Biometrics, 2024, 80(1), ujad024
https://doi.org/10.1093/biomtc/ujad024
Biometric Methodology

Penalized deep partially linear cox models with application to CT scans of lung cancer patients

Yuming Sun [1], Jian Kang [2,*], Chinmay Haridas[3], Nicholas Mayne[4], Alexandra Potter[3], Chi-Fu Yang[3], David C. Christiani[5], Yi Li [2,*]

[1]Department of Mathematics, William & Mary, Williamsburg, VA 23185, United States, [2]Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, United States, [3]Division of Thoracic Surgery, Department of Surgery, Massachusetts General Hospital, Boston, MA 02114, United States, [4]Department of Medicine, Duke University, Durham, NC 27710, United States, [5]Department of Environmental Health and Epidemiology, Harvard T.H. Chan School of Public Health, Boston, MA 02115, United States

[*]Corresponding authors: Jian Kang, Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA (jiankang@umich.edu); Yi Li, Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA (yili@umich.edu).

ABSTRACT

Lung cancer is a leading cause of cancer mortality globally, highlighting the importance of understanding its mortality risks to design effective patient-centered therapies. The National Lung Screening Trial (NLST) employed computed tomography texture analysis, which provides objective measurements of texture patterns on CT scans, to quantify the mortality risks of lung cancer patients. Partially linear Cox models have gained popularity for survival analysis by dissecting the hazard function into parametric and nonparametric components, allowing for the effective incorporation of both well-established risk factors (such as age and clinical variables) and emerging risk factors (eg, image features) within a unified framework. However, when the dimension of parametric components exceeds the sample size, the task of model fitting becomes formidable, while nonparametric modeling grapples with the curse of dimensionality. We propose a novel Penalized Deep Partially Linear Cox Model (Penalized DPLC), which incorporates the smoothly clipped absolute deviation (SCAD) penalty to select important texture features and employs a deep neural network to estimate the nonparametric component of the model. We prove the convergence and asymptotic properties of the estimator and compare it to other methods through extensive simulation studies, evaluating its performance in risk prediction and feature selection. The proposed method is applied to the NLST study dataset to uncover the effects of key clinical and imaging risk factors on patients' survival. Our findings provide valuable insights into the relationship between these factors and survival outcomes.

KEYWORDS: CT texture analysis; deep neural network; error rate; feature selection; regularization; selection consistency; survival prediction.

1 INTRODUCTION

Even with the advent of modern medicine, lung cancer mortality remains high, with a 5-year survival rate lower than 20% among advanced patients (Bade and Cruz, 2020). Identifying risk factors relevant to lung cancer survival is essential for designing cancer prevention programs (Barbeau et al., 2006) for prevention and early detection. The National Lung Cancer Screen Trial (NLST) was designed to investigate the use of computed tomography (CT) for lung cancer detection, enrolling more than 53 000 participants from August 2002 through April 2004, with about 26 000 randomly assigned to receive CT (Team, 2011). In addition, clinical information, such as age, gender, smoking history, and cancer stage, was collected for each patient. The study found a 20% decrease in lung cancer mortality for patients screened by CT. It is of interest to examine whether CT confers valuable features to help predict lung cancer survival and design efficient disease management strategies. CT texture analysis provides objective assessments of the texture patterns of the tumor by evaluating the relationship of voxel intensities (Lubner et al., 2017). Identifying reproducible and robust texture features in the presence of other clinical factors affecting patients' outcomes remains a challenge due to the sensitivity of radiomic features to factors such as scanner type, segmentation, and organ motion (Lambin et al., 2017).

Partially linear Cox models have gained popularity as a useful extension of the classic Cox models (Cox, 1972) for survival analysis. This model offers more flexibility in the risk function by separating the hazard function into parametric relative risks for certain covariates and nonparametric relative risks for the others (Huang, 1999). In the NLST analysis, we have chosen to adopt this model by assigning the parametric risks to the texture features and the nonparametric risks to the clinical features such as age, gender, and race. This setup provides a clear interpretation of texture features as in regular Cox models, facilitates the selection of crucial radiomic features, and offers extra flexibility in modeling the effects and potential interactions of the well-known clinical features.

To estimate the nonparametric risk function, researchers have proposed various methods, such as polynomial splines (Huang, 1999). Recently, Zhong et al. (2022) made a breakthrough by us-

Received: March 8, 2023; Revised: September 22, 2023; Accepted: December 6, 2023

ing deep neural networks (DNNs) to estimate the nonparametric risk function in partially linear Cox models and established an optimal minimax rate of convergence for the DNN-based estimator, and showed that DNN approximates a wide range of nonparametric functions with faster convergence. However, the performance of this method remains unknown when dealing with a large number of texture features, which is the case in the NLST study.

In many applications, the neural network has proven to be powerful for approximating complex functions by providing accurate approximations of continuous functions (Leshno et al., 1993). Under some smoothness and structural assumptions, Schmidt-Hieber (2020) showed that DNN estimators may circumvent the curse of dimensionality and achieve the optimal minimax rate of convergence. With limited samples, however, a complex DNN can still lead to overfitting (Srivastava et al., 2014). Early stopping during training (Li et al., 2020), and adding dropout layers (Srivastava et al., 2014), have been proposed to address overfitting, but none has been studied in the survival context.

To fill this gap, we propose a Penalized Deep Partially Linear Cox Model (Penalized DPLC). This framework identifies valuable radiomic features and models the complex relationships between survival outcomes and established clinical characteristics such as age, body mass index (BMI), and pack-years of smoking. The main contributions of our work lie in the proposed penalized estimation, in the context of DNN, to select texture features that influence survival outcomes while avoiding overfitting, combining feature selection, and deep learning in one solution. Second, we demonstrate the asymptotic properties of the estimator by determining its convergence rates and proving selection consistency. Finally, we perform comprehensive simulations to validate the proposed model's theoretical properties and compare it with the other methods in risk prediction and feature selection.

In the following, Section 2 introduces the Penalized DPLC model and Section 3 presents an efficient alternating optimization algorithm. Theoretical results are provided in Section 4, where we prove the convergence rate and variable selection consistency. In Section 5, we conduct simulations to evaluate the performance of the Penalized DPLC and compare it with other state-of-the-art models. We apply the Penalized DPLC to a dataset from the NLST study in Section 6 to identify important texture features related to patient survival.

2 SCAD-PENALIZED DEEP PARTIALLY LINEAR COX MODELS

A partially linear Cox model assumes a hazard function:

$$\lambda(t|\mathbf{x}, \mathbf{z}) = \lambda_0(t) \exp(\boldsymbol{\beta}_0^\top \mathbf{x} + g_0(\mathbf{z})), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{z} \in \mathbb{R}^r$ are two covariate vectors, and $\lambda_0(t)$ is the baseline hazard. This class of models contains the ordinary Cox proportional hazards model as a special case if $g_0(\mathbf{z})$ is a linear function of $\mathbf{z}$. In NLST, $\mathbf{x}$ represents texture features and $\mathbf{z}$ represents known clinical features such as age BMI, gender, race, and cancer stage. The coefficients measuring the impact of texture features are represented by $\boldsymbol{\beta}_0$, while the nonparametric risk function of clinical features is represented by g_0 and is to be approximated by a function in a DNN. We consider a practical setting where p, the dimension of $\mathbf{x}$, can exceed the sample size, which necessitates variable selection. As such, $\boldsymbol{\beta}_0$ is an s_β-sparse vector, that is, $\|\boldsymbol{\beta}_0\|_0 = s_\beta < p$. On the other hand, the important clinical features have a moderate dimension of r, and their complex impacts are to be modeled by a DNN.

As defined in Schmidt-Hieber (2020) and Zhong et al. (2022), a DNN with architecture $(L, \mathbf{p})$ has $L + 1$ layers, including an input layer, $L - 1$ hidden layers and an output layer, and a width vector $\mathbf{p} = (p_1, p_2, \ldots, p_{L+1})$ whose elements are the numbers of neurons in the corresponding layer. In this context, a DNN has 2 or more hidden layers, while shallow networks are those with only 1 hidden layer (Schmidt-Hieber, 2020). In our case, the dimension of the input features, $p_1 = r$, and the dimension of output, $p_{L+1} = 1$. An $(L + 1)$-layered neural network with an architecture $(L, \mathbf{p})$ can be expressed as a composite function, $g : \mathbb{R}^r \to \mathbb{R}^1$, with L folds, that is, $g = g_L \circ g_{L-1} \circ \cdots \circ g_1$, where '$\circ$' is the functional composition, and the lth fold function, $g_l(\cdot) = \sigma_l(\mathbf{W}_l \cdot + \mathbf{b}_l) : \mathbb{R}^{p_l} \to \mathbb{R}^{p_{l+1}}$ with $l = 1, \ldots, L$. Here, $\mathbf{W}_l$ is a $p_{l+1} \times p_l$ weight matrix, $\mathbf{b}_l$ is a p_{l+1}-dimensional bias vector and '$\cdot$' represents an input from layer l. We use Θ to denote the set of parameters for the neural network containing all the weight matrices and bias vectors to be estimated. The function $\sigma_l : \mathbb{R}^{p_{l+1}} \to \mathbb{R}^{p_{l+1}}$ is an activation function, possibly nonlinear, that operates component-wise on a vector.

