OXFORD

Briefings in Bioinformatics, 2025, **26(2)**, bbaf108
https://doi.org/10.1093/bib/bbaf108
Problem Solving Protocol

Cox-Sage: enhancing Cox proportional hazards model with interpretable graph neural networks for cancer prognosis

Ruijun Mao[1,‡], Li Wan[1,‡], Minghao Zhou[1], Dongxi Li[2,*]

[1]College of Artificial Intelligence, Taiyuan University of Technology, 79 Yingze West Avenue, Wanbailin District, Taiyuan, Shanxi Province 030024, China
[2]College of Computer Science and Technology, 79 Yingze West Avenue, Wanbailin District, Taiyuan University of Technology, Taiyuan, Shanxi Province 030024, China
*Corresponding author. College of Computer Science and Technology, Taiyuan University of Technology, 79 Yingze West Avenue, Wanbailin District, Taiyuan, Shanxi Province 030024, China. E-mail: dxli0426@126.com
‡Ruijun Mao and Li Wan contributed equally to this work

Abstract

High-throughput sequencing technologies have facilitated a deeper exploration of prognostic biomarkers. While many deep learning (DL) methods primarily focus on feature extraction or employ simplistic fully connected layers within prognostic modules, the interpretability of DL-extracted features can be challenging. To address these challenges, we propose an interpretable cancer prognosis model called Cox-Sage. Specifically, we first propose an algorithm to construct a patient similarity graph from heterogeneous clinical data, and then extract protein-coding genes from the patient's gene expression data to embed them as features into the graph nodes. We utilize multilayer graph convolution to model proportional hazards pattern and introduce a mathematical method to clearly explain the meaning of our model's parameters. Based on this approach, we propose two metrics for measuring gene importance from different perspectives: mean hazard ratio and reciprocal of the mean hazard ratio. These metrics can be used to discover two types of important genes: genes whose low expression levels are associated with high cancer prognosis risk, and genes whose high expression levels are associated with high cancer prognosis risk. We conducted experiments on seven datasets from TCGA, and our model achieved superior prognostic performance compared with some state-of-the-art methods. As a primary research, we performed prognostic biomarker discovery on the LIHC (Liver Hepatocellular Carcinoma) dataset. Our code and dataset can be found at https://github.com/beeeginner/Cox-sage.

Keywords: Cox proportional hazards model; cancer prognosis; graph neural networks; biomarker discovery

Introduction

Cancer prognosis analysis is a vital research area in the medical field. In recent years, significant progress has been made in this field due to the widespread adoption of omics data [1] and advancements in deep learning technologies [2]. The Cox proportional hazards model, one of the classical survival analysis methods, has been widely applied in cancer prognosis analysis [3]. For example, Abadi *et al.* [4] enhanced the accuracy of breast cancer prognosis by categorizing patients into eight distinct groups, integrating clinical and gene expression data, and employing Cox models with Schoenfeld residuals for validation. Similarly, Chai *et al.* [5] leveraged denoising autoencoders to extract multi-omics features, constructing Cox models and conducting Kaplan–Meier analyses to differentiate between high- and low-risk groups, further identifying nine cancer-related biomarkers through an XGboost model combining with Differential Expression Gene Analysis.

The utilization of deep learning in cancer survival prognosis has attracted growing attention. Lopez-Garcia *et al.* [6] transformed RNA-seq samples into gene expression images, facilitating the extraction of high-level features via convolutional networks. Lim *et al.* [7] employed edgeR for differential gene expression analysis, utilizing differentially expressed genes (DEGs) to build prediction models with random forests, revealing prognostic significance in three DEGs through Cox-PH survival analysis. Huang *et al.* [8] compared several deep learning-based Cox proportional hazards models for cancer prognosis, including Cox-nnet and DeepSurv. They proposed AECOX (an autoencoder with a Cox regression network) for extracting low-dimensional features for Cox proportional hazards. To mitigate issues like data noise and overfitting, researchers have developed diverse regularization and optimization strategies, such as Group Lasso for enhancing model generalization [9]. Zhan *et al.* [10] proposed using a two-stage Cox-nnet model, which significantly improved the accuracy of survival prediction for cancer patients by integrating pathological images and gene expression data.

With the advancement of deep learning, graph neural networks (GNNs) [11] are also being utilized in cancer prognosis. Wang *et al.* [12] utilized graph convolutional networks (GCNs) to extract features, integrating these with fully connected layers and Cox models to establish the GraphSurv model. Zhang *et al.* [13] proposed LAGProg, a local augmented GCN-based framework for

Received: September 19, 2024. **Revised:** February 8, 2025. **Accepted:** February 25, 2025

cancer prognosis prediction and analysis. They used conditional variational autoencoder to generate features from multi-omics data and biological networks, which were then fed into a cancer prognosis prediction model. Graph convolution was utilized to extract features, and a multilayer perceptron with a nonlinear proportional hazard objective function based on the Cox proportional hazard model was employed. Zhu *et al.* [14] introduced a novel supervised deep learning method called geometric graph neural network (GGNN). They used a GNN to incorporate geometric features obtained from the genomic network and prior biological knowledge. By employing a local-global principle to select highly predictive features and directly feeding them into the last layer for a multivariable Cox proportional-hazards model, advanced prognosis predictive performance was achieved.So Kim *et al.* [15] proposed an innovative approach utilizing GNNs to enhance survival prediction by integrating patient similarity networks and clinical data, significantly improving the accuracy of prognosis for urologic cancer patients and providing strong support for personalized treatment.

