Default risk prediction and feature extraction using a penalized deep neural network

Cunjie Lin1,2 · Nan Qiao2 · Wenli Zhang2 · Yang Li1,2,3 · Shuangge Ma2,4

Received: 28 April 2022 / Accepted: 26 August 2022 / Published online: 15 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Online peer-to-peer lending platforms provide loans directly from lenders to borrowers without passing through traditional financial institutions. For lenders on these platforms to avoid loss, it is crucial that they accurately assess default risk so that they can make appropriate decisions. In this study, we develop a penalized deep learning model to predict default risk based on survival data. As opposed to simply predicting whether default will occur, we focus on predicting the probability of default over time. Moreover, by adding an additional one-to-one layer in the neural network, we achieve feature selection and estimation simultaneously by incorporating an \( L_1 \)-penalty into the objective function. The minibatch gradient descent algorithm makes it possible to handle massive data. An analysis of a real-world loan data and simulations demonstrate the model’s competitive practical performance, which suggests favorable potential applications in peer-to-peer lending platforms.

Keywords Feature extraction · Loan data · Neural network · Risk prediction · Survival analysis

1 Introduction
The assessment of loan default risk is crucial for financial institutions such as banks and lending platforms (Everett 2015). Compared to traditional bank deposits, online peer-to-peer (P2P) lending platforms, such as the Lending Club, enable investors to obtain a higher return on investment. In recent years, the P2P lending market has experienced a substantial increase in popularity. In 2014, the P2P lending generated approximately $5.5 billion worth of loans in the USA according to PricewaterhouseCoopers (2015), and the market may reach $150 billion or higher by 2025 (Calabrese et al. 2019). There is a risk, however, that loans and interest may not be repaid due to information asymmetry between borrowers and lenders, which can lead to adverse selection and moral hazard problems (Lin et al. 2017). It was reported that the UK COVID loan scheme could lose 26 billion pounds from fraud, defaults, and “Criminals may have made billions of pounds by exploiting a government loan scheme to help British businesses hit by the coronavirus, while large losses are expected from firms unable to repay,” according to the country’s spending watchdog. A rational strategy for reducing such losses is to make appropriate loan decisions based on accurate default risk prediction. Therefore, it is particularly important for lenders to predict the probability of default to avoid unsecured borrowers and allocate investment more efficiently (Jagtiani and Lemieux 2019). Meanwhile, such a problem has also attracted considerable interest from statisticians to develop feasible credit risk analysis methodologies (Ding et al. 2012; Yuan et al. 2018).

Traditional statistical models for credit risk generate a credit score for each borrower that is indicative of whether the borrower can repay the loan (Babaev et al. 2019; Zheng et al. 2019). For example, Li et al. (2022) use a logistic regression model to estimate the probability of default, while considering the highly imbalanced distributions of the event of interest (default). Xu et al. (2021) study the estimation of credit risk with logistic regression augmented with information on observed interfirm financial transactions for small
and medium-sized enterprises. And logistic regression is also applied in Costa e Silva et al. (2020) to evaluate the default risk of consumer loans. Although these models are popular and useful, they have limitations. First, those based on logistic regression cannot take full advantage of the temporal information in time-to-event data, as these models use the indicator variable of default but ignore a borrower’s time of default. Second, the traditional models only generate a risk score, whereas users can be equally interested in the time of default and default probability over time. In fact, it has been shown that early intervention under guidance can both effectively minimize arrears and reduce the number of accounts that become bad debts and result in losses (Wang et al. 2018; Li and Chen 2021). Accordingly, more informative predictive outcomes are required for early guidance as well as for improved efficiency in post-loan risk management (Dirick et al. 2017). This has motivated us to develop credit risk models with the aid of survival analysis techniques.

Several studies have examined credit risk assessment using survival analysis techniques (Baesens et al. 2005; Stepanova and Thomas 2002). For example, Hassan et al. (2018) incorporate time-varying macroeconomic variables into their modeling and show that survival analysis has better prediction performance than logistic regression. Jaber et al. (2017) estimate the probability of default by fitting right-censored data from a bank in Jordan with both parametric and non-parametric survival models. To deal with the challenges of high-dimensional predictors, Wan et al. (2019) use the Cox’s proportional hazards (CPH) model with a Lasso penalty to select important factors for P2P loan repayment risk. However, the aforementioned model assumptions and other existing approaches, such as proportional hazards, linearity in coefficients, and independent censoring, may not be tenable in real applications (Wang and Li 2019).

Compared to the aforementioned approaches, machine learning methods, such as neural networks and random forests, are much more powerful due to their flexibility and high-quality prediction under very weak model assumptions. As pointed out in Murdoch et al. (2019), “Machine-learning models have demonstrated great success in learning complex patterns that enable them to make predictions about unobserved data.” Recently, studies have introduced neural networks to improve the performance of traditional survival models. They partially retain the assumption of the CPH model, which is also referred to as the linear proportional hazards condition. For example, Katzman et al. (2018) combine deep neural network (DNN) with the CPH model to develop DeepSurv, which they use to predict the risk of an event occurrence for patients and provide personalized treatment recommendations. Ching et al. (2018) propose the Cox-nnet algorithm with a two-layer neural network, where one hidden layer implements the dimension reduction of input covariates, and the output layer performs the Cox regression based on the activation levels of the hidden layer. Hao et al. (2021) apply the Cox-based DNN prediction survival model to high and ultrahigh dimensional survival data. This Cox-like model can be obtained by replacing the linear combination of covariates in its log-likelihood function with the output layer function in the neural network. These methods are nonlinear extensions of the CPH model, and most of the advantages of the CPH model are preserved.

In applications, however, survival models that maintain parts of the proportional hazards structure do not fit survival data with nonlinear risk functions well. Some researchers have considered modeling non-proportional hazards with neural networks. One example is Nnet-survival, which is proposed in Gensheimer and Narasimhan (2019). This is a discrete-time survival model with neural networks, and its baseline hazard rate and the effect of input data on hazard can vary with follow-up time. Nezhad et al. (2019) propose a deep active survival analysis approach, which is a new survival analysis framework that uses deep learning and active learning with a new sampling strategy, and this approach has better performance in treatment recommendation for new patients. Lee et al. (2018) introduce an approach called Deep-Hit, which transforms the problem of learning the distribution of survival time into a discrete-time classification problem and employs a deep neural network to estimate the survival function.