Various activation functions exist, with rectified linear units (ReLU), that is, $\max(0, \mathbf{a})$, being a commonly used choice. Our primary emphasis lies in neural networks employing ReLU functions across all layers, although these can be readily modified. Moreover, DNNs with complex network structures and a large number of parameters are prone to overfitting. This work concentrates on a class of DNNs with sparsity constraints on the weight and bias matrices (Zhong et al., 2022; Schmidt-Hieber, 2020):

$$\begin{aligned}&\mathcal{G}(L, \mathbf{p}, s, G)\\ &= \{g \in \mathcal{G}(L, \mathbf{p}) : \sum_{l=1}^{L} \|\mathbf{W}_l\|_0 + \|\mathbf{b}_l\|_0 \le s, \|g\|_\infty \le G\}.\end{aligned}$$

Here, $s \in \mathbb{N}_+$ (the set of positive integers), $G > 0$, $\|g\|_\infty = \sup\{|g(z)| : z \in \mathbb{D} \subset \mathbb{R}^r\}$ is the sup-norm of function g, and $\mathbb{D}$ is a bounded set. In implementation, directly specifying or determining s, which controls network sparsity, is not the norm. Instead, a commonly employed technique is a "dropout" procedure within the hidden layers, which randomly removes hidden neurons with a defined probability, referred to as the dropout rate (Srivastava et al., 2014). To determine an appropriate dropout rate, we conduct a grid search as done in our later simulations and data analysis.

With right censoring, we let U_i and C_i denote the survival and censored times for subject i, respectively. We observe $T_i = \min(U_i, C_i)$, and $\Delta_i = 1(U_i \le C_i)$, where $1(\cdot)$ is the indicator function, and assume the observed data $\mathcal{D} = \{(T_i, \Delta_i, \mathbf{x}_i, \mathbf{z}_i), i = 1, \ldots, n\}$ are independently and identically distributed (IID). To estimate g_0 in (1), we suggest using a DNN, denoted as $\mathcal{G}(L, \mathbf{p}, s, \infty)$, which takes $\mathbf{z} \in \mathbb{R}^r$ as input features and produces a scalar output. To achieve variable selection among $\mathbf{x}$, we propose a penalized estimation approach.

To proceed, we define the partial likelihood as

$$\ell(\boldsymbol{\beta}, g) = \frac{1}{n}\sum_{i=1}^{n}\Delta_i\Big[\boldsymbol{\beta}^\top \mathbf{x}_i + g(\mathbf{z}_i) - \log\Big\{\sum_{j\in R_i}\exp\left(\boldsymbol{\beta}^\top \mathbf{x}_j + g(\mathbf{z}_j)\right)\Big\}\Big], \quad (2)$$

where $R_i = \{j\colon T_j \geq T_i\}$, the at-risk set at time T_i, and $g \in \mathcal{G}(L, \mathbf{p}, s, \infty)$. We would estimate $\boldsymbol{\beta}$ and $g(\cdot)$ by maximizing (2), where, to accommodate sparsity, we propose to use the SCAD penalty (Fan and Li, 2001; 2002) defined as

$$p'_\lambda(|\beta|) = \lambda\Big\{\mathrm{I}(|\beta| \leq \lambda) + \frac{(a\lambda - |\beta|)_+}{(a-1)\lambda}\mathrm{I}(|\beta| > \lambda)\Big\}, \quad a > 2,$$

yielding a penalized log partial likelihood, $PL(\boldsymbol{\beta}, g) = \ell(\boldsymbol{\beta}, g) - \sum_{j=1}^{p} p_\lambda(|\beta_j|)$. The SCAD penalty is indeed a quadratic spline function with knots at λ and $a\lambda$, where $\lambda > 0$ is viewed as the tuning parameter controlling the sparsity of $\boldsymbol{\beta}$, and is assumed to converge to 0 as $n \to \infty$, though for simplification we omit its dependence on n.

We estimate $(\boldsymbol{\beta}_0, g_0)$ by maximizing $PL(\boldsymbol{\beta}, g)$, or, equivalently, minimizing the loss function which is defined as the negative penalized log partial likelihood:

$$Q(\boldsymbol{\beta}, g) = q(\boldsymbol{\beta}, g) + \sum_{j=1}^{p} p_\lambda(|\beta_j|), \quad (3)$$

where $q(\boldsymbol{\beta}, g) = -\ell(\boldsymbol{\beta}, g)$. That is, the estimate of $(\boldsymbol{\beta}_0, g_0)$ is obtained via

$$(\widehat{\boldsymbol{\beta}}, \widehat{g}) = \arg\min_{\boldsymbol{\beta}, g\in\mathbb{R}^p\times\mathcal{G}} Q(\boldsymbol{\beta}, g). \quad (4)$$

We present below an optimization algorithm for solving (4) alternately, which uses the adaptive moment estimation (Adam) algorithm to estimate g given an estimate of $\boldsymbol{\beta}$, and, subsequently, uses the resulting estimate $\widehat{g}$ to estimate $\boldsymbol{\beta}$ via coordinate descent.

Step 1. Initialize $\boldsymbol{\beta}$ with $\widehat{\boldsymbol{\beta}}^{(0)}$.

Step 2. Denote by $\widehat{\boldsymbol{\beta}}^{(k-1)}$ the estimate of $\boldsymbol{\beta}$ at the $(k-1)$th iteration. Solve (4) for g, with $\boldsymbol{\beta}$ fixed at $\widehat{\boldsymbol{\beta}}^{(k-1)}$, by using Adam (Algorithm 1), where $\widehat{g}^{(k)}$ denotes the current estimate.

Step 3. With g fixed at $\widehat{g}^{(k)}$, solve (4) for $\boldsymbol{\beta}$ by using the coordinate descent algorithm (Algorithm 2), where $\widehat{\boldsymbol{\beta}}^{(k)}$ denotes the estimate at the current iteration.

We repeat steps 2 and 3 until convergence. In step 2, we employ an adapted Adam algorithm (Algorithm 1), a form of stochastic gradient descent (Kingma and Ba, 2014), to estimate Θ (the weight matrices and bias vectors) in the neural network. The algorithm is adaptive as the update of Θ at each iteration step stems from adaptive estimation of the first and second moments of the stochastic gradients of the empirical loss (Kingma and Ba, 2014). We initialize the biases to be 0 and use *Xavier initialization* to initialize the weights (Glorot and Bengio, 2010). To ensure numerical stability, we add a small $\epsilon_0 > 0$ to the denominator, and the update for each parameter is determined by the adaptive estimates for the first and second moments of the gradients of the empirical loss at each iteration. Algorithm 1 distinguishes from the traditional Adam method in that it updates the parameters in the neural network while fixing $\boldsymbol{\beta}$ at its previous iteration, rather than updating all parameters simultaneously. When implementing Algorithm 1, we do not require convergence with a given update of $\boldsymbol{\beta}$. In our experience, several iterative steps would be sufficient. Also as a large number of iterations may lead to overfitting of DNN, early stopping may prevent overfitting and can produce a consistent network (Ji et al., 2021).

Algorithm 1: Adam in alternating optimization

Input : $r_1, r_2, \gamma, \widehat{\boldsymbol{\beta}}^{(k-1)}, \iota$

1 Initialize $m^{(0)} \leftarrow 0, v^{(0)} \leftarrow 0, t \leftarrow 1, \Theta^{(0)}$

2 **while** $\|\widehat{\Theta}^{(t)} - \widehat{\Theta}^{(t-1)}\|_2 > \iota$ **do**

3 $\quad m^{(t)} \leftarrow r_1 \cdot m^{(t-1)} + (1-r_1)\cdot \nabla_\Theta Q(\widehat{\boldsymbol{\beta}}^{(k-1)}, \widehat{g}^{(t)})$

4 $\quad v^{(t)} \leftarrow r_2 \cdot m^{(t-1)} + (1-r_2)\cdot \{\nabla_\Theta Q(\widehat{\boldsymbol{\beta}}^{(k-1)}, \widehat{g}^{(t)})\}^2$

5 $\quad \widehat{m}^{(t)} \leftarrow m^{(t)}/(1-r_1^t), \widehat{v}^{(t)} \leftarrow v^{(t)}/(1-r_2^t)$

6 $\quad \widehat{\Theta}^{(t)} \leftarrow \widehat{\Theta}^{(t-1)} - \gamma\widehat{m}^{(t)}/(\sqrt{\widehat{v}^{(t)}} + \epsilon_0)$

7 $\quad t \leftarrow t+1$

Output: $\widehat{g}^{(k)} \leftarrow g(\cdot \mid \widehat{\Theta}^{(t)})$

8 Note: the square, division and square root from lines 3 to 6 are operated elementwise.

Step 3 carries out a coordinate descent algorithm. The advantage of coordinate descent is that the parameters, $\boldsymbol{\beta}$, are updated individually, where the closed-form solution for each parameter is available, greatly facilitating the computation (Breheny and Huang, 2011). Specifically, let $\boldsymbol{\xi} = \mathbf{X}\boldsymbol{\beta} \in \mathbb{R}^n$, where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ is the covariate $(\mathbf{x})$ matrix of the n subjects in the data. We denote the gradient and Hessian of the function q with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$ given the current estimate of the neural network, $\widehat{g}^{(k)}$, as $q'(\boldsymbol{\beta}; \widehat{g}^{(k)})$, $q''(\boldsymbol{\beta}; \widehat{g}^{(k)})$, $q'(\boldsymbol{\xi}; \widehat{g}^{(k)})$, and $q''(\boldsymbol{\xi}; \widehat{g}^{(k)})$. To simplify notation, we will omit $\widehat{g}^{(k)}$ in the following. The function $q(\boldsymbol{\beta})$ is approximated using a second order Taylor expansion around $\widehat{\mathbf{b}}^{(t)}$:

$$\begin{aligned} q(\boldsymbol{\beta}) &\approx q(\widehat{\mathbf{b}}^{(t)}) + (\boldsymbol{\beta} - \widehat{\mathbf{b}}^{(t)})^\top q'(\widehat{\mathbf{b}}^{(t)}) \\ &\quad + (\boldsymbol{\beta} - \widehat{\mathbf{b}}^{(t)})^\top q''(\widehat{\mathbf{b}}^{(t)})(\boldsymbol{\beta} - \widehat{\mathbf{b}}^{(t)})/2 \\ &= \frac{1}{2}(y(\widehat{\boldsymbol{\xi}}^{(t)}) - \boldsymbol{\xi})^\top q''(\widehat{\boldsymbol{\xi}}^{(t)})(y(\widehat{\boldsymbol{\xi}}^{(t)}) - \boldsymbol{\xi}) \\ &\quad + C(\widehat{\boldsymbol{\xi}}^{(t)}, \widehat{\mathbf{b}}^{(t)}), \end{aligned}$$

