log Normal	0.7645	0.0172	(0.731, 0.797)
deepAFT	0.8320	0.0174	(0.800, 0.864)
deepAFT-IPCW	0.8061	0.0176	(0.771, 0.836)
deepAFT-Tranform	0.8139	0.0148	(0.784, 0.840)
deepSurv	0.8085	0.0156	(0.774, 0.838)
RSF	0.8497	0.0125	(0.825, 0.873)
AFT model (1) with $\mu(\boldsymbol{Z})$ given in (6) and $\sigma = 1.0$			
Cox PH	0.7182	0.0184	(0.684, 0.751)
Log normal	0.7182	0.0185	(0.684, 0.753)
deepAFT	0.7575	0.0176	(0.722, 0.791)
deepAFT-IPCW	0.7351	0.0275	(0.695, 0.775)
deepAFT-Tranform	0.7453	0.0427	(0.695, 0.784)
deepSurv	0.7456	0.0180	(0.708, 0.779)
RSF	0.7747	0.0179	(0.739, 0.807)
PH model (7)			
Cox PH	0.6852	0.0191	(0.648, 0.723)
Log Normal	0.6852	0.0192	(0.646, 0.723)
deepAFT	0.7196	0.0193	(0.682, 0.756)
deepAFT-IPCW	0.7141	0.0197	(0.675, 0.750)
deepAFT-Tranform	0.7129	0.0259	(0.656, 0.750)
deepSurv	0.7084	0.0187	(0.670, 0.745)
RSF	0.7361	0.0177	(0.700, 0.771)
Sparse data with PH model (8)			
Cox PH	0.5006	0.0101	(0.481, 0.521)
Log Normal	0.5007	0.0100	(0.482, 0.522)
deepAFT	0.6960	0.0088	(0.678, 0.713)
deepSurv	0.6977	0.0088	(0.679, 0.714)
RSF	0.6960	0.0090	(0.678, 0.714)

Note: Monte-Carlo standard deviation (sd) and percentiles are calculated across simulation replications.

deepSurv and RSF methods provide almost identical C-index values (0.696 by deepAFT and RSF, 0.697 by deepSurv) and the corresponding 2.5 and 97.5 percentiles of 0.68 and 0.71.

It is worth noting that the AFT model assumption for the proposed deepAFT methods is not satisfied in the second and the third simulation scenario of PH models (7) and (8). However, in Table 1, the proposed deepAFT methods still outperform the regular regression models, and achieve comparable C-indices as the two deep learning methods (deepSurv and RSF). This indicates the proposed deepAFT methods are robust in the presence of model misspecification.

Each boxplot in Figure 1 shows the difference in C-indices between each of the four methods (Cox, LgNormal, deepSurv and RSF) and the first deepAFT method, for data from the first simulation scenario with model (1) specified by