This study is motivated by the appealing flexibility and prediction accuracy of deep learning, which has made extraordinary achievements in many applications, such as computer science, personalized treatment recommender system, and voice/image recognition. As pointed out by Jordan and Mitchell (2015), “Machine learning addresses the question of how to build computers that improve automatically through experience. It is one of today’s most rapidly growing technical fields, lying at the intersection of computer science and statistics, and at the core of artificial intelligence and data science.” But it has not been well developed in predicting default risk, especially in the framework of survival analysis. The goal of this study is to further expand the paradigm of default risk prediction via penalized deep learning based on time-to-event data. This study can contribute beyond the existing literature in the following aspects. First, it significantly differs from traditional classification studies by making full use of the temporal information of time-to-event data. Instead of merely predicting whether default will occur, the proposed method provides more detailed information on the probability of default over time in the framework of survival analysis. Second, this study proposes to improve default risk prediction with survival data via deep learning as opposed to model-based methods. It can also identify important features, which can provide guidance to P2P lending platforms regarding post-loan risk management. Although the CPH model with penalization can be applied effectively to
accommodate small samples, when dealing with large-scale settings, these methods no longer have computational superiority. Third, the network structure in this study is different from that of other neural networks for survival analysis. For example, the output layer has multiple nodes, which can be converted to the estimate of survival probability. By adding an additional one-to-one layer, we can achieve feature extraction and estimation simultaneously through the incorporation of an $L_1$-penalty into the objective function, which is unavailable in Nnet-survival (Ching et al. 2018) and other deep learning models with survival data. As most research on deep learning has not paid sufficient attention to interpretability, we, instead, pay more attention to features’ statistical importance. Moreover, computation is made more efficient via the minibatch gradient descent algorithm, which splits the training set into smaller batches and implements gradient descent on each one. This makes it possible to accommodate massive data.

The rest of this paper is divided into four sections. Section 2 outlines the proposed approach, called Survival Analysis with Feature Extraction via Deep Neural Network (SAFE-DNN). In Sect. 3, a real-world loan dataset is converted to the estimate of survival probability. By adding an additional one-to-one layer, we can achieve feature extraction, which is the survival time, which is the time of default in this study.

$\lambda(t) = \lim_{h \to 0^+} \frac{P(t \leq T < t + h \mid T \geq t)}{h},$

which is the conditional probability that a borrower defaults at time $t$ given that he or she has not defaulted up to that point. For the Cox’s proportional hazards model, the hazard function is assumed to have the form:

$\lambda(t \mid X_i) = \lambda_0(t) \exp(\beta^T X_i),$

where $\lambda_0(\cdot)$ is the baseline hazard function, and $\beta$ is the $p$-dimensional coefficient vector. As stated in Sect. 1, the linear proportional hazards condition can be restrictive in application. In this study, we propose a means of prediction for default risk via deep learning.

What is proposed in this study is a multi-layer network, including one layer of input nodes, one layer of output nodes, and multiple layers of hidden nodes. A schematic presentation of the structure is shown in Fig. 1a. Consider a vector of covariates $X_i = (x_{i1}, ..., x_{ip})^T$ for the $i$-th borrower as input to the network. The $j$-th input node takes value $x_{ij}$ for this input vector ($j = 1, ..., p$). Each input is connected directly to only one node of the weighted input layer, and a weight $w_j$ is associated with the connection between input $x_{ij}$ and the corresponding weighted input node. As suggested in the literature, we consider the scenario where some input features are non-informative for predicting default risk. Correspondingly, the connected coefficients, $w = (w_1, ..., w_p)^T$, are assumed to be sparse with some zero values. In all hidden layers, each node is fully connected to the nodes in the next layer. There are $K$ hidden layers, and the connected weights of the $k$-th hidden layer are collectively denoted by $W^{(k)}$. The activation function is set as the popular Sigmoid function.

For the output layer, there are $m$ nodes, and the output value is $(y_1, ..., y_m)^T$, which is related to the prediction of survival probability. Specifically, we divide the follow-up time into $m$ intervals, with $t_1, t_2, ..., t_m$ being their right endpoints. Additionally, the intervals are left-open and right-closed with $I_1 = (0, t_1], I_2 = (t_1, t_2], ..., I_m = (t_{m-1}, t_m]$. Let $p_k$ be the probability of default in the $k$-th interval, which can be estimated by the output vector, i.e.,

$\hat{p}_k = \frac{\exp(y_k)}{\sum_{j=1}^{m} \exp(y_j)}, \quad k = 1, ..., m.$

Now the survival function, $S(t) = P(T \geq t)$, of the default time can be estimated by:

$\hat{S}(t) = 1 - \sum_{k=1}^{j} \hat{p}_k, \quad t \in I_j.$

Here, the cut-points for the time intervals can be equally spaced. As suggested in Gensheimer and Narasimhan (2019), another option is to space the time intervals more widely apart, with increasing follow-up time, especially when the frequency of events decreases with follow-up time. Compared with equal intervals, varying intervals ensure that the number of events in the different intervals is more or less comparable, which may help improve estimation. In this study, we use $m = 20$ to $30$ time intervals, and the cut-points are determined by:
$t_j = \frac{-\log(1 - 0.03 j) \cdot Y^*}{\log(2)}, \quad j = 1, 2, \ldots, m,$

where $Y^* = 0.27Y_{\text{max}}$ and $Y_{\text{max}}$ is the maximum observed survival time.

### 2.2 Objective function

We set the loss function as:

$$L = -\sum_{i=1}^{n} \log(L_i),$$

where $L_i$ is the likelihood of the $i$-th subject. To specify the likelihood, we consider two cases according to whether the observation is a failure or is censored. First, if observation $i$ is a failure in interval $j$ (i.e., $Y_i \in I_j$ and $\delta_i = 1$), the likelihood is the probability of failing in this interval:

$$L_i = p_{ij},$$

where $p_{ij}$ is the probability of default in interval $j$ for individual $i$. Second, if individual $i$ is censored, with the censoring time falling in the second half of interval $j - 1$ or the first half of interval $j$, that is,

$$(t_{j-2} + t_{j-1}) / 2 \leq Y_i < (t_{j-1} + t_j) / 2, \quad \text{and} \quad \delta_i = 0,$$

the failure time is treated as being in the $j$-th interval or later. This means that if the subject is censored in the second half of an interval, it is considered to be alive in this interval. By doing this, we can avoid a downward bias of survival estimates (Brown et al. 1997). Hence, the likelihood is the probability of surviving through intervals 1 to $j - 1$:

$$L_i = 1 - \sum_{l=1}^{j-1} p_{il}. $$

For the ease of calculation, we convert the observed $(Y_i, \delta_i)$ $i = 1, \ldots, n$ into two vectors, $\text{surv}$ and $\text{fail}$, both of length $m$. Specifically, for an uncensored subject with failure time $t$ in the $j$-th interval, the $j$-th component of $\text{surv}$ and $\text{fail}$ are defined, respectively, as

$$\text{surv}(j) = 1,$$

$$\text{fail}(j) = \begin{cases} 1, & \text{if } t_{j-1} \leq t < t_j, \\ 0, & \text{otherwise} \end{cases}.$$
\( \text{surv}(j) = \begin{cases} 1, & \text{if } t \geq \frac{1}{2}(t_{j-1} + t_j), \\ 0, & \text{otherwise} \end{cases} \)

\( \text{fail}(j) = 0. \)

Then, for the \( i \)-th individual, the log-likelihood function is defined as:

\[
l_i = \log \left( 1 - \sum_{j=1}^{m} \text{surv}_i(j) \cdot p_{ij} + \sum_{j=1}^{m} \text{fail}_i(j) \cdot p_{ij} \right).
\]

(1)

More details about formula (1) are elaborated in Appendix.