where $y(\widehat{\boldsymbol{\xi}}^{(t)}) = \widehat{\boldsymbol{\xi}}^{(t)} - q''(\widehat{\boldsymbol{\xi}}^{(t)})^{-1}q'(\widehat{\boldsymbol{\xi}}^{(t)})$ and $C(\widehat{\boldsymbol{\xi}}^{(t)}, \widehat{\mathbf{b}}^{(t)})$ does not depend on $\boldsymbol{\beta}$. The equalities hold as $q'(\boldsymbol{\beta}) = \mathbf{X}^\top q'(\boldsymbol{\xi})$ and $q''(\boldsymbol{\beta}) = \mathbf{X}^\top q''(\boldsymbol{\xi})\mathbf{X}$ by the chain rule. Then the loss function (3) at iteration t can be approximated by the penalized weighted sum of squares, $Q(\boldsymbol{\beta}) \approx \frac{1}{2}(y(\widehat{\boldsymbol{\xi}}^{(t)}) - \boldsymbol{\xi})^\top q''(\widehat{\boldsymbol{\xi}}^{(t)})(y(\widehat{\boldsymbol{\xi}}^{(t)}) - \boldsymbol{\xi}) + C(\widehat{\boldsymbol{\xi}}^{(t)}, \widehat{\boldsymbol{\beta}}^{(t)}) + \sum_{j=1}^{p} p_\lambda(|\beta_j|)$. To speed up the algorithm, we may replace $q''(\widehat{\boldsymbol{\xi}}^{(t)})$ by a

diagonal matrix, $\mathbf{W}(\widehat{\boldsymbol{\xi}}^{(t)})$, with the diagonal entries of $q''(\widehat{\boldsymbol{\xi}}^{(t)})$:

$$\mathbf{W}(\widehat{\boldsymbol{\xi}}^{(t)})_{m,m} = q''(\widehat{\boldsymbol{\xi}}^{(t)})_{m,m}$$
$$= \frac{1}{n}\sum_{i\in C_m}\Delta_i\left\{\frac{e^{\widehat{\xi}_m^{(t)}+\widehat{g}_m^{(k)}}\sum_{j\in R_i}e^{\widehat{\xi}_j^{(t)}+\widehat{g}_j^{(k)}} - (e^{\widehat{\xi}_m^{(t)}+\widehat{g}_m^{(k)}})^2}{(\sum_{j\in R_i}e^{\widehat{\xi}_j^{(t)}+\widehat{g}_j^{(k)}})^2}\right\},$$

where $C_m = \{i\colon T_i \le T_m\}$. In this case,

$$y(\widehat{\boldsymbol{\xi}}^{(t)})_m = \widehat{\boldsymbol{\xi}}_m^{(t)}$$
$$+\frac{1}{n\mathbf{W}(\widehat{\boldsymbol{\xi}}^{(t)})_{m,m}}\left\{\Delta_m - \sum_{i\in C_m}\Delta_i\left(\frac{e^{\widehat{\xi}_m^{(t)}+\widehat{g}_m^{(k)}}}{\sum_{j\in R_i}e^{\widehat{\xi}_j^{(t)}+\widehat{g}_j^{(k)}}}\right)\right\}.$$

In the iteration of coordinate descent, the parameters are updated individually; each parameter has a closed-form solution, making the computation manageable. We employ an adaptive rescaling technique (Breheny and Huang, 2011); the following SCAD-thresholding operator returns the univariate solution for the SCAD-penalized optimization:

$$f_{SCAD}(h, v; a, \lambda) = \begin{cases} \frac{S(h,\lambda)}{v}, & \text{if } |h| \le 2\lambda \\ \frac{S(h,a\lambda/(a-1))}{v(1-1/(a-1))}, & \text{if } 2\lambda < |h| \le a\lambda \\ h/v, & \text{if } |h| > a\lambda, \end{cases}$$

where $S(\cdot, \lambda)$ is the soft-thresholding operator with a threshold parameter, $\lambda > 0$ (Donoho and Johnstone, 1994), that is, $S(h, \lambda) = sign(h)(|h| - \lambda)_+$. Here, the sign function $sign(h)$ equals $h/|h|$ if $h \neq 0$, and 0 if $h = 0$; $(h)_+ = \max(h, 0)$. Let $\mathbf{r} = y(\boldsymbol{\xi}) - \boldsymbol{\xi}$ and $v_j = \mathbf{x}_j^\top \mathbf{W}(\boldsymbol{\xi})\mathbf{x}_j$. We define the following input at the tth iteration, that is, $h_j = \mathbf{x}_j^\top \mathbf{W}(\widehat{\boldsymbol{\xi}}^{(t)})\mathbf{r} + v_j\beta_j^{(t)}$. The coordinate descent algorithm is presented in Algorithm 2.

Algorithm 2: Coordinate Descent in alternating optimization

Input : $a, \lambda, \widehat{\mathbf{b}}^{(0)} = \widehat{\boldsymbol{\beta}}^{(k-1)}, \widehat{g}^{(k)}, \iota$

Initialize $t \leftarrow 1, \widehat{\boldsymbol{\xi}}^{(0)} \leftarrow \mathbf{X}\widehat{\mathbf{b}}^{(0)}$, and $\mathbf{r} \leftarrow y(\widehat{\boldsymbol{\xi}}^{(0)}) - \widehat{\boldsymbol{\xi}}^{(0)}$

while $\|\widehat{\mathbf{b}}^{(t)} - \widehat{\mathbf{b}}^{(t-1)}\|_2 > \iota$ **do**

  for $j \leftarrow 1$ **to** p **do**

    $h_j \leftarrow \mathbf{x}_j^\top \mathbf{W}(\widehat{\boldsymbol{\xi}}^{(t-1)})\mathbf{r} + v_j\beta_j^{(t-1)}$

    $\widehat{\beta}_j^{(t)} \leftarrow f_{SCAD}(h_j, v_j; a, \lambda)$

    $\mathbf{r} \leftarrow \mathbf{r} - (\widehat{\beta}_j^{(t)} - \widehat{\beta}_j^{(t-1)})\mathbf{x}_j$

  $\widehat{\boldsymbol{\xi}}^{(t)} \leftarrow \mathbf{X}\widehat{\mathbf{b}}^{(t)}$

  $t \leftarrow t + 1$

Output: $\widehat{\boldsymbol{\beta}}^{(k)} \leftarrow \widehat{\mathbf{b}}^{(t)}$

3 REGULARITY CONDITIONS AND STATISTICAL PROPERTIES

We impose sparsity on $\boldsymbol{\beta}_0 = (\beta_{10}, \ldots, \beta_{p0})^\top = (\boldsymbol{\beta}_{10}^\top, \boldsymbol{\beta}_{20}^\top)^\top$ by, without loss of generality, assuming $\boldsymbol{\beta}_{20} = \mathbf{0}$. We restrict the true nonparametric function g_0 to belong to a composite Hölder class of smooth functions, $\mathcal{H}(q, \alpha, \mathbf{d}, \widetilde{\mathbf{d}}, M)$, where the q composition functions are Hölder smooth functions with parameters $\alpha = (\alpha_1, \ldots, \alpha_q)$ (the orders of smoothness) and M (bound). The concept of the composite Hölder smooth function has been widely used to facilitate the discussion of the theoretical properties of DNN (Schmidt-Hieber, 2020; Zhong et al., 2022). Here, $\mathbf{d} = (d_1, \ldots, d_q)$ and $\widetilde{\mathbf{d}} = (\widetilde{d}_1, \ldots, \widetilde{d}_q)$ are 2 types of dimension parameters; the former is the dimension of input at each 'layer,' while the latter quantifies the intrinsic dimension of the arguments of activation functions at each layer (Zhong et al., 2022), often much smaller than the feature dimension at each layer. We will prove that the convergence rate of DNN depends on $\widetilde{\mathbf{d}}$, instead of $\mathbf{d}$, meaning a faster convergence rate than the other nonparametric estimators. Details can be found in the Supplementary Materials.

Throughout, $\mathbb{E}$ denotes the expectation of random variables; unless otherwise specified, for any function (random or nonrandom) f and a random vector, $\mathbf{v}$, we define $\mathbb{E}\{f(\mathbf{v})\} := \int f(\mathbf{t})f_{\mathbf{v}}(\mathbf{t})d\mathbf{t}$, where $f_{\mathbf{v}}(\cdot)$ is the density function of $\mathbf{v}$. Thus, the expectation is taken with respect to only the arguments of the f function. For a vector $\mathbf{a}$, define $||\mathbf{a}|| = (\mathbf{a}^\top\mathbf{a})^{1/2}$, and for a function g, define $\|g\|_{L^2}^2 = \mathbb{E}\{g^2(\mathbf{z})\}$. We denote $\widetilde{\alpha}_i = \alpha_i\prod_{k=i+1}^q(\alpha_k \wedge 1)$ and $\gamma_n = \max_{i=1,\ldots,q} n^{-\widetilde{\alpha}_i/(2\widetilde{\alpha}_i+\widetilde{d}_i)}$, and assume the following.