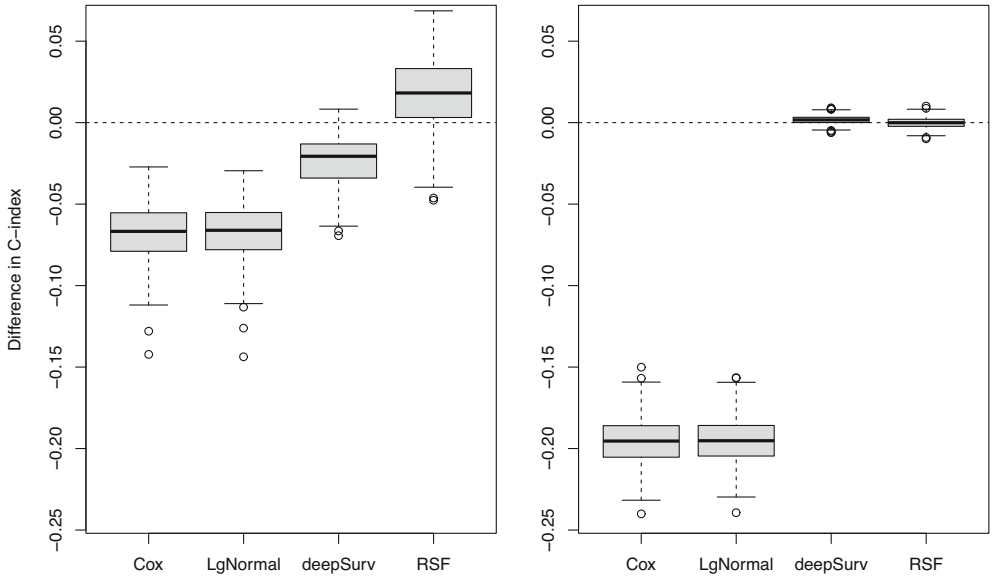

FIGURE 1 Box plots for the differences in C-index between the studied methods and the original deepAFT method, for data from AFT model (1) with $\sigma = 0.5$, $n = 750$ training samples (Left) and sparse data from model (8), $n = 4000$ training samples (Right). Cox stands for the Cox PH regression model, LgNormal for the log normal AFT regression model, deepSurv for the deep neural network method for survival data by Katzman et al, and RSF for the random survival forest method.

TABLE 2 Performance based on mean squared error (AMSE (4)), averaged across $R = 500$ replications, for data simulated from AFT model (1) with $\sigma = 0.5$ and 1.0.

Method	AMSE	sd	(2.5, 97.5) percentiles
Data from model (1) with $\sigma = 0.5$			
Log Norml	0.9266	0.075	(0.798, 1.102)
deepAFT	0.7475	0.060	(0.634, 0.878)
deepAFT-IPCW	0.8061	0.075	(0.688, 0.978)
deepAFT-Tranform	0.8119	0.073	(0.689, 0.967)
Data from model (1) with $\sigma = 1.0$			
Log Normal	1.2037	0.109	(1.034, 1.469)
deepAFT	1.1301	0.093	(0.976, 1.335)
deepAFT-IPCW	1.1621	0.110	(0.977, 1.405)
deepAFT-Tranform	1.1418	0.141	(0.944, 1.357)

Note: Monte-Carlo standard deviation (sd) and percentiles are calculated across simulation replications. The three proposed deepAFT methods are compared to Log Normal AFT regression model.

(6) and $\sigma = 0.5$ (Left), and the third simulation scenario of the sparse data model (8) (Right). The plot further highlights the remarkable improvements of the deepAFT method over the regular regression models in terms of the discriminatory performance, and the comparable performance of deepAFT and the other two deep learning methods.

For the simulation in Table 2, the numbers of neurons in the hidden layers of the ANN models are 6, 10, and 4, respectively. We use ReLU as the activation function for all hidden layers and an identity function for the output layer, respectively. The hyper parameters are tuned using the R function *hyperTuning()* to minimize the cross-validation mean square error. Table 2 compares the prediction performance of the methods through mean squared error, that is, the AMSE of form (4). The three deepAFT methods again remarkably outperform the regular regression models in the simulation scenario with data from the AFT model (1). Among the three proposed deepAFT methods, the first deepAFT method achieves the smallest mean squared errors, with the smallest variability; the deepAFT-IPCW and deepAFT-Transform

have slightly worse but often comparable performance. The Cox regression model, deepSurv and RSF methods are not included in Table 2, because as discussed earlier, it is not straightforward to obtain the mean squared error for these methods.

4 | APPLICATION

We apply the proposed methods to the Canadian Cancer Trials Group (CCTG) LY12 lymphoma trial dataset. LY.12 is a randomized clinical trial for patients with aggressive lymphoma to a treatment with gemcitabine, dexamethasone, and cisplatin (GDP) or to dexamethasone, cytarabine, and cisplatin (DHAP). The trial has demonstrated that GDP is as effective as DHAP and is less toxic at the same time.[47]