To implement the selection of important features, we first consider the penalized objective function:

\[
Q_1 = -\sum_{i=1}^{n} l_i + \lambda_1 \sum_{i=1}^{p} |w_i|,
\]

where \( w_i \) is the connected weight of the \( i \)-th neuron in the one-to-one layer, and \( \lambda_1 \) is the tuning parameter and controls the sparsity of parameters. It can be selected according to the loss function of a validation set. Further, the overfitting issue of deep learning can be addressed by adding a regularization term: In this study, \( L_2 \) penalization is applied, and the final objective function is:

\[
Q = -\sum_{i=1}^{n} l_i + \lambda_1 \sum_{i=1}^{p} |w_i| + \lambda_2 \sum_{k=1}^{K+1} \| W^{(k)} \|^2,
\]

(2)

where \( W^{(k)} \)'s are the connected coefficients in the hidden layers, and \( \lambda_2 \) is the tuning parameter to control model complexity. Here, we use \( L_2 \) regularization for preventing overfitting, and it yields satisfactory performance, as evidenced by our numerical studies. We note that, other techniques, such as early stopping (Li et al. 2020), \( L_1 \) regularization (Mollalo et al. 2019), and Monte Carlo dropout (Hinton et al. 2012; Gal and Ghahramani 2016), can also be used for addressing the problem of overfitting. It is beyond our scope to evaluate which technology is more efficient.

### 2.3 Statistical feature importance

In addition to using neural network for prediction, interpretation and inference has been attracting increasing attention. However, it is difficult to define a straightforward measurement for statistical feature importance due to the complex structure of neural network (Murdoch et al. 2019). A possible way is to use the local interpretable model-agnostic explanation (LIME) method (Sun et al. 2020), which provides prediction importance for each variable and each subject by perturbing the feature values and evaluating how the prediction results change. It is computationally demanding to perform LIME. Moreover, it may not be very reasonable to fit a linear regression model between the perturbed feature values and their corresponding output values, especially for the output of SAFE-DNN with multiple nodes.

Intuitively, the feature with very small \( w_i \) has a weak signal in the following transmission of the neural network, and its influence on the final output, equivalently the survival probability, is insignificant. Hence, we select the features with relatively large weights in the one-to-one layer by penalizing \( w_i \) in the objective function \( Q \). Nonetheless, it does not mean that the selected features would definitely have clear implications for the final outputs. For rigorously assessing the relative importance of features, we adopt the concept of gradient for a function. Consider the feature vector \( X = (X_1, \ldots, X_p)^\top \) as an input to the SAFE-DNN network. The corresponding output vector is denoted as:

\[
g(X) = (g_1(X), \ldots, g_m(X))^\top,
\]

where \( g_j(X) \) is the output value in the \( j \)-th interval, \( j = 1, \ldots, m \). Then for interval \( I_j \), the “gradient” of \( g_j(X) \) with respect to the \( i \)-th feature \( X_i \) is defined as:

\[
\frac{\partial}{\partial X_i} g_j(X) = \frac{g_j(X_1, \ldots, X_{i-1}, X_i + \Delta X_i, X_{i+1}, \ldots, X_p) - g_j(X_1, \ldots, X_p)}{\Delta X_i},
\]

(3)

where \( \Delta X_i \) is a small disturbance. Similarly, we can define the “gradient” of \( g(X) \) with respect to \( X_i \) as:

\[
\frac{\partial}{\partial X_i} g(X) = \frac{[g(X_1, \ldots, X_{i-1}, X_i + \Delta X_i, X_{i+1}, \ldots, X_p) - g(X_1, \ldots, X_p)]}{\Delta X_i},
\]

(4)

where \( \| \cdot \| \) is the \( L_2 \) norm of a vector. The “gradient” \( \frac{\partial}{\partial X_i} g(X) \) is always positive, measures the change of the output caused by a feature’s disturbance, and can be regarded as a reasonable measurement for assessing the relative influence of a feature.

With traditional statistical models, such as CPH, we can compute confidence intervals for the regression coefficients and make inference for the statistical feature importance. Similarly, we can compute the “gradient” \( \frac{\partial}{\partial X_i} g(X) \) for all observations and use the sample confidence interval for making inference. It is interesting to note that the proposed “gradient” measures the influence of a feature on the whole neural network, while the weight \( w_i \) reflects its impact on the first layer. Comparatively, the “gradient” provides a more comprehensive index for measuring a feature’s effect on the survival probability. In this study, we propose to select the features via penalizing the weights \( w_i \) instead of “gradient,” since it is more practical and easier to implement penalizing the weights than the “gradient.” Moreover, the feature with a small weight will typically have little impact on the following network transmission, and possibly result in relatively small “gradient.” In this sense, penalization of weights \( w_i \)'s would not miss the features with large “gradient.” Considering the computational complexity of selecting features via “gradient,” we suggest using weights \( w_i \)'s for implementing feature selection. Nonetheless, the proposed “gradient” is a
feasible measurement for inferring the statistical importance of features.

2.4 Minibatch gradient descent algorithm

The framework of the proposed SAFE-DNN is described in Fig. 1b. To train the model efficiently, we propose using the minibatch gradient descent algorithm (Khirirat et al. 2017). The detailed algorithm is described in Algorithm 1.

Algorithm 1 SAFE-DNN training algorithm

Input: \( (Y_i, \hat{t}_i, X_i), i = 1, 2, \ldots, n \)
Output: \((\hat{p}_1, \hat{p}_{i2}, \ldots, \hat{p}_{iM}), i = 1, 2, \ldots, n\)
1: Randomly divide data into \(B\) subsets. Use \(B - 1\) subsets for training, and 1 for validation;
2: \(B\)
3: for \(k\) in \(1, 2, \ldots, L\) do
4: \(1)\) Select a set of hyper parameters;
5: \(2)\) Set the initial value of the model;
6: \(3)\) Set \(loss^{opt} = +\infty, n^{impro} = 0, n^{epoch} = 0, \eta^k = 0.005\);
7: \(4)\)
8: while \(n^{epoch} < E\) do
9: Divide all training data into small sample sets with sample size \(batch\_size\);
10: for batches \(1, \ldots, S\) do
11: Update parameter: \(\beta^{k+1} = \beta^k - \eta^k Q(\beta^k) + k(\beta^k - \beta^{k-1})\);
12: end for
13: Calculate \(loss\) for the validation set, that is, the value of \(Q\) in (2):
14: if \(loss < loss^{opt}\) then
15: Update \(loss^{opt} = loss\), and keep \(n^{not\_improved} = 0\);
16: else
17: Set \(n^{not\_improved} = n^{not\_improved} + 1\);
18: end if
19: if \(n^{not\_improved} = n^{stop}\) then
20: Break;
21: end if
22: if \(n^{epoch} < 100\) then
23: Reduce learning rate: \(\eta = \eta \times 0.8, n^{epoch} = n^{epoch} + 1\);
24: end if
25: end while
26: end for
27: end while
28: When \(loss^{opt}\) reaches its minimum, the best model is obtained.

Briefly, the training samples are randomly divided into smaller groups. In our numerical study, the sample size of each group is equal to \(batch\_size = 200\), and the number of batches \(S\) depends on the size of data and batch size. The model parameters are updated on these small sample sets. Here \(loss^{opt}\) is the optimal loss of the validation set. When the loss does not improve persistently in the validation set, we use \(n^{not\_improved}\) to record the present iteration number. The maximum number of iterations, \(E\), is set as 10,000. In addition, the number of subsets, \(B\), is chosen as 10, and \(L\) equals 30.