1. Considering a class of s-sparse DNNs or $\mathcal{G}(L, \mathbf{p}, s, G)$, we assume $L = O(\log n)$, $s = O(n\gamma_n^2 \log n)$ and $n\gamma_n^2 < \min_{l=1,\ldots,L} p_l \le \max_{l=1,\ldots,L} p_l < n$.
2. With slightly overuse of notation, denote by $\mathbf{x}$ and $\mathbf{z}$ the random vectors underlying the observed IID copies of $\mathbf{x}_i$ and $\mathbf{z}_i$, respectively. Assume $(\mathbf{x}^\top, \mathbf{z}^\top)^\top$ take values in a bounded subset, $\mathbb{D}$, of $\mathbb{R}^{p+r}$ with a joint probability density function bounded away from zero, and $\boldsymbol{\beta}_0$ lies in a compact set, that is., $\boldsymbol{\beta}_0 \in \{\boldsymbol{\beta} \in \mathbb{R}^p : \|\boldsymbol{\beta}\| \le B\}$.
3. Assume that the nonparametric function g_0 belongs to a mean 0 composite Hölder smooth class, that is, $g_0 \in \mathcal{H}_0 := \{g \in \mathcal{H}(q, \alpha, \mathbf{d}, \widetilde{\mathbf{d}}, M) : \mathbb{E}\{g(\mathbf{z})\} = 0\}$ and the matrix $\mathbb{E}\{\mathbf{x} - \mathbb{E}(\mathbf{x}|\mathbf{z})\}^{\otimes 2}$ is nonsingular, where $\mathbf{a}^{\otimes 2} = \mathbf{a}\mathbf{a}^\top$ for a column vector $\mathbf{a}$.
4. Let $\tau < \infty$ be the maximal followup time. We assume that there exits a $\delta > 0$ such that $P(\Delta = 1|\mathbf{x}, \mathbf{z}) > \delta$ and $P(U > \tau|\mathbf{x}, \mathbf{z}) > \delta$ almost surely.

Condition 1 restricts the architecture of neural networks, balancing the network's flexibility with the estimation accuracy (Zhong et al., 2022). Condition 2 is commonly assumed for semiparametric partially linear models (Horowitz, 2009). The Hölder smoothness in Condition 3 ensures that the function can be approximated by a DNN, while the zero expectation assumption yields the identifiability of the DPLC (Zhong et al., 2022). In Condition 4, $P(\Delta = 1|\mathbf{x}, \mathbf{z}) > \delta$ specifies that there is nonzero probability of observing an event, and $P(U > \tau|\mathbf{x}, \mathbf{z}) > \delta$ ensures that there is nonzero probability that some subjects are still alive at the end of the study, both of which guarantee that the partially linear Cox model can be estimated using the observed data.

With $a_n = \max\{p'_\lambda(|\beta_{j0}| : \beta_{j0} \neq 0)\}$ and $b_n = \max\{p''_\lambda(|\beta_{j0}| : \beta_{j0} \neq 0)\}$, the following theorem establishes the existence and the convergence rates of $\widehat{\boldsymbol{\beta}}$ and $\widehat{g}$.

Theorem 1 *Under Conditions 1–4, and if $b_n \to 0$ (with properly chosen λ), then there exists a local maximizer $(\widehat{\boldsymbol{\beta}}, \widehat{g})$ of $PL(\boldsymbol{\beta}, g)$ satisfying $\mathbb{E}\{\widehat{g}(\mathbf{z})\} = 0$, such that*

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_p(\gamma_n \log^2 n + a_n), \quad \|\widehat{g} - g_0\|_{L^2} = O_p(\gamma_n \log^2 n + a_n).$$

Remark 1 *The theorem shows that the rate of convergence does not depend on the number of input features, but rather on the intrinsic dimension and smoothness of the function g_0, unlike other nonparametric estimators whose convergence rate also depends on the feature dimension. As a result, the DNN estimator may have an advantage when the intrinsic dimension of the true function is low.*

We now show that the estimator $\widehat{\boldsymbol{\beta}} = (\widehat{\boldsymbol{\beta}}_1^\top, \widehat{\boldsymbol{\beta}}_2^\top)^\top$ for $(\boldsymbol{\beta}_{10}^\top, \boldsymbol{\beta}_{20}^\top = \mathbf{0}^\top)^\top$ possesses a selection consistency property, that is, $\widehat{\boldsymbol{\beta}}_2 = \mathbf{0}$ with probability going to 1.

Theorem 2 *Assume that $\liminf_{n\to\infty} \liminf_{u\to 0^+} p'_\lambda(u)/\lambda > 0$, and λ is chosen such that $a_n = O(\gamma_n log^2 n)$, and $\lambda min\,(n^{1/2}, \{\gamma_n log^2(n)\}^{-1}) \longrightarrow \infty$, and the conditions of Theorem 1 hold. Then with probability tending to 1, the estimator $\widehat{\boldsymbol{\beta}}$ in Theorem 1 must satisfy $\widehat{\boldsymbol{\beta}}_2 = \mathbf{0}$.*

Remark 2 *Both theorems apply to a broad range of penalty functions. In particular, as shown in Fan and Li (2001), the SCAD penalty function satisfies $\liminf_{n\to\infty} \liminf_{u\to 0^+} p'_\lambda(u)/\lambda > 0$, and as $\lambda \to 0+$, $a_n = 0$ when n is sufficiently large. Consequently, if λ converges to 0 at an appropriate rate, the SCAD function guarantees both the convergence rates (Theorem 1) and variable selection consistency (Theorem 2).*

4 SIMULATIONS

We conducted simulations to assess the finite sample performance of our proposed estimator by comparing it with the SCAD-penalized Cox Model (Fan and Li, 2002), SCAD-penalized Partially Linear Cox Model using polynomial splines (Hu and Lian, 2013), Cox Boosting (Binder et al., 2009), Random Forest (Ishwaran et al., 2008) and Deep Survival Model (Katzman et al., 2018). For $i = 1, \ldots, n$, we generated $(\mathbf{x}_i, \mathbf{z}_i)$ from a multivariate Gaussian distribution, $\mathcal{N}_{p+r}\left\{\mathbf{0}, \begin{pmatrix} 1 & 0.2 & \ldots & 0.2 \\ \vdots & \vdots & \ddots & \vdots \\ 0.2 & 1 & \ldots & 1 \end{pmatrix}\right\}$, and then generated the true survival time U_i from an exponential distribution with a hazard $\mu \exp(\boldsymbol{\beta}_0^T \mathbf{x}_i + g_0(\mathbf{z}_i))$, where μ was tuned to adjust for censoring rate and $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ was a sparse vector simulated from the uniform distribution. The number of nonzero elements in $\boldsymbol{\beta}_0$ was s_β, chosen to be much less than the dimension of $\boldsymbol{\beta}_0$. The censored time C_i was simulated from $\mathcal{U}[0, \mathcal{C}]$, where $\mathcal{C}$ was chosen so that the censoring rate in the simulated data is around 30%.

We simulated data sets with varying sample sizes and feature sizes. Specifically, we fixed the clinical feature size, r, at 8 and the number of nonzero radiomic features, s_β, to be 10, while varying the training sample size, n, to be 500 or 1500 and radiomic feature size, p, to be 600 or 1200. We assessed the performance of the model under these four scenarios with different numbers of training samples and feature sizes. For each simulation setup or configuration, a total of 500 independently simulated datasets were generated.

We set $g_0 : \mathbb{R}^8 \to \mathbb{R}$ to be a linear or nonlinear function, respectively. That is, $g_0(\mathbf{z}) = \boldsymbol{\alpha}_0^\top \mathbf{z}$ with $\boldsymbol{\alpha}_0 \in \mathbb{R}^8$ generated from $\mathcal{U}(-2, 2)$ or $0.68\exp(z_1) - 0.45\log\{(z_2 - z_3)^2\} + 0.32\sin(z_4 z_5) - 0.45(z_6 - z_7 + z_8)^2 - 0.32$. We tuned parameters for each method on each simulated dataset. Specifically, to identify the neural network structure, we tuned the number of hidden layers and the number of neurons in the hidden layers over a grid of values, that is, 1–4 for the number of hidden layers and 2–8 for the number of neurons in the hidden layers, and tuned the dropout rate and the learning rate from 0.3 to 0.5 and from 0.005 to 0.02, respectively. For the SCAD penalty, we set $a = 3.7$ as suggested by Fan and Li (2001) and used grid search over $[0.05,5]$ to find the best λ based on the Bayesian Information Criterion (BIC): $-2n\ell(\widehat{\boldsymbol{\beta}}, \widehat{g}) + \log n \cdot \widehat{s}_\beta$, where $\widehat{s}_\beta$ is the number of nonzero coefficient estimates; for illustration, Figure S1(a and b) in the Supplementary Material display the selection of λ for SCAD-Penalized DPLC on 10 simulated datasets with $(n, p) = (500, 1, 200)$ and the solution path for $\widehat{\boldsymbol{\beta}}$ with one randomly selected dataset. We tuned for Cox Boosting by determining the penalty value that yielded an optimal count of boosting steps (with a maximum of 200). For Random Forest, we tuned the terminal node size from 1 to 150.

To visually evaluate the accuracy of the DNN estimator in approximating g_0 when it is nonlinear, Figure 1 displays contour plots of the true function and the average DNN estimates based on 500 simulated datasets with n, p varying from 500 to 1500 and from 600 to 1200, respectively. When creating these plots, we fixed the values of the last 6 arguments of the function at their population means and varied the first 2 arguments. The results indicate that the DNN estimates provided a good approximation of the true function, with increasing accuracy observed as n increased for a fixed value of p.