In a biomarker study, the gene expression levels of BCL2 and MYC are collected and evaluated in a subset of 84 patients from the LY.12 trial. BCL2 (B-cell lymphoma 2) gene is a founding member of the BCL-2 family of regulator proteins that regulate cell apoptosis. MYC is a family of regulator genes and proto-oncogenes that code for transcription factors. MYC gene often leads to the increased expression of many genes which are involved in cell proliferation, contributing to the formation of cancer.[48] We make use of 10 variables of interest in constructing the prognostic models, including 8 clinical variables and the 2 aforementioned biomarkers. The eight clinical variables included in this analysis are age (continuous), Eastern Cooperative Oncology Group performance status (ECOG status, 0, 1, 2, 3), Lactate Dehydrogenase (LDH) level (continuous), revised International Prognostic Index (rIPI, 0 or 1 vs 2 vs 3+), receive a transplant (Yes vs No), receive the Rituximab treatment (Yes vs No), response to the previous treatment (response with duration greater than one year, response with duration less than or equal to one year vs stable or progress disease), response to the protocol treatment (Yes vs No). Values of each of these potential covariates are normalized to a mean of 0 and a variance of 1 before analysis. Event-free survival (EFS) was the outcome variable in this sub-study. EFS was calculated from the time of enrollment to the time of disease progression (in years), relapse, initiation of new lymphoma therapy, or death from any causes. Patients who were alive and free of the above events were censored at the time of the trial's final analysis. The censoring rate is 25% in the data set.

We first carry out tests for the proportional hazards (PH) assumption among all 84 patients by applying the *cox.zph*() function in R, which are methods developed by Grambsch and Therneau[49] based on Schoenfeld residuals.[50] The results in Table 3 suggest the PH assumption is not valid for the MYC biomarker (p-value <0.005) and two other clinical variables (p-values <0.05), and marginally invalid for the BCL2 biomarker and three other variables (p-values between 0.05 and 0.10), and deemed invalid overall for the model with all these 10 variables (with highly significant p-value <0.0002).

In this application data analysis, we specify the ANN model to have two hidden layers, and the numbers of neurons in the hidden layers to be 8, and 6, respectively. We use ReLU as an activation function for all hidden layers and an identity function for the output layer, respectively. The learning rate, the coefficient for the momentum gradient descent and

TABLE 3 Tests for proportional hazards assumption of the application CCTG LY.12 data.

Covariate	chisq	df	*p*
AGE	0.974	1	0.3238
ECOG status	3.903	1	0.0482
LDH	2.468	1	0.1162
rIPI score	1.060	2	0.5886
Receive a transplant	3.527	1	0.0604
Receive Rituximab treatment	2.795	1	0.0946
Response to previous treatment	7.231	2	0.0269
Response to protocol treatment	2.883	1	0.0895
MYC biomarker	7.904	1	0.0049
BCL2 biomarker	3.039	1	0.0813
GLOBAL	37.630	12	0.0002

TABLE 4 Application to the CCTG LY.12 data.

	C-index (s.e.)		
Method	**Training data**	**Testing data**	**Cross validation (7 folds)**
Cox	0.842 (0.023)	0.764 (0.049)	0.777 (0.058)
deepSurv	0.885 (0.020)	0.783 (0.053)	0.789 (0.060)
deepAFT	0.857 (0.021)	0.813 (0.053)	0.823 (0.046)
deepAFT-IPCW	0.838 (0.023)	0.793 (0.049)	0.785 (0.077)
deepAFT-Tranform	0.913 (0.016)	0.837 (0.044)	0.825 (0.063)
RSF	0.826 (0.022)	0.788 (0.053)	0.795 (0.073)

Note: Cox stands for the Cox PH regression model, deepSurv for the deep neural network method for survival data by Katzman et al, deepAFT, deepAFT-IPCW and deepAFT-Transform for methods proposed in this article, and RSF for the random survival forest method.

the L_2 regularization parameters for this application are tuned using our *hyperTuning()* R function in the *dnn* package for *deepAFT*. We randomly sample 63 (75%) of the patients to form the training set, and use the remaining as the test set, for assessing the performance of the prediction models; we also compare the prediction methods through a 7-fold cross-validation on all the patients. The R function *concordance*() reports the C-index values. For the C-indices reported on the training and test data in Table 4, we also report the standard errors (s.e.) by the built-in infinitesimal jackknife of the *concordance*() function. For the 7-fold cross-validation, we report the cross-validated C-index value as well as the empirical standard error (s.e.) across all 7 folds.