Here, the derivation of the objective function is calculated with the back propagation algorithm. In \(\beta^{k+1} = \beta^k - \eta^k Q(\beta^k) + k(\beta^k - \beta^{k-1})\), the gradient of loss function, \(Q(\beta^k)\), is calculated using the Theano module in Python, which omits the specific derivation of gradient. Thus, we define the objective function according to Theano. The learning rate, \(\eta^k\), and momentum term, \(k(\beta^k - \beta^{k-1})\), are tuned to optimize the minibatch gradient descent algorithm and make the model converge faster and more stably. In our numerical implementation, all variables have the same learning rate, and its initial value is 0.005. It is multiplied by 0.8 every 100 iterations. Moreover, it decreases as the iteration number increases to improve convergence efficiency. The momentum, \(\beta^k - \beta^{k-1}\), records the direction of the last iteration coefficient to avoid a local optimum.

The hyper parameters \(\lambda_1\) and \(\lambda_2\) are selected by cross-validation. Note that \(\lambda_2\) is used to control model complexity and avoid overfitting. A larger value of \(\lambda_2\) corresponds to a simpler model. We also use early stopping to avoid overfitting. In this study, we set the early stopping parameter as \(n^{stop} = 300\) to achieve “double safety.” We stop training if the loss function of the validation set does not decrease.

2.5 Advancements from existing works

In this study, we stress solving the practical problem of predicting credit risk with deep learning. Although deep learning has had extensive applications, it has not been fully developed in survival analysis. In SAFE-DNN, the traditional proportional hazards condition is discarded so that it can better accommodate flexible relationships between features and hazard function. A network structure with a one-to-one layer is designed to achieve feature selection, which is unavailable in Nnet-survival. Unlike in the existing deep learning studies, we pay more attention to statistical feature importance and propose a new concept of feature importance that has meaningful interpretations and is much easier to implement than LIME. Meanwhile, there are both similarities and differences between the proposed SAFE-DNN model and traditional survival models.

First, they share the common target of predicting survival/hazard function based on time-to-event data, but with different model assumptions. In the SAFE-DNN model, some model assumptions, such as the proportional hazards assumption, are discarded, and the proposed neural network allows us to learn highly complex and nonlinear relationships between features and hazard function. The outputs of a SAFE-DNN neural network are the probabilities of default in divided intervals, which can be organized into survival/hazard function as the traditional statistical models can provide. However, the default probabilities in different intervals may be more informative for applications owing to their straightforward interpretations.
Second, the most commonly used technique—maximum likelihood estimation as in traditional survival models—is also used here to train the SAFE-DNN network. It is noted that we use the “full” likelihood in the loss function, which can be more efficient than the partial likelihood used in estimating CPH. Meanwhile, we implement feature selection through penalizing parameters $w_i$, $i = 1, \ldots, p$ in the one-to-one layer, just like penalizing coefficients in the CPH model. Somewhat differently, the objective function of SAFE-DNN includes a regularization term on $W(k)$ to overcome the overfitting issue of deep learning.

Third, unlike the coefficients in a CPH model, the parameters in a SAFE-DNN network have no straightforward interpretations, but the “gradient” defined in (4) has some implications of statistical feature importance. Thus, we can infer the influence of a feature through its “gradient,” similar to a coefficient in CPH.

Fourth, the minibatch gradient descent algorithm used in SAFE-DNN is a variation of the gradient descent algorithm, which is often used for finding weights of machine learning algorithms or coefficients of statistical models. With the essence of gradient descent, the minibatch gradient descent algorithm splits the training data into small batches that are used to calculate model error and update parameters, which allows the efficiency of not having all training data in memory. Thus it provides a computationally more efficient process and more robust convergence for the optimization and then enables us to handle big data.

3 Analysis of loan data from Lending Club

3.1 Data description

The Lending Club is one of the largest P2P lending marketplaces in the U.S. This dataset contains more than 90k records with 280 features, such as borrower credit characteristics, borrower assessments, loan characteristics, borrower indebtedness, credit history, and status of loan (default or non-default). This dataset is publicly available at the Lending Club website (https://www.lendingclub.com).

We analyze data from the first quarter of 2017 and note that a larger-scale analysis is also possible. Before analysis, we delete features that have missing rates of more than 60% or more than 95% of subjects have the same values. For continuous variables, the missing values are imputed with medians. For categorical variables, the missing values are imputed with modes. In addition, categorical variables are recoded with numerical coding or dummy variables. Finally, we have a total of 96,781 observations and 114 explanatory variables. In the analysis, the survival time is defined as the length of the sustained repayment period, which is calculated by dividing the total repayment amount by that of each period. If borrowers have fully paid or are still in the repayment period, their event times are right-censored, as default events have not happened. A total of 15,670 subjects have default events by the end of the study.

3.2 Results

The flowchart of data analysis is shown in Fig. 2. For comparison, we consider the following alternatives: Nnet-survival-$L_1$, which is based on Nnet-survival (Gensheimer and Narasimhan 2019) and revised with an $L_1$ penalty to accommodate high-dimensional features, and Cox-DAC, which is proposed by Wang et al. (2018). Cox-DAC can deal with massive data where the sample size, $n$, is exceedingly large and the covariate dimension, $p$, is not small but $n \gg p$. Both the SAFE-DNN and Nnet-survival-$L_1$ are trained with the minibatch gradient descent algorithm, and the tuning parameter of the Cox-DAC method is selected by minimizing the BIC score.

When evaluating the prediction accuracy of different methods, the following measure is used:

C-index The concordance index (C-index) is a rank correlation that counts concordant pairs between the predicted and true outcomes (Uno et al. 2011). The value of C-index is between 0 and 1, and 1 means an ideal prediction. We use the Kaplan–Meier estimate (Fig. 5 in the Appendix) of the survival function as the true value when calculating the C-index.

To evaluate the prediction and identification performance, the dataset is randomly divided into two parts: 70% of the data is used for training, and the other 30% for testing. This procedure is repeated 100 times. We calculate the C-index in period 15, 35, and 55. We find that SAFE-DNN and Cox-DAC share the same C-index value (0.679 for period 15, 35, 55), which is larger than that of Nnet-survival-$L_1$ (0.675 for period 15 and 35, 0.676 for period 55).

The results of feature extraction are shown in Table 1. The SAFE-DNN method identifies 11 features, and Nnet-survival-$L_1$ and Cox-DAC identify 11 and 10 features, respectively. All the features belong to one of the four categories: loan contract contents, asset strength, external credit rating, and external credit situation, which have been suggested to play important roles in credit risk (Dirick et al. 2017). Among them, all three methods select listed amount of loan applied for by a borrower, interest rate on loan, and number of mortgage accounts, which have been found to be positively related to default risk (Shi et al. 2018). SAFE-DNN identifies number of inquiries in the past six months, number of personal finance inquiries, number of credit inquiries in the past 12 months and the home ownership status provided by the borrower, which are missed by both Nnet-survival-$L_1$ and Cox-DAC. A literature search (Hackbarth et al. 2006) suggests that these features may have important implications.
Fig. 2 Flowchart of loan data analysis for credit risk, which may provide support to the validity of the proposed approach. In addition, Cox-DAC identifies five features that are absent from the selection results of SAFE-DNN and Nnet-survival-$L_1$, while Nnet-survival-$L_1$ identifies more unique features.