Figure 2 compared the Penalized DPLC's prediction performance with the competing methods using the C-Index as the criterion. When g_0 is linear or the ordinary Cox model holds, three Cox model-based methods, Cox-SCAD, SCAD splines, and Boosting, excelled with a highest median C-Index of approximately 0.92 across various combinations of n and p. Our penalized DPLC yielded a competitive median C-Index, ranging from 0.83 to 0.87 at $n = 500$ and improved to 0.89 at $n = 1,500$; importantly, it outperformed two nonparametric methods: Random Forest (median C-Index values: 0.77–0.80) and Deep Survival Model (0.70–0.89). When g_0 is nonlinear, our Penalized DPLC model clearly outperformed the others across various n and p. The highest median C-Index of 0.868 [Interquartile range (IQR): 0.011] was achieved with $(n, p) = (1, 500, 600)$. As the feature size increased, the prediction performance decreased slightly, for example, the median C-Index for Penal-

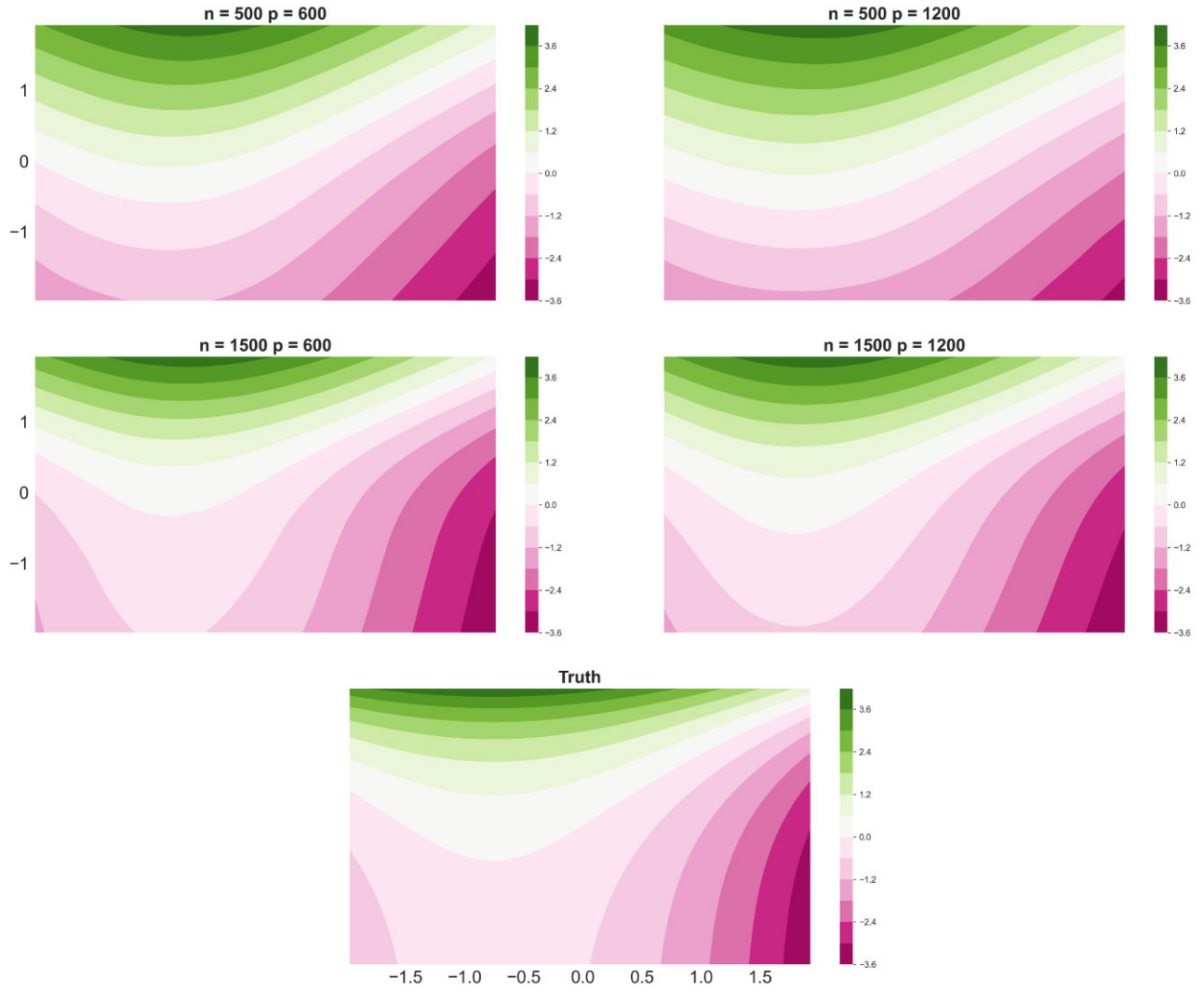
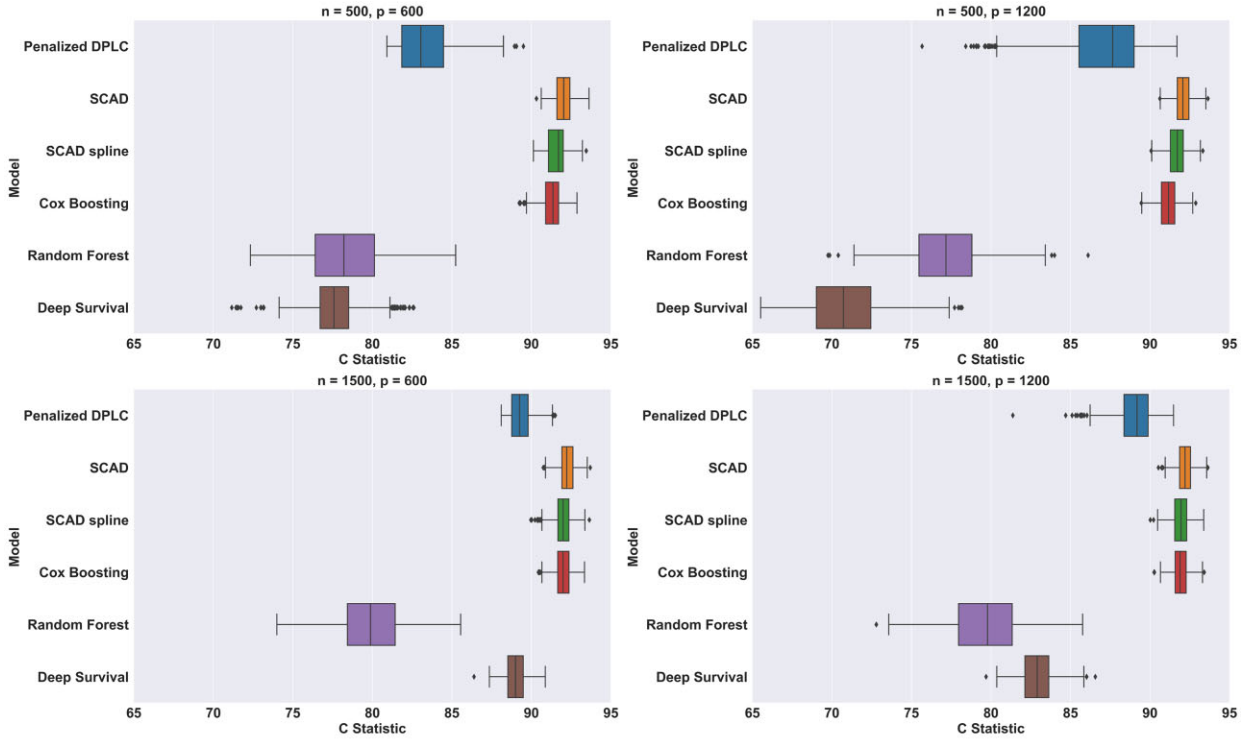


FIGURE 1 The Average Estimates of the Nonlinear Function using 500 Simulated Datasets with Varying n, p**.** The plots are made by varying the first 2 arguments fixing the other 6 arguments.

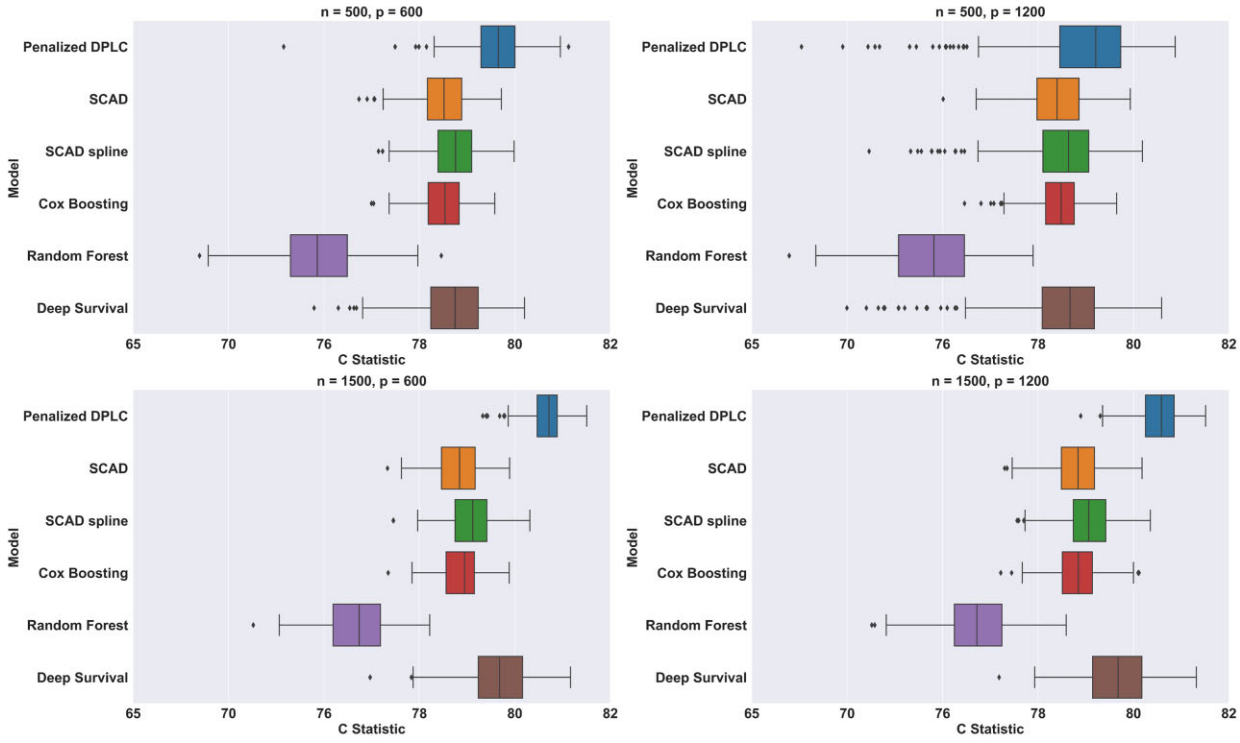
ized DPLC decreased from 0.841 (IQR: 0.018) to 0.830 (IQR: 0.032) when the feature size increased from 600 to 1200 with 500 samples. The prediction performance improved with more samples; the median C-Index for Penalized DPLC rose to 0.865 (IQR: 0.015) when the sample size increased to 1500, compared to 500 samples with 1200 features.