On the training data, the deepAFT-Transform method gives the highest C-index value of 0.91. On the test sets, the first deepAFT and deepAFT-Transform methods achieve better predictions (C-index > 0.81) than the other methods, the deepAFT-IPCW, deepSurv, and RSF all have C-indices around 0.78 and 0.79; similar behavior patterns are seen from the cross-validation evaluations. The deepAFT methods, especially the first deep AFT and deepAFT-Transform, turn out to perform the best in this application scenario.

To evaluate the computation cost of the proposed methods and the other deep learning methods considered, we compare computation time for the cross-validated C-index in this application. In a MacBook Pro computer with an Apple M1 process, it takes 0.687, 0.122, 0.094, 0.625 and 0.390 seconds to run the cross-validation processes for the deepAFT, deepAFT-IPCW, deepAFT-Transform, deepSurv and the RSF methods, respectively. The deepAFT-Transform method is about 7 times faster than the first deepAFT method, and the deepAFT and deepSurv methods have similar computation time. The computation time for the RSF method turns out to be between the deepAFT-IPCW and the deepSurv methods.

5 | DISCUSSION

The proposed deepAFT methods extend the ANN to right censored survival data, and are useful for predicting survival time when the underlying model is an accelerated failure time model. These proposed methods based on the AFT model are natural counterparts of deepSurv.[7] The deepAFT methods rely on the general AFT model structure, while the deepSurv relies on the general PH model structure, neither assumes a specific distribution for the survival time. The proposed methods outperform the existing regression models when the location parameter in the AFT model is a complicated nonlinear function of the covariates; they have similar prediction performance compared to the existing deep learning methods such as the deepSurv and the random survival forest[14] method that have been shown to work well for survival data. Their performance appears robust even when the AFT model assumption is not satisfied.

Overall, the proposed deepAFT methods add powerful new tools for survival predictions in situations where the PH model assumption for the popular Cox model is violated, or nonlinear covariate effects exist. Unlike other deep learning methods that often require one learning implementation for each specified output type, the proposed deepAFT methods are ready to offer various predictions all at once upon completion of the ANN, for different outcome measures such as (mean) survival time and restricted mean survival time, and for different distributional functions such as hazard and survival functions.

The three proposed deepAFT methods each rely on a different imputation or transformation method for censored survival time. The first deepAFT method imputes the error term of a censored response by the conditional mean of the

error distribution based on the current AFT model form. It is considered the optimal way of imputation, but an iteration is required to fit the model. The deepAFT-IPCW method relies on a common idea of re-weighting at the failure time to compensate for the censored responses. The deepAFT-Transform method imputes the censored responses directly based on the imputation scheme suggested by Fan and Gijbels.[36] We find the first deepAFT performs the best with a modest winning edge, although it also has the drawback that it is an iterative method and takes more time in computation. The deep-IPCW only makes an adjustment in the loss function, deep-Transform only requires a transformation, and neither methods require iterations; both are more efficient than the first deepAFT in terms of computation time. One limitation of the deepAFT-IPCW and deepAFT-Transform methods is that they require the censoring time to have a proper distribution, hence are not applicable to situations with degenerated censoring distributions. Among the proposed methods, we recommend the first deepAFT if computation time is not a concern, and recommend deepAFT-Transform if a faster algorithm is preferred. As a future direction, we will extend the deepAFT methods to incorporate other ways of managing censoring, for example, adopting the doubly robust adjustments.[11]

We made the source code for the proposed *deepAFT* methods available in an R package *dnn*. This package was submitted to the Comprehensive R Archive Network-. The data that support the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGEMENTS

The authors wish to thank both the associate editor and the two reviewers for their insightful comments and suggestions. This work was supported in part by the discovery grants from the Natural Sciences and Engineering Research Council of Canada (NSERC). The computation facilities were provided by Digital Research Alliance of Canada.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Bingshu E. Chen https://orcid.org/0000-0001-6139-0696

REFERENCES

1. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer; 2009.
2. Rumelhart D, Hinton G, Williams R. Learning internal representations by error propagation. In: Rumelhart D, McClelland J, eds. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA, USA: The MIT Press; 1986.
3. Anderson J, Rosenfeld E. *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT Press; 1988.
4. LeCun Y, Bengio Y, Hinton G. Deep Learning. *Nature*. 2015;521:436-444.
5. Faraggi D, Simon R. A neural network model for survival data. *Stat Med*. 1995;14:73-82.
6. Liestbl K, Andersen PK, Andersen U. Survival analysis and neural nets. *Stat Med*. 1994;13:1189-1200.
7. Katzman JL, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y. DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Med Res Methodol*. 2018;18:24.
8. Eleuteri A, Tagliaferri R, Milano L, De Placido S, De Laurentiis M. A novel neural network-based survival analysis model. *Neural Netw*. 2003;16:855-864.
9. Chapfuwa P, Tao C, Li C, et al. Adversarial time-to-event modeling. *Proceedings of the 35th International Conference on Machine Learning*. Vol 35. PMLR; 2018:1-14.
10. Giunchiglia E, N Emchenko A, Scharr VDM. RNN-SURV: a deep recurrent model for survival analysis. *International Conference on Artificial Neural Networks*. Vol 1. New York: Springer; 2018:23-32.
11. Steingrimsson JA, Morrison S. Deep learning for survival outcomes. *Stat Med*. 2020;39:2339-2349.
12. Ranganath R, Perotte A, Elhadad N, Blei D. Deep survival analysis. In: Doshi-Velez F, Fackler J, Kale D, Wallace B, Wiens J, eds. *Proceedings of Machine Learning Research: the first Machine Learning for Healthcare Conference*. Vol 56. Boston, MA, USA: Northeastern University; 2016:101-116.
13. Baek ET, Yang HJ, Kim SH, et al. Survival time prediction by integrating cox proportional hazards network and distribution function network. *BMC Bioinformatics*. 2021;22:192.
14. Ishwaran H, Kogalur U, Blackstone E, Lauer M. Random survival forests. *Ann Appl Stat*. 2008;2:841-860.
15. Yousefi S, Amrollahi F, Amgad M, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Nat Sci Rep*. 2017;7:11707.
16. Che Z, Purushotham S, Khemani R, Liu Y. Interpretable deep models for ICU outcome prediction. *AMIA Ann Symp Proc*. 2017;e2016:371-380.
17. Hao J, Kim Y, Mallavarapu T, Oh JH, Kang M. Interpretable deep neural network for cancer survival analysis by integrating genomic and clinical data. *BMC Med Genet*. 2019;12:1-13.