### 3.3 Evaluation

With real data, it is difficult to objectively evaluate identification accuracy. To evaluate the stability of feature extraction, we calculate the observed occurrence index (OOI, Huang and Ma (2010)) for each variable. Specifically, we randomly select 80% of the subjects and perform the algorithm to identify features using the selected samples. This procedure is repeated 100 times. For the $j$-th feature, OOI is defined as $o_j = c_j/100$, where $c_j$ is the number of times that the feature is identified. Hence, OOI measures the relative importance and stability of the selected feature. Overall, a larger OOI indicates higher stability. The results are plotted in Fig. 3. Each row represents one feature, and each vertical column represents one repeat. A black spot means the feature is selected in the corresponding repeat, and otherwise it is blank. The mean values of OOI for SAFE-DNN, Nnet-survival-$L_1$, and Cox-DAC are 0.825, 0.760, and 0.806, respectively. Combined with the results in Fig. 3, we can conclude that the proposed method has prominent performance in selection stability, while Cox-DAC is slightly worse, and Nnet-survival-$L_1$ is the most unstable method among the three.

For assessing the relative importance of the selected variables, we calculate “gradient” based on (4). Specifically, we take the disturbance $\Delta x_{ij} = 0.01 \cdot x_{ij}$. For the selected features in Table 1, we calculate the “gradient” $G(X_j)$, $j = 1, \ldots, 22$ for all 96,781 subjects, and the boxplot and corresponding 95% confidence intervals are shown in Fig. 4a and b, respectively. Here, we focus on the relative sizes of “gradient” instead of their absolute values. As shown in Fig. 4a, the SAFE-DNN selected features have larger “gradient” values than most of the other features, which means that the SAFE-DNN selected features have a higher influence on the survival probability. We also note that $x_{13}$ has a relatively large “gradient” value, suggesting that SAFE-DNN may still miss important features. Nonetheless, it selects most features with large “gradient.” A similar conclusion can be reached...
Table 1 Data analysis: feature extraction results

| Category | Label | Variable | SAFE-DNN | Nnet-survival-$L_1$ | Cox-DAC |
|----------|-------|----------|----------|---------------------|---------|
| A        | $x_1$ | Listed amount of loan applied for by a borrower. | ✓        | ✓                   | ✓       |
| A        | $x_2$ | Interest rate on loan. | ✓        | ✓                   | ✓       |
| A        | $x_3$ | Number of payments on loan. | ✓        | ✓                   | ✓       |
| B        | $x_4$ | Self-reported annual income. |          |                     | ✓       |
| B        | $x_5$ | The home ownership status provided by the borrower. | ✓        |                     |         |
| B        | $x_6$ | Job title when applying for loan. | ✓        |                     |         |
| B        | $x_7$ | Total collection amount ever owed. | ✓        |                     |         |
| B        | $x_8$ | Number of financial trades. | ✓        |                     |         |
| B        | $x_9$ | Number of trades opened in the past 24 months. | ✓        |                     | ✓       |
| B        | $x_{10}$ | Total credit revolving balance. | ✓        |                     |         |
| B        | $x_{11}$ | Revolving line utilization rate. | ✓        |                     |         |
| B        | $x_{12}$ | Months since the oldest revolving account opened. | ✓        |                     |         |
| B        | $x_{13}$ | Months since the most recent bankcard account opened. | ✓        |                     |         |
| B        | $x_{14}$ | Number of currently active revolving trades. | ✓        |                     |         |
| B        | $x_{15}$ | Number of mortgage accounts. | ✓        | ✓                   | ✓       |
| C        | $x_{16}$ | If income verified by L.C. | ✓        | ✓                   | ✓       |
| C        | $x_{17}$ | Borrower’s last FICO. | ✓        | ✓                   | ✓       |
| D        | $x_{18}$ | Number of inquiries in the past six months. | ✓        |                     |         |
| D        | $x_{19}$ | Number of personal finance inquiries. | ✓        |                     |         |
| D        | $x_{20}$ | Number of credit inquiries in the past 12 months. | ✓        |                     |         |
| D        | $x_{21}$ | Credit history: date a borrower’s earliest reported credit line was opened. | ✓        |                     |         |
| D        | $x_{22}$ | Total bankcard high credit/credit limit. | ✓        |                     |         |

A loan contract contents; $B$ asset strength; $C$ external credit rating; $D$ external credit situation

Fig. 3 Data analysis: OOI of the three models
Fig. 4 Data analysis: features’ “gradient”

(a) Boxplot of gradient for the selected features.

(b) Confidence interval diagram of gradient.

(c) Gradient on samples of the features selected by SAFE-DNN

(d) Gradient of $x_2$ in 27 intervals.

from Fig. 4b. Besides, we show the “gradient” for the SAFE-DNN selected features and each subject in Fig. 4 c, which is plotted similarly to the LIME method (Sun et al. 2020). In this figure, each row represents one feature and each vertical column represents one sample. A darker color suggests that the feature is more important for this sample. Among the selected features, $x_2$ has the greatest influence on the output for almost all samples. We plot the “gradient” of $x_2$ for all 27 intervals in Fig. 4 d. It is observed that the values of “gradient” in most of the intervals are relatively large. $x_2$ represents interest rate on loan, which has been shown to play a vital role in loan default Ntiamoah et al. (2014).

4 Simulation

We conduct simulations to further evaluate performance of the proposed method under settings with different hazard functions, censoring rates, and dimensions of features. We consider two data generating processes, and the correspond-
ing survival times are generated with hazard functions:

\[ s1 : \lambda(t | X) = \exp(h(X, \beta)), \]
\[ s2 : \lambda(t | X) = (0.75h(X, \beta)^3 + 0.75h(X, \beta) + 1.2)t, \]

where

\[ h(X, \beta) = X^\top \beta + 0.8(X^\top \beta)^2 - 0.5(X^\top \beta)^3 + 0.0001 \exp(X^\top \beta). \]

These hazards represent complicated scenarios for which deep learning and other flexible methods may especially be needed. The censoring times are drawn from the uniform distribution \( U(a, b) \). We obtain 25% and 50% censoring rates (cr) by adjusting \( a \) and \( b \). The \( p \)-dimensional variables, \( X \), are generated from the multivariate normal distribution with mean 0 and covariance matrix \( I_{p \times p} \), where \( I_{p \times p} \) is the identity matrix. We consider different dimensions, with \( p = 50, 100, 150 \) for both cases, and the coefficients \( \beta \) are set as:

\[ \beta_1 = \cdots = \beta_{\frac{p}{2}} = 0.2, \quad \beta_{\frac{p}{2}+1} = \cdots = \beta_p = 0.1, \]
\[ \beta_{\frac{p}{2}+1} = \cdots = \beta_{\frac{3p}{2}} = 0.05, \quad \beta_{\frac{3p}{2}+1} = \cdots = \beta_p = 0. \]

The sample size of the training set is 10,000, and that of the testing set is 5,000. For each scenario, 100 replicates are simulated.