To evaluate the selection performance, we reported the number of selected features, false positive number (FPN), false positive rate (FPR), false negative number (FNN), and false negative rate (FNR). Let $\mathcal{S}$ and $\widehat{\mathcal{S}}$ represent the actual and estimated (ie, the selected features) support of $\boldsymbol{\beta}$, and $Card(\cdot)$ the cardinality of a set. Then FPN $= Card(\widehat{\mathcal{S}}\backslash\mathcal{S})$, FPR $=$ FPN$/\{p - Card(\mathcal{S})\}$, FNN $= Card(\mathcal{S}\backslash\widehat{\mathcal{S}})$, and FNR $=$ FNN$/Card(\mathcal{S})$. When $g_0(\mathbf{z})$ is linear on $\mathbf{z}$, in which case the model assumptions were satisfied for the SCAD-penalized Cox model with and without polynomial splines, they outperformed the other competing methods, including the Penalized DPLC (Table 1). However, the performance of the penalized DPLC was comparable to them. For example, the 2 penalized Cox models reported an FPN of less than 1, while the penalized DPLC reported only 0.6 more FPNs on average than them. In addition, the average FNN, that is, the missed 'active' features, of the Penalized DPLC was only 0.74–2.31 (across various considered scenarios) higher than the penalized Cox models. On the other hand, the performance of the Penalized DPLC was clearly better than Cox Boosting and Random Forest. Cox Boosting tended to select more features; when $(n, p) = (1\,500, 1\,200)$, Cox Boosting reported an FPN of 33.64 (SE: 0.60), whereas the FPN for the Penalized DPLC was 1.40 (SE: 0.10). For Random Forest, the average FNN varied from 4.12 to 6.17, compared to 1.33–3.49 for the Penalized DPLC. The average FPN for Random Forest varied from 3.75 to 6.11, while it was 0.31–1.69 for the Penalized DPLC.

When $g_0(\mathbf{z})$ is nonlinear, the Penalized DPLC outperformed almost all of the other methods (except for Cox Boosting) in FNN. Cox Boosting had an FNN of 1.69 (SE: 0.04), while the



(a) Linear Case



(b) Nonlinear Case

FIGURE 2 Prediction Performance Based on 500 Simulated Datasets.

TABLE 1 Selection performance of different algorithms using 500 simulated datasets.

	Method	Selected Features[1]	FPN[2]	FPR (%)[3]	FNN[4]	FNR(%)[5]
Linear Case						
$(n, p) = (500, 600)$	Penalized DPLC	6.90 (0.07)	0.39 (0.04)	0.07 (0.01)	3.49 (0.04)	34.90 (0.39)
	SCAD	9.47 (0.19)	0.48 (0.09)	0.08 (0.01)	1.32 (0.11)	13.20 (1.10)
	SCAD spline	9.51 (0.21)	0.69 (0.15)	0.12 (0.03)	1.18 (0.11)	11.80 (1.10)
	Cox Boosting	48.10 (1.02)	38.75 (1.01)	6.57 (0.17)	0.65 (0.08)	6.50 (0.82)
	Random Forest	9.51 (0.21)	4.90 (0.26)	0.83 (0.04)	5.39 (0.13)	53.90 (1.29)
$(n, p) = (500, 1\,200)$	Penalized DPLC	9.90 (0.08)	1.69 (0.09)	0.14 (0.01)	1.79 (0.05)	17.86 (0.49)
	SCAD	9.92 (0.08)	0.85 (0.06)	0.07 (0.01)	0.93 (0.04)	9.32 (0.40)
	SCAD spline	9.94 (0.08)	0.89 (0.07)	0.07 (0.01)	0.95 (0.04)	9.48 (0.40)
	Cox Boosting	49.22 (0.54)	39.91 (0.54)	3.35 (0.05)	0.69 (0.03)	6.88 (0.34)
	Random Forest	9.94 (0.08)	6.11 (0.10)	0.51 (0.01)	6.17 (0.06)	61.74 (0.56)
$(n,p) = (1\,500, 600)$	Penalized DPLC	8.99 (0.08)	0.31 (0.07)	0.05 (0.01)	1.33 (0.04)	13.26 (0.38)
	SCAD	9.34 (0.03)	0.01 (0.00)	0.00 (0.00)	0.67 (0.03)	6.68 (0.33)
	SCAD spline	9.63 (0.04)	0.22 (0.03)	0.04 (0.00)	0.59 (0.03)	5.88 (0.33)
	Cox Boosting	42.68 (0.53)	33.00 (0.53)	5.59 (0.09)	0.33 (0.02)	3.26 (0.25)
	Random Forest	9.63 (0.04)	3.75 (0.07)	0.64 (0.01)	4.12 (0.06)	41.20 (0.63)
$(n, p) = (1\,500, 1\,200)$	Penalized DPLC	9.87 (0.12)	1.40 (0.10)	0.12 (0.01)	1.53 (0.05)	15.32 (0.55)
	SCAD	9.23 (0.04)	0.04 (0.01)	0.00 (0.00)	0.81 (0.04)	8.08 (0.39)
	SCAD spline	9.65 (0.05)	0.37 (0.04)	0.03 (0.00)	0.72 (0.04)	7.16 (0.37)
	Cox Boosting	43.19 (0.61)	33.64 (0.60)	2.83 (0.05)	0.45 (0.03)	4.46 (0.28)
	Random Forest	9.65 (0.05)	4.36 (0.08)	0.37 (0.01)	4.71 (0.06)	47.08 (0.65)
Nonlinear Case						
$(n, p) = (500, 600)$	Penalized DPLC	11.04 (0.11)	2.52 (0.10)	0.43 (0.02)	1.48 (0.03)	14.76 (0.29)
	SCAD	12.68 (0.19)	4.66 (0.17)	0.79 (0.03)	1.98 (0.05)	19.78 (0.47)
	SCAD spline	12.32 (0.18)	4.13 (0.16)	0.70 (0.03)	1.81 (0.05)	18.08 (0.46)
	Cox Boosting	34.73 (0.49)	26.02 (0.48)	4.41 (0.08)	1.29 (0.04)	12.90 (0.38)
	Random Forest	12.32 (0.18)	7.27 (0.18)	1.23 (0.03)	4.95 (0.06)	49.50 (0.59)
$(n, p) = (500, 1{,}200)$	Penalized DPLC	10.74 (0.13)	2.88 (0.12)	0.24 (0.01)	2.14 (0.04)	21.38 (0.37)
	SCAD	12.89 (0.21)	5.25 (0.19)	0.44 (0.02)	2.36 (0.05)	23.64 (0.51)
	SCAD spline	17.87 (0.97)	10.02 (0.97)	0.84 (0.08)	2.15 (0.05)	21.48 (0.48)
	Cox Boosting	34.90 (0.49)	26.59 (0.47)	2.23 (0.04)	1.69 (0.04)	16.94 (0.44)
	Random Forest	17.87 (0.97)	13.16 (0.96)	1.11 (0.08)	5.28 (0.05)	52.84 (0.51)
$(n, p) = (1\,500, 600)$	Penalized DPLC	9.14 (0.04)	0.26 (0.02)	0.04 (0.00)	1.12 (0.03)	11.16 (0.33)
	SCAD	8.76 (0.05)	0.47 (0.03)	0.08 (0.01)	1.71 (0.05)	17.06 (0.45)
	SCAD spline	10.49 (0.11)	1.82 (0.10)	0.31 (0.02)	1.32 (0.04)	13.24 (0.40)
	Cox Boosting	33.12 (0.52)	24.02 (0.51)	4.07 (0.09)	0.90 (0.03)	9.00 (0.34)
	Random Forest	10.49 (0.11)	4.08 (0.13)	0.69 (0.02)	3.58 (0.06)	35.84 (0.59)
$(n, p) = (1\,500, 1\,200)$	Penalized DPLC	9.20 (0.08)	0.94 (0.06)	0.08 (0.00)	1.74 (0.05)	17.40 (0.46)
	SCAD	9.04 (0.09)	1.16 (0.07)	0.10 (0.01)	2.12 (0.05)	21.20 (0.50)
	SCAD spline	10.35 (0.13)	2.17 (0.11)	0.18 (0.01)	1.83 (0.05)	18.26 (0.47)
	Cox Boosting	33.20 (0.55)	24.56 (0.54)	2.06 (0.05)	1.36 (0.04)	13.58 (0.40)
	Random Forest	10.35 (0.13)	4.60 (0.13)	0.39 (0.01)	4.26 (0.06)	42.56 (0.61)

Notes: [1] The number of true 'active' features is set to be ten.
[2] False positive number (FPN) is the number of features that are 'inactive' but selected by the model as 'active' features.
[3] False positive rate (FPR) is the FPN divided by the true number of 'inactive' features and reported as a percentage ($\times$ 100).
[4] False negative number (FNN) is the number of features that are 'active' but selected by the model as 'inactive' features.
[5] False negative number (FNR) is the FNN divided by the true number of 'active' features and reported as a percentage ($\times$ 100). * Reported numbers are means and SEs .

Penalized DPLC reported a comparable FNN of 2.14 (SE: 0.04) with 500 samples and 1200 features. However, Cox Boosting had a much higher FPN of 26.59 (SE: 0.47) compared to Penalized DPLC's 2.88 (SE: 0.12). The average number of falsely selected features using Penalized DPLC was 0.26–2.88, compared to 0.47–5.25 for the penalized Cox model. The selection performance of Penalized DPLC improved with more samples and fewer features, achieving the best performance when $(n, p) = (1500, 600)$ with an FPR of 0.04% and an FNR of 11.16%.