18. Mobadersany P, Yousefi S, Amgad M, et al. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proc National Acad Sci*. 2018;115:2970-2979.
19. Matsuo K, Purushotham S, Jiang B, et al. Survival outcome prediction in cervical cancer: Cox models vs deep-learning model. *Am J Obstet Gynecol*. 2019;220:381.
20. Tashi QA, Saad MB, Sheshadri A, et al. SwarmDeepSurv: swarm intelligence advances deep survival network for prognostic radiomics signatures in four solid cancers. *Patterns*. 2023;4:100777.
21. Zhong Q, Mueller J, Wang JL. Deep learning for the partially linear Cox model. *Ann Stat*. 2022;50:1348-1375.
22. Wiegrebe S. Deep learning for survival analysis: a review. *Artif Intell Rev*. 2024;57:1-30.
23. Wang P, Li Y, Reddy CK. Machine learning for survival analysis: a survey. arXiv:1708.04649v1 2017: 1-39.
24. Spooner A, Chen E, Sowmya A, et al. A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. *Nat Sci Rep*. 2020;10:20410.
25. Hao L, Kim J, Kwon S, Ha ID. Deep learning-based survival analysis for high-dimensional survival data. *Mathematics*. 2021;9:1244.
26. Salerno S, Li Y. High-Dimensional Survival Analysis: Methods and Applications. *Annu Rev Stat Appl*. 2023;10:25-49.
27. Huang Y, Li J, Li M, Aparasu RR. Application of machine learning in predicting survival outcomes involving real-world data: a scoping review. *BMJ Medical Res*. 2023;23:268.
28. Sargent DJ. Comparison of artificial neural networks with other statistical approaches. *Cancer*. 2001;91:1636-1642.
29. Cox DR. Regression models and life tables (with discussion). *J R Stat Soc Ser B*. 1972;34:187-220.
30. Cox DR. Partial likelihood. *Biometrika*. 1975;62:269-276.
31. Lawless JF. *Statistical Models and Methods for Lifetime Data*. Ltd: John Wiley and Sons; 2002.
32. Buckley J, James I. Linear regression with censored data. *Biometrika*. 1979;66:429-436.
33. Miller R, Halpern J. Regression with censored data. *Biometrika*. 1982;69:521-531.
34. Lin DY. Linear regression analysis of censored medical costs. *Biostatistics*. 2000;1:35-47.
35. Robins JM, Rotnitzky A. AIDS epidemiology. In: Jewell NP, Dietz K, Farewell VT, eds. *Recovery of Information and Adjustment for Dependent Censoring Using Surrogate Marker*. Boston, MA: Springer; 1992:297-331.
36. Fan J, Gijbels I. Censored regression: Local linear approximations and their applications. *J Am Stat Assoc*. 1994;89:560-570.
37. Robins JM, Rotnitzky A, Zhao LP. Estimation of regression coefficients when some regressors are not always observed. *J Am Stat Assoc*. 1994;89:846-866.
38. Jin Z, Lin DY, Wei LJ, Ying Z. Rank-based inference for the accelerated failure time model. *Biometrika*. 2003;90:341-353.
39. Bang H, Robins J. Doubly robust estimation in missing data and causal inference models. *Biometrics*. 2005;61:962-973.
40. Hahnloser R, Sarpeshkar R, Mahowald M, Douglas R, Seung H. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*. 2000;405:947-951.
41. Kaplan EL, Meier P. Nonparametric estimation from incomplete observations. *J Am Stat Assoc*. 1958;53:457-481.
42. Robbins H, Monro S. A Stochastic Approximation Method. *Ann Math Stat*. 1951;22:400-407.
43. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *J Mach Learn Res*. 2012;13:281-305.
44. Bang H, Tsiatis AA. Estimating medical costs with censored data. *Biometrika*. 2000;87:329-343.
45. Leurgans S. Linear models, random censoring and synthetic data. *Biometrika*. 1987;74(2);301-309.
46. Harrell FE Jr, Califf RM, Pryor DB, Lee KL, Rosati RA. Evaluating the yield of medical tests. *J Am Med Assoc*. 1982;247:2543-2546.
47. Crump M, Kuruvilla J, Couban S, et al. Randomized comparison of gemcitabine, dexamethasone, and cisplatin versus dexamethasone, cytarabine, and cisplatin chemotherapy before autologous stem-cell transplantation for relapsed and refractory aggressive lymphomas: NCIC-CTG LY.12. *J Clin Oncol*. 2014;32:3490-3496.
48. Bosch M, Akhter A, Chen B, et al. Abioclinical prognostic model using MYC and BCL2 predicts outcome in relapsed/refractory diffuse large B-cell lymphoma. *Haematologica*. 2018;103:288-296.
49. Grambsch P, Therneau T. Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*. 1994;81:515-526.
50. Schoenfeld D. Partial residuals for the proportional hazards regression model. *Biometrika*. 1982;69:239-241.

How to cite this article: Norman PA, Li W, Jiang W, Chen BE. deepAFT: A nonlinear accelerated failure time model with artificial neural network. *Statistics in Medicine*. 2024;43(19):3689-3701. doi: 10.1002/sim.10152