When comparing different methods, in addition to the C-index, we also consider the following to evaluate prediction accuracy:

**Brier score:** This can be regarded as the mean squared error between the estimated survival function, \( \hat{S}(t) \), and the actual survival function, \( S(t) \), at time \( t \) (DeVogel et al. 2019), defined as

\[ B(t) = \frac{1}{n} \sum_{i=1}^{n} (S_i(t) - \hat{S}_i(t))^2. \]

A smaller Brier score value implies better performance.

The accuracy of feature extraction is evaluated using Recall, Precision, and their harmonic mean F1 score.

We calculate C-index values at 0.25\( Y_{\text{max}} \), 0.5\( Y_{\text{max}} \), and 0.75\( Y_{\text{max}} \), respectively, where \( Y_{\text{max}} \) is the maximum survival time. The results of C-index are presented in Table 2 of Appendix. Because of the violation of the CPH assumption, Cox-\( L_1 \) has the worst prediction accuracy. Overall, SAFE-DNN has superior prediction performance, with the largest C-index in most of the scenarios. Figures 6 and 7 provide the curves of Brier score for cases s1 and s2, respectively. SAFE-DNN outperforms the other approaches with the smallest Brier scores. Cox-\( L_1 \) performs badly in terms of Brier score with a maximum error of 0.14, while SAFE-DNN has a Brier score of no more than 0.04. Although Nnet-survival-\( L_1 \) enjoys the advantages of being model-free and has better prediction performance than Cox-\( L_1 \), it is inferior to SAFE-DNN in terms of prediction accuracy.

Table 3 presents the results of Recall, Precision, and F1 score. Cox-\( L_1 \) has the worst identification performance because of model misspecification. Except for a small number of cases, SAFE-DNN produces larger Recall and Precision values than Nnet-survival-\( L_1 \). Nonetheless, the result of F1 score indicates that SAFE-DNN outperforms Nnet-survival-\( L_1 \) in feature extraction.

The proposed approach is also computationally affordable, making it possible to have broad practical applications. Overall, it clearly and significantly outperforms the two most relevant competitors.

## 5 Conclusions and future work

We have developed a new approach to predict default risk using penalized deep learning that can be used for loan default analysis with time-to-event data. Rather than predicting whether borrowers will default based on classification methods, SAFE-DNN can predict default risk over time, which provides not only the ult but also forecasts when default will most probably occur. Although several studies have considered risk analysis based on time-to-event data, this study is among the first to conduct penalized deep learning in the context of survival analysis without model structure assumptions. Moreover, we can predict default risk and identify important features simultaneously via a penalized objective function. The minibatch gradient descent algorithm has made it possible to accommodate large-scale data, and numerical studies show the practical superiority of the proposed method. Overall, this study has delivered a practical and useful new way to analyze default risk with massive time-to-event data.

Extensive research on survival analysis has shown its appealing advantages and merits in modeling default risk (Caselli et al. 2021; Xia et al. 2021; Blumenstock et al. 2022). As an alternative, one can also consider analysis based on extreme value theory for default events (Molenberghs and Verbeke 2011). Developing such an analysis and comparing with the proposed is beyond our scope. In this study, we have considered right-censored data. Future work can be extended to other data settings, such as truncated data and interval censored data. Further, when several commercial datasets are available for analysis, it can be important and challenging to recognize and accommodate cross-data heterogeneity. We leave it to future research to study the prediction of default risk in an integrative analysis framework. Although the minibatch gradient descent algorithm makes it
possible to deal with massive data, determining how to train the model with parallel computation to improve computational efficiency deserves additional research. In addition, the structure of the proposed neural network is still not sufficiently complicated. Some studies have considered more advanced neural network structures to improve model performance. In future work, one can examine other deep neural networks or more complex computing algorithms to improve performance.

Acknowledgements The authors are grateful to the Editor, Associate Editor and reviewer for their constructive suggestions and insightful comments, which have led to a great improvement of this paper. This work was supported by the National Natural Science Foundation of China [grant numbers 11701561, 72271237], Foundation from Ministry of Education of China [grant number 20JZD023], and National Institutes of Health [CA204120], and Platform of Public Health & Disease Control and Prevention, Major Innovation & Planning Interdisciplinary Platform for the “Double-First Class” Initiative, Renmin University of China.

Declarations

Conflict of interest There is no conflict of interest.

Appendix

A1. A note on the likelihood function

There are some details about formula (1). For a failure individual \( i \) with failure time \( t \) in the \( j \)-th interval, its likelihood is \( L_i = 1 - \sum_{l=1}^{j-1} p_{il} \). According to the definition of \( \text{surv} \) and \( \text{fail} \), we have \( \text{fail}_i(l) = 0 \) for \( l = 1, \ldots, m, \text{surv}_i(l) = 1 \) for \( l = 1, \ldots, j-1 \) and \( \text{surv}_i(j) = 0 \) for \( l = j, \ldots, m \). Then formula (1) is written as

\[
\log \left( 1 - \sum_{j=1}^{m} \text{surv}_i(j) \cdot p_{ij} + \sum_{j=1}^{m} \text{fail}_i(j) \cdot p_{ij} \right)
\]

\[
= \log \left( 1 - \sum_{l=1}^{j-1} p_{il} \right).
\]

A2. Additional numerical results

See Figures 5, 6, 7 and Tables 2 and 3.

For a censored individual \( i \), if its censored time \( t \) is in \( j \)-th interval, its likelihood is \( L_i = 1 - \sum_{l=1}^{j-1} p_{il} \). According to the definition of \( \text{surv} \) and \( \text{fail} \), we have \( \text{fail}_i(l) = 0 \) for \( l = 1, \ldots, m, \text{surv}_i(l) = 1 \) for \( l = 1, \ldots, j-1 \) and \( \text{surv}_i(j) = 0 \) for \( l = j, \ldots, m \). Then formula (1) is written as

\[
l_i = \log \left( 1 - \sum_{j=1}^{m} \text{surv}_i(j) \cdot p_{ij} + \sum_{j=1}^{m} \text{fail}_i(j) \cdot p_{ij} \right)
\]

\[
= \log \left( 1 - \sum_{l=1}^{j-1} p_{il} \right).
\]

Fig. 5 Kaplan–Meier estimate for loan data
Fig. 6 Simulation results of Brier score for case s1. Censoring rate: 25% for the first column, 50% for the second column. $p = 50, 100,$ and $150$ for the first, second, and third rows, respectively.
Fig. 7  Simulation results of Brier score for case s2. Censoring rate: 25% for the first column, 50% for the second column. $p = 50, 100, \text{ and } 150$ for the first, second, and third rows, respectively.
Table 2  Simulation results: mean(sd) of C-index