5 APPLICATION

We applied the Penalized DPLC to analyze a dataset from NLST, investigating what and how CT features were related to the mortality of lung cancer patients. The dataset includes a total of 368 subjects from NLST who were diagnosed with lung cancer and screened with CT (Table 2). Out of them, 96 patients died during follow-up. The median age was 63.5 years old (IQR: 59.0, 68.0), with 55% being male and over 90% being white. Most patients were in the early cancer stage, and hypertension was the

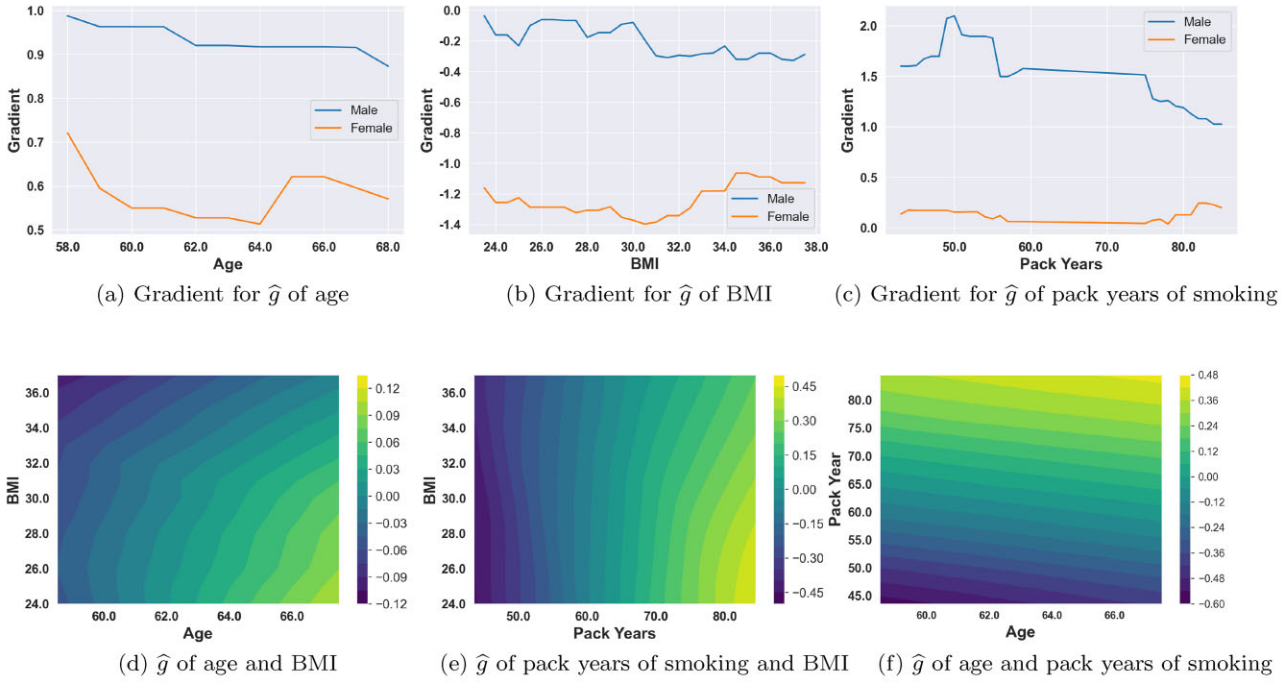
TABLE 2 Clinical characteristics of patients from the national lung cancer screen trial.

Characteristic	Overall, N = 368[1]	Alive, N = 272[1]	Dead, N = 96[1]
Median follow-up time (days)	2072 (1962, 2151)		
Age (years)	63.5 (59.0, 68.0)	63.0 (59.0, 67.0)	66.0 (60.0, 70.0)
BMI	26.3 (24.3, 29.2)	26.3 (24.3, 29.2)	26.1 (24.1, 29.2)
Gender			
Male	201 (55%)	137 (50%)	64 (67%)
Female	167 (45%)	135 (50%)	32 (33%)
Race			
White	339 (92%)	251 (92%)	88 (92%)
Black	14 (3.8%)	11 (4.0%)	3 (3.1%)
Asian	8 (2.2%)	6 (2.2%)	2 (2.1%)
Other	6 (1.6%)	3 (1.1%)	3 (3.1%)
Unknow	1 (0.3%)	1 (0.4%)	0 (0%)
Cigarette smoking status			
Former	171 (46%)	135 (50%)	36 (38%)
Current	197 (54%)	137 (50%)	60 (62%)
Pack years of smoking	58 (46, 80)	57 (45, 80)	60 (49, 84)
Histology			
Adenocarcinoma	185 (50%)	137 (50%)	48 (50%)
Squamous cell carcinoma	73 (20%)	50 (18%)	23 (24%)
Large cell C=carcinoma	16 (4.3%)	9 (3.3%)	7 (7.3%)
Adenosquamous carcinoma	8 (2.2%)	3 (1.1%)	5 (5.2%)
Neuroendocrine/Carcinoid tumors	1 (0.3%)	1 (0.4%)	0 (0%)
Bronchioloalveolar carcinoma	70 (19%)	59 (22%)	11 (11%)
NSCLC NOS	15 (4.1%)	13 (4.8%)	2 (2.1%)
Pathologic stage			
IA	230 (62%)	188 (69%)	42 (44%)
IB	49 (13%)	36 (13%)	13 (14%)
IIA	11 (3.0%)	8 (2.9%)	3 (3.1%)
IIB	39 (11%)	26 (9.6%)	13 (14%)
IIIA	33 (9.0%)	13 (4.8%)	20 (21%)
IIIB	3 (0.8%)	1 (0.4%)	2 (2.1%)
IV	3 (0.8%)	0 (0%)	3 (3.1%)
Radiotherapy	27 (7.3%)	9 (3.3%)	18 (19%)
Chemotherapy	83 (23%)	49 (18%)	34 (35%)
Surgery type			
Wedge/MultiplewWedge resection	45 (12%)	30 (11%)	15 (16%)
Segmentectomy	14 (3.8%)	8 (2.9%)	6 (6.2%)
Lobectomy	287 (78%)	222 (82%)	65 (68%)
Bilobectomy	15 (4.1%)	9 (3.3%)	6 (6.2%)
Pneumonectomy	7 (1.9%)	3 (1.1%)	4 (4.2%)
Asthma	27 (7.3%)	18 (6.6%)	9 (9.4%)
Bronchitis	35 (9.5%)	23 (8.5%)	12 (12%)
COPD	39 (11%)	24 (8.8%)	15 (16%)
Diabetes	33 (9.0%)	20 (7.4%)	13 (14%)
Emphysema	48 (13%)	32 (12%)	16 (17%)
Heart disease	52 (14%)	35 (13%)	17 (18%)
Hypertension	134 (36%)	98 (36%)	36 (38%)
Prior pneumonia	77 (21%)	53 (19%)	24 (25%)
Obstructive lung disease	88 (24%)	58 (21%)	30 (31%)

Note. [1] Median (IQR); *n* (%)

most prevalent comorbidity (36%), followed by obstructive lung disease (24%) and prior pneumonia (21%).

To extract features from CT scans, we followed the image processing pipeline as outlined in Figure S2 in the Supplementary Material. We first removed noise from the images through gray-scale normalization and adaptive histogram equalization. We then normalized the voxel intensity of each image to a standard range of 0 (black)–255 (white) units and improved the contrast with adaptive histogram equalization. We next identified the regions of interests (ROIs) and segmented the tumor regions based on their location and size. We used *pyradiomics* to extract texture features from the ROIs, including first-order features, shape-based features, and higher-order features (Amadasun and King, 1989). We applied image filtration using the Laplacian of Gaussian filter and a 3D LBP-based filter; the Laplacian of Gaussian filter highlights areas of gray level change (Kong et al., 2013), and the 3D LBP-based filter computes local binary patterns in 3D using spherical



(a) Gradient for $\widehat{g}$ of age (b) Gradient for $\widehat{g}$ of BMI (c) Gradient for $\widehat{g}$ of pack years of smoking

(d) $\widehat{g}$ of age and BMI (e) $\widehat{g}$ of pack years of smoking and BMI (f) $\widehat{g}$ of age and pack years of smoking

FIGURE 3 Estimated Nonlinear Function and Gradients using NLST: The gradients for $\widehat{g}$ of age, BMI, and pack years smoking history stratified by gender are plotted in (a), (b), and (c). $\widehat{g}$ of age, BMI, and pack years of smoking is plotted in (d) and (e). The other variables are fixed at their sample means (for continuous variables) or modes (for categorical variables).

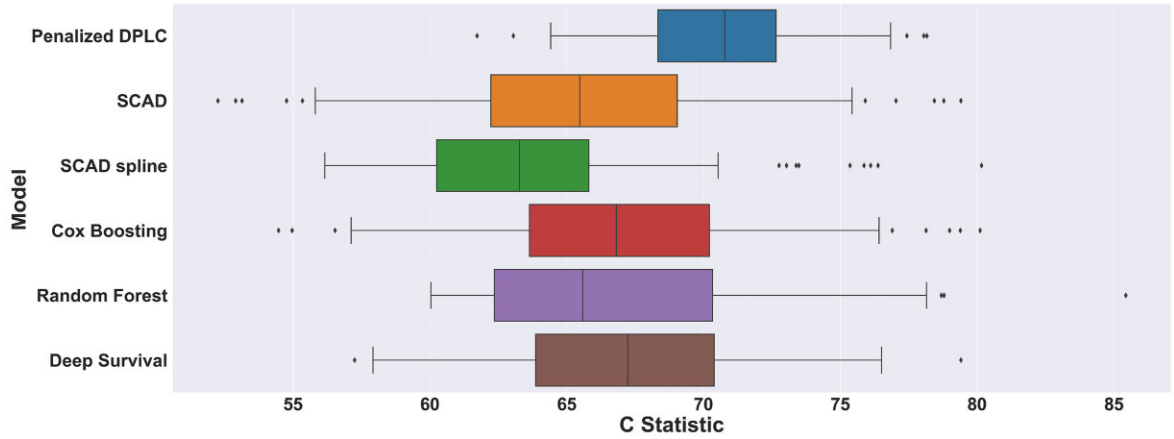


FIGURE 4 Prediction Performance of 100 Experiments using Data from the National Lung Cancer Screen Trial: during each experiment, 80% data iare randomly selected as training data, and 20% data are selected as testing data. The censoring rate in the testing data and training data are controlled to be the same as that in the entire population.

harmonics (Banerjee et al., 2012). A total of 320 image features were extracted.