| p   | cr (%) | \(0.25Y_{\text{max}}\) | \(0.5Y_{\text{max}}\) | \(0.75Y_{\text{max}}\) |
|-----|--------|----------------|----------------|----------------|
|     |        | Cox- \(L_1\) | SAFE-DNN | Nnet-survival- \(L_1\) | Cox- \(L_1\) | SAFE-DNN | Nnet-survival- \(L_1\) |
| s1  | 50   | 25  | 0.575 | 0.647 | 0.575 | 0.644 | 0.575 | 0.649 | 0.643 |
|     |     |     | (0.005) | (0.006) | (0.005) | (0.006) | (0.005) | (0.007) | (0.026) |
|     | 50   | 25  | 0.585 | 0.642 | 0.585 | 0.635 | 0.585 | 0.632 | 0.611 |
|     |     |     | (0.009) | (0.012) | (0.009) | (0.011) | (0.009) | (0.010) | (0.050) |
|     | 100  | 25  | 0.613 | 0.691 | 0.613 | 0.688 | 0.613 | 0.674 | 0.670 |
|     |     |     | (0.005) | (0.007) | (0.005) | (0.007) | (0.005) | (0.010) | (0.010) |
|     | 50   | 25  | 0.594 | 0.714 | 0.594 | 0.712 | 0.594 | 0.709 | 0.692 |
|     |     |     | (0.010) | (0.009) | (0.010) | (0.009) | (0.010) | (0.008) | (0.012) |
|     | 150  | 25  | 0.578 | 0.715 | 0.578 | 0.710 | 0.578 | 0.703 | 0.659 |
|     |     |     | (0.005) | (0.006) | (0.005) | (0.007) | (0.005) | (0.009) | (0.013) |
| s2  | 50   | 25  | 0.660 | 0.775 | 0.660 | 0.770 | 0.660 | 0.764 | 0.755 |
|     |     |     | (0.009) | (0.003) | (0.009) | (0.004) | (0.009) | (0.008) | (0.009) |
|     | 50   | 25  | 0.643 | 0.770 | 0.643 | 0.762 | 0.643 | 0.755 | 0.744 |
|     |     |     | (0.012) | (0.006) | (0.012) | (0.006) | (0.012) | (0.010) | (0.011) |
|     | 100  | 25  | 0.543 | 0.782 | 0.543 | 0.779 | 0.543 | 0.768 | 0.739 |
|     |     |     | (0.006) | (0.004) | (0.006) | (0.004) | (0.006) | (0.008) | (0.014) |
|     | 150  | 25  | 0.492 | 0.786 | 0.492 | 0.782 | 0.492 | 0.771 | 0.738 |
|     |     |     | (0.008) | (0.005) | (0.008) | (0.006) | (0.008) | (0.008) | (0.018) |
|     | 50   | 25  | 0.517 | 0.782 | 0.517 | 0.780 | 0.517 | 0.770 | 0.741 |
|     |     |     | (0.007) | (0.008) | (0.007) | (0.007) | (0.007) | (0.008) | (0.010) |
|     | 50   | 25  | 0.566 | 0.791 | 0.566 | 0.789 | 0.566 | 0.778 | 0.726 |
|     |     |     | (0.006) | (0.006) | (0.006) | (0.006) | (0.006) | (0.007) | (0.016) |
Table 3  Simulation results: mean(sd) of Recall, Precision, F1 score

| p   | Recall | Precise | F1 score |
|-----|--------|---------|----------|
|     | Cox-L₁ | SAFE-DNN | Nnet-survival- L₁ | Cox-L₁ | SAFE-DNN | Nnet-survival- L₁ | Cox-L₁ | SAFE-DNN | Nnet-survival- L₁ |
| s1  | 0.943  | 0.907   | 0.771  | 0.720   | 0.848  | 0.922  | 0.816  |
|     | (0.037) | (0.069) | (0.037) | (0.057) | (0.075) | (0.079) | (0.008) |
| 100 | 0.983  | 0.862   | 0.895  | 0.821   | 0.668  | 0.942  | 0.898  |
|     | (0.066) | (0.008) | (0.031) | (0.029) | (0.037) | (0.040) | (0.014) |
| s2  | 0.993  | 0.975   | 0.962  | 0.962   | 0.938  | 0.986  | 0.974  |
|     | (0.013) | (0.023) | (0.023) | (0.016) | (0.012) | (0.016) | (0.012) |
| 100 | 0.960  | 0.975   | 0.932  | 0.946   | 0.850  | 0.964  | 0.960  |
|     | (0.034) | (0.020) | (0.049) | (0.043) | (0.034) | (0.024) | (0.020) |
| 150 | 0.822  | 0.975   | 0.932  | 0.939   | 0.828  | 0.954  | 0.938  |
|     | (0.057) | (0.041) | (0.099) | (0.044) | (0.039) | (0.028) | (0.026) |
| 150 | 0.831  | 0.969   | 0.946  | 0.958   | 0.804  | 0.972  | 0.964  |
|     | (0.033) | (0.018) | (0.073) | (0.035) | (0.029) | (0.026) | (0.012) |
| 50  | 0.818  | 0.983   | 0.947  | 0.940   | 0.802  | 0.970  | 0.904  |
|     | (0.040) | (0.040) | (0.036) | (0.036) | (0.025) | (0.030) | (0.018) |

References

Babaev, D., Savchenko, M., Tuzhilin, A., Umerenkov, D.: Et-rnn: applying deep learning to credit loan applications. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2183–2190 (2019). https://doi.org/10.1145/3292500.3330693

Baesens, B., Van Gestel, T., Stepanova, M., Van den Poel, D., Vanthienen, J.: Neural network survival analysis for personal loan data. J. Oper. Res. Soc. 56(9), 1089–1098 (2005). https://doi.org/10.1057/palgrave.jors.2601990

Blumenstock, G., Lessmann, S., Seow, H.V.: Deep learning for survival and competing risk modelling. J. Oper. Res. Soc. 73(1), 26–38 (2022)

Brown, S.F., Branford, A.J., Moran, W.: On the use of artificial neural networks for the analysis of survival data. IEEE Trans. Neural Netw. 8(5), 1071–1077 (1997). https://doi.org/10.1109/72.623209

Calabrese, R., Osmetti, S.A., Zanin, L.: A joint scoring model for peer-to-peer and traditional lending: a bivariate model with copula dependence. J. Royal Stat. Soc. Ser. A (Stat. Soc.) 182(4), 1163–1188 (2019). https://doi.org/10.1111/rssa.12523

Caselli, S., Corbetta, G., Cucinelli, D., Rossolini, M.: A survival analysis of public guaranteed loans: does financial intermediary matter? J. Financ. Stab. 54, 100880 (2021)

Ching, T., Zhu, X., Garmire, L.X.: Cox-nnet: an artificial neural network method for prognosis prediction of high-throughputomics data. PLoS Comput. Biol. 14(4), 1006076 (2018). https://doi.org/10.1371/journal.pcbi.1006076

Costa e Silva, E., Lopes, I.C., Correia, A., Faria, S.: A logistic regression model for consumer default risk. J. Appl. Stat. 47(13–15), 2879–2894 (2020). https://doi.org/10.1080/02664763.2020.1759030

DeVogel, N., Yoganandan, N., Banerjee, A., Pintar, F.: Hierarchical process using brier score metrics for lower leg injury risk curves in vertical impact. BMJ Mil. Health (2019). https://doi.org/10.1136/jramc-2018-001124

Ding, A.A., Tian, S., Yu, Y., Guo, H.: A class of discrete transformation survival models with application to default probability prediction. J. Am. Stat. Assoc. 107(499), 990–1003 (2012). https://doi.org/10.1080/01621459.2012.682806

Dirick, L., Claeskens, G., Baesens, B.: Time to default in credit scoring using survival analysis: a benchmark study. J. Oper. Res. Soc. 68(6), 652–665 (2017). https://doi.org/10.1057/s41274-016-0128-9

Everett, C.R.: Group membership, relationship banking and loan default risk: the case of online social lending. Bank. Finance Rev. (2015). https://doi.org/10.2139/ssrn.1114428

Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: International Conference on Machine Learning, 1050–1059 (2016). PMLR

Gensheimer, M.F., Narasimhan, B.: A scalable discrete-time survival model for neural networks. PeerJ 7, e2657 (2019). https://doi.org/10.7717/peerj.6257

Plzen, Czech Republic
Hackbarth, D., Miao, J., Morellec, E.: Capital structure, credit risk, and macroeconomic conditions. J. Financ. Econ. 82(3), 519–550 (2006)

Hao, L., Kim, J., Kwon, S., Ha, I.D.: Deep learning-based survival analysis for high-dimensional survival data. Mathematics 9(11), 1244 (2021). https://doi.org/10.3390/math9112244

Hassan, M.K., Brodman, J., Rayfield, B., Huda, M.: Modeling credit risk in credit unions using survival analysis. Int. J. Bank Market. 36(3), 482–495 (2018). https://doi.org/10.1108/IJBM-05-2017-0091

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)

Huang, J., Ma, S.: Variable selection in the accelerated failure time hazards model. J. Am. Stat. Assoc. 105(492), 1178–1189 (2010). https://doi.org/10.1080/01621459.2010.1047134

Jaber, J.J., Ismail, N., Ramli, S.N.M.: Credit risk assessment using support vector machine approach to survival analysis with competing risks. In: Thirty-Second AAAI Conference on Artificial Intelligence, 2314–2321 (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16160

Jagtiani, J., Lemieux, C.: The roles of alternative data and machine learning in fintech lending: evidence from the lending club consumer platform. Financ. Manag. 48(4), 1009–1029 (2019). https://doi.org/10.1111/fima.12295

Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science 349(6245), 255–260 (2015). https://doi.org/10.1126/science.aad415

Katzman, J.L., Shalam, 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. 18(1), 24 (2018). https://doi.org/10.1186/s12874-018-0482-1

Khiriart, S., Feyzmahdavian, H.R., Johansson, M.: Mini-batch gradient descent: Faster convergence under data sparsity. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), 2880–2887 (2017). https://doi.org/10.1109/CDC.2017.8264077

Lee, C., Zame, W.R., Yoon, J., van der Schaar, M.: Deephit: a deep learning approach to survival analysis with competing risks. In: Thirty-Second AAAI Conference on Artificial Intelligence, 2314–2321 (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16160

Li, M., Solntanolkotabi, M., Oymak, S.: Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In: International Conference on Artificial Intelligence and Statistics, 4313–4324 (2020). PMLR

Li, Y., Chen, W.: Entropy method of constructing a combined model for improving loan default prediction: A case study in china. J. Oper. Res. Soc. 72(5), 1099–1109 (2021). https://doi.org/10.1080/01605682.2019.1702905

Li, Y., Adams, N., Bellotti, T.: A relabeling approach to handling the class imbalance problem for logistic regression. J. Comput. Gr. Stat. 31(1), 241–253 (2022). https://doi.org/10.1080/10618600.2021.1978470

Lin, D.Y., Wei, L.-J.: The robust inference for the cox proportional hazards model. J. Am. Stat. Assoc. 84(408), 1074–1078 (1989). https://doi.org/10.2307/2290085

Lin, X., Li, X., Zheng, Z.: Evaluating borrower’s default risk in peer-to-peer lending: evidence from a lending platform in China. Appl. Econ. 49(33), 3538–3545 (2017). https://doi.org/10.1080/00036846.2016.1262526

Molenberghs, G., Verbeke, G.: On the Weibull-Gamma frailty model, its infinite moments, and its connection to generalized log-logistic, logistic, cauchy, and extreme-value distributions. J. Stat. Plan. Inference 141(2), 861–868 (2011)

Mollaloo, A., Mao, L., Rashidi, P., Glass, G.E.: A GIS-based artificial neural network model for spatial distribution of tuberculosis across the continental United States. Int. J. Environ. Res. Public Health 16(1), 157 (2019)

Murdock, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Definitions, methods, and applications in interpretable machine learning. Proc. Natl. Acad. Sci. 116(44), 22071–22080 (2019). https://doi.org/10.1073/pnas.1900654116

Nezhad, M.Z., Sadati, N., Yang, K., Zhu, D.: A deep active survival analysis approach for precision treatment recommendations: application of prostate cancer. Exp. Syst. Appl. 115, 16–26 (2019). https://doi.org/10.1016/j.eswa.2018.07.070

Ni, J.-B., Zhu, X., Huang, Y., Zhang, H., Ma, S.: A forward and backward stage-wise algorithm for nonconvex loss functions with adaptive lasso. Comput. Stat. Data Anal. 124, 235–251 (2018). https://doi.org/10.1016/j.csda.2018.03.006

Stepanova, M., Thomas, L.: Survival analysis methods for personal loan data. Oper. Res. 50(2), 277–289 (2002). https://doi.org/10.1287/opre.50.2.277.426

Sun, T., Wei, Y., Chen, W., Ding, Y.: Genome-wide association study-based deep learning for survival prediction. Stat. Med. 39(30), 4605–4620 (2020). https://doi.org/10.1002/sim.8743

Uno, H., Cai, T., Pencina, M.J., D’Agostino, R.B., Wei, L.-J.: On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. Stat. Med. 30(10), 1105–1117 (2011). https://doi.org/10.1002/sim.4154

Wang, J., Zhang, H., Zhu, X., Sun, X., Li, G.: Research on influencing factors of p2p network loan prepayment risk based on cox proportional hazards. Procedia Comput. Sci. 162, 842–848 (2019). https://doi.org/10.1016/j.procs.2019.12.058

Wang, H., Li, G.: Extreme learning machine Cox model for high-dimensional survival analysis, Stat. Med. 38(12), 2139–2156 (2019). https://doi.org/10.1002/sim.8090

Wang, Z., Jiang, C., Ding, Y., Lyu, X., Liu, Y.: A novel behavioral scoring model for estimating probability of default over time in peer-to-peer lending. Electron. Commer. Res. Appl. 27, 74–82 (2018). https://doi.org/10.1016/j.elerap.2017.12.006

Wang, Y., Hong, C., Palmer, N., Di, Q., Schwartz, J., Kohane, I., Cai, T.: A fast divide-and-conquer sparse cox regression. Biostatistics (2018). https://doi.org/10.1093/biostatistics/kxz036

Xia, Y., He, L., Li, Y., Fu, Y., Xu, Y.: A dynamic credit scoring model based on survival gradient boosting decision tree approach. Technol. Econ. Dev. Econ. 27(1), 96–119 (2021)

Xu, B., Su, Z., Celler, J.: Evaluating default risk and loan performance in UK peer-to-peer lending: evidence from funding circle. J. Adv. Comput. Intell. Inform. 25(5), 530–538 (2021). https://doi.org/10.20965/jaciii.2021.p0530

Yuan, M., Tang, C.Y., Hong, Y., Yang, J.: Disentangling and assessing uncertainties in multiperiod corporate default risk predictions. Ann. Oper. Res. 2018:1–19 (2019). https://doi.org/10.1007/s10479-018-2701-4

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.