To compare the prediction and selection accuracy of the Penalized DPLC with other competing methods, we conducted 100 experiments. In each experiment, we tuned the number of hidden layers and the number of neurons in each hidden layer over the grids of [1, 2, 3, 4] and [2, 4, 8, 16], respectively, when constructing the DNN, and randomly divided the data into 80% for training and the remaining 20% for testing. To ensure that the censoring rate in the training and testing data remained the same as in the entire population, we split the data by stratifying the vital status of the patients. Similar to the simulation study, we tuned the number of hidden layers and the number of neurons in each hidden layers over the grid of [1, 2, 3, 4] and [2, 4, 8, 16], respectively.

Figure 3a–f illustrate the estimated effects of age, BMI, and pack years of smoking while holding other variables constant at their mean (for continuous variables) or mode (for categorical variables), as derived from the estimated $\widehat{g}$ function. These contour plots clearly reveal the nonlinear relationships between age, BMI, and pack years of smoking and survival. The gradients of $\widehat{g}$ for age, BMI, and pack years, stratified by gender, are presented in Figure 3 a–c, reflecting the local change in the log haz-

ard for small changes in the corresponding variables. Figure 3a and c exhibit positive gradients for age and pack years, indicating that mortality increases with increasing age and pack years, consistent with the literature (Tindle et al., 2018). In contrast, Figure 3b shows that BMI has a protective effect on patient survival, in agreement with the obesity paradox (Lee and Giovannucci, 2019). Moreover, we observe that gender has a significant impact on lung cancer survival. As seen in the gradient figures, male patients exhibit a steeper increase in mortality risk compared to female patients for small increments in age and pack years, as shown in Figure 3a and c. On the other hand, Figure 3b highlights that an increased BMI has a stronger protective effect for female patients compared to male patients, consistent with previous findings of better survival outcomes for female patients (Visbal et al., 2004).

The Penalized DPLC method has selected 5 radiomic features as risk factors: large dependence low gray level emphasis (LDLGLE), large area emphasis (LAE), large area low gray level emphasis (LALGLE), cluster shade, and contrast. Figure S3 in the Supplementary Material demonstrates the reproducibility of feature selection by the Penalized DPLC and the hazard ratios for the selected features. LDLGLE (HR: 1.07) and cluster shade (HR: 1.09) were selected 71 and 57 times out of 100 experiments, respectively. Although LALGLE (Frequency: 51, HR: 1.02) and contrast (Frequency: 41, HR: 1.02) were selected less frequently than the other texture features, they were still more frequently selected by the Penalized DPLC.

The selected radiomic features have biological significance. LDLGLE and LALGLE represent the extent of low voxel intensities or soft-tissue attenuation, indicating the presence of lymphatic or vascular invasion (Higgins et al., 2012); LAE, cluster shade, and contrast quantify the roughness and heterogeneity of textures (Amadasun and King, 1989).

As shown in Figure 4, the median C-Index for Penalized DPLC is 0.708 (IQR: 0.043), outperforming the other competing methods. Deep Survival (Median: 0.672, IQR: 0.065), Random Forest (Median: 0.656, IQR: 0.080), and Cox Boosting (Median: 0.668, IQR: 0.066) all had better prediction performance than the SCAD-penalized Cox model (Median: 0.655, IQR: 0.068) and the SCAD-penalized partially linear Cox model (Median: 0.633, IQR: 0.065).

6 DISCUSSION

To address the analytical needs of the National Lung Screening Trial (NLST), we propose the Penalized DPLC model, which simultaneously selects and models the effects of prognostic radiomic features. Our adopted partial linear model assumes a log-linear relationship between radiomic features and hazards, allowing us to use the SCAD penalty to identify important image features. Clinical features with known associations with survival outcomes are modeled using a nonparametric function to account for their nonlinear effects. Despite this structured approach, we maintain the flexibility to model selected radiomic features using nonparametric functions like the clinical features. Our method provides a convenient means to explore new predictors while fully characterizing the impact of established risk factors.

There is significant potential for future work. Our modeling framework can be extended to incorporate alternative penalties, such as the LASSO and MCP (Tibshirani, 2011). We are currently utilizing a DNN estimator with a fixed and moderate dimension, which is suitable for our dataset where the number of clinical variables is moderate. It is feasible to develop DNN estimators that can handle high-dimensional predictors. Moreover, quantifying the uncertainty of the estimates remains a significant challenge.

SUPPLEMENTARY MATERIALS

Supplementary material is available at *Biometrics* online.

Figures, proofs of theorems, simulation data, and codes to run the simulations as referenced in Sections 3–5 are available with this paper at the Biometrics website on Oxford Academic.

FUNDING

The work is partially sponsored by an NIH grant (5R01CA249096) and we are grateful toward the Editor, AE and referee for their thoughtful and helpful comments.

CONFLICT OF INTEREST

None declared.

DATA AVAILABILITY

The data that support the findings in this paper may be available through a Data Use Agreement process (available at https://hcsra.sph.harvard.edu/data-use-agreement-dua) and are not publicly available due to privacy or ethical restrictions.

REFERENCES

Amadasun, M. and King, R. (1989). Textural features corresponding to textural properties. *IEEE Transactions on systems, man, and Cybernetics*, 19, 1264–1274.

Bade, B. C. and Cruz, C. S. D. (2020). Lung cancer 2020: epidemiology, etiology, and prevention. *Clinics in Chest Medicine*, 41, 1–24.

Banerjee, J., Moelker, A. and Walsum, T. (2012). 3D LBP-based invariant region description. *Asian Conference on Computer Vision*, 26–37, Springer, Berlin.

Barbeau, E. M., Li, Y., Calderon, P., Hartman, C., Quinn, M., Markkanen, P. et al. (2006). Results of a union-based smoking cessation intervention for apprentice iron workers. *Cancer Causes and Control*, 17, 53–61.

Binder, H., Allignol, A., Schumacher, M. and Beyersmann, J. (2009). Boosting for high-dimensional time-to-event data with competing risks. *Bioinformatics*, 25, 890–896.

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5, 232.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34, 187–202.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81, 425–455.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96, 1348–1360.

Fan, J. and Li, R. (2002). Variable selection for Cox's proportional hazards model and frailty model. *The Annals of Statistics*, 30, 74–99.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, JMLR Workshop and Conference Proceedings. ML Research Press, Netherlands.

Higgins, K. A., Chino, J. P., Ready, N., D'Amico, T. A., Berry, M. F., Sporn, T. et al. (2012). Lymphovascular invasion in non–small-cell lung cancer: implications for staging and adjuvant therapy. *Journal of Thoracic Oncology*, 7, 1141–1147.

Horowitz, J. L. (2009). *Semiparametric and Nonparametric Methods in Econometrics*, 12, Springer, Berlin.

Hu, Y. and Lian, H. (2013). Variable selection in a partially linear proportional hazards model with a diverging dimensionality. *Statistics and Probability Letters*, 83, 61–69.

Huang, J. (1999). Efficient estimation of the partly linear additive Cox model. *The Annals of Statistics*, 27, 1536–1563.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H. and Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, 2, 841–860.

Ji, Z., Li, J. and Telgarsky, M. (2021). Early-stopped neural networks are consistent. *Advances in Neural Information Processing Systems*, 34, 1805–1817.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T. and Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18, 1–12.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kong, H., Akakin, H. C. and Sarma, S. E. (2013). A generalized Laplacian of Gaussian filter for blob detection and its applications. *IEEE transactions on cybernetics*, 43, 1719–1733.

Lambin, P., Leijenaar, R. T., Deist, T. M., Peerlings, J., de Jong, E. E. C., van Timmeren, J. et al. (2017). Radiomics: the bridge between medical imaging and personalized medicine. *Nature Reviews Clinical Oncology*, 14, 749–762.

Lee, D. H. and Giovannucci, E. L. (2019). The obesity paradox in cancer: epidemiologic insights and perspectives. *Current Nutrition Reports*, 8, 175–181.

Leshno, M., Lin, V. Y., Pinkus, A. and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6, 861–867.

Li, M., Soltanolkotabi, M. and Oymak, S. (2020). Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *International conference on artificial intelligence and statistics*, 4313–4324, PMLR, Netherlands.

Lubner, M. G., Smith, A. D., Sandrasegaran, K., Sahani, D. V. and Pickhardt, P. J. (2017). CT texture analysis: definitions, applications, biologic correlates, and challenges. *Radiographics*, 37, 1483–1503.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48, 1875–1897.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15, 1929–1958.

Team N. L. S. T. R. (2011). Reduced lung-cancer mortality with low-dose computed tomographic screening. *New England Journal of Medicine*, 365, 395–409.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 273–282.

Tindle, H., Greevy, R. A., Vasan, R. S., Kundu, S., Massion, P. P. and Freiberg, M. S. (2018). Lifetime smoking history and risk of lung cancer: results from the framingham heart study. *JNCI: Journal of the National Cancer Institute*, 110, 1201–1207.

Visbal, A. L., Williams, B. A.,Nichols, C., F., Marks, R. S., Jett, J. R., Aubry, M.-C. et al. (2004). Gender differences in NSCL survival: an analysis of 4,618 patients diagnosed between 1997 and 2002. *The Annals of Thoracic Surgery*, 78, 209–215.

Zhong, Q., Mueller, J. and Wang, J.-L. (2022). Deep learning for the partially linear Cox model. *The Annals of Statistics*, 50, 1348–1375.

Received: March 8, 2023; Revised: September 22, 2023; Accepted: December 6, 2023