Interpretability is crucial in biomedical research, and there are also some deep learning-based cancer prognosis methods that have explored interpretability. Knottenbelt *et al.* [16] proposed the use of the CoxKAN model, which offers a novel approach to survival analysis by combining high interpretability with robust performance, addressing the critical need for reliable decision-making tools in medical practice. Jiang *et al.* [17] proposed the AutoSurv model, which entails the development and validation of a deep learning framework designed to extract pertinent information from high-dimensional gene expression and micro RNA expression data. Furthermore, they employed an interpretability methodology to mitigate the "black box" nature of deep neural networks, successfully identifying critical features that differentiate high-risk from low-risk patients in prognostic assessments.

Many deep learning-based prognostic approaches focus on feature extraction, often resulting in features that lack interpretability. Even methods that possess interpretability are rare in their ability to directly explain the importance of input features. In the stage of prognosis, numerous methods persist in using traditional Cox proportional hazards models or simplistic combinations of fully connected layers with these models. These methods overlook the similarities among patients. To address these challenges, we propose the Cox-Sage model, a powerful and interpretable cancer prognosis model. We first present an algorithm that can extract information from heterogeneous clinical data of patients to construct a similarity graph. Then, we utilize multilayer graph convolution to model the proportional hazard patterns and introduce a gradient-based mechanism for interpreting the model parameters, from which we derive two metrics [mean hazard ratio (MHZ) and reciprocal of the mean hazard ratio (RMHZ)] for discovering cancer prognosis biomarkers. Through these two metrics, we can discover cancer prognostic genes from two different perspectives: genes whose low expression levels are associated with high hazard, and genes whose high expression levels are associated with high hazard.

Materials and methods

Background

Cox proportional hazards model: Let $\mathbf{x}_i = (x_{i1}, \cdots, x_{id})^T$ be the realized values of the covariates for sample.The Cox proportional hazards model stipulates conditional hazards function as

$$h(t|\mathbf{x}_i) = h_0(t)\exp(\beta_1 x_{i1} + \cdots + \beta_d x_{id}) = h_0(t)\exp(\beta^T \mathbf{x}_i),$$

where β is a $d \times 1$ vector of unknown parameters and $h_0(t)$ is an unknown function giving hazard function for the standard set of $\beta = \mathbf{0}$ [3].This can be further generalized to

$$h(t|\mathbf{x}_i) = h_0(t)e^{\theta_i}, \tag{1}$$

where θ_i can be any function. But in this paper, our main enhancement involves integrating graph information into θ_i, resulting in an extended model.

For model evaluation, **Harrell's C − index** is the most widely used metric for the global evaluation of prognostic models in survival analysis, defined as [18]

$$\hat{C} = \frac{\sum_{i=1}^{N} \Delta_i \sum_{j=i+1}^{N} I(T_i^{obs} < T_j^{obs}) I(M_i > M_j)}{\sum_{i=1}^{N} \Delta_i \sum_{j=i+1}^{N} I(T_i^{obs} < T_j^{obs})},$$

where T_i^{obs} denotes the observed value of survival time of the ith patient, M_i represents the predicted risk score of the ith patient by model, and Δ_i equals to 1 if and only if the ith subject experiences a failure event, otherwise it is 0.

Preliminaries

Our model incorporates gene expression data and clinical data. We denote the clinical data matrix for all patients as $\mathbf{X\_clinical}_{N\times d_1} = (c_1^T, c_2^T, \cdots; c_N^T)$,where $\mathbf{c_i} \in R^{d_1\times 1}$ represents the ith patient's clinical feature vector, defined as $c_i = (c_i^1, c_i^2, \cdots, c_i^{d1})^T$. Similarly, we denote the gene expression data matrix for all patients as $\mathbf{X}_{N\times d2} = (x_1^T; x_2^T; \cdots; x_N^T)$, where $x_i = (x_i^1, x_i^2, \cdots, x_i^{d2})^T$ denotes the gene expression vector of the ith patient.

Definition 1 proportional hazards

Equation (1) can be transformed as follows, defining it as the proportional hazards, which can be interpreted as the logarithm of the multiplier by which the risk increases under the influence of covariates:

$$\theta_i = ln\left(\frac{h(t|\mathbf{x}_i)}{h_0(t)}\right). \tag{2}$$

Definition 2 patients' similarity graph

The Patients' Similarity Graph, denoted as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, consists of a vertex set $\mathbf{V}$ and an edge set $\mathbf{E}$. Here, $\mathbf{V} = \{v_1, v_2, \cdots, v_N\}$ and represents the ith patient. An edge $(v_i, v_j) \in \mathbf{E}$ exists between vertices v_i and v_j if and only if the similarity between their clinical data is high.

Model framework

Overview

The overall framework of our model is shown in Fig. 1, which consists of three modules: the graph build and feature selection module, the prognostic module, and the prognostic genes discovery module. In this section, we first propose an algorithm for extracting information from heterogeneous clinical data of patients to construct a similarity graph. Then, we introduce the concepts and formulas of our Cox-Sage model as shown in Fig. 1(a,b). Finally, we provide a mathematical interpretation of the model's parameters and present two metrics for the discovery of prognostic genes as shown in Fig. 1(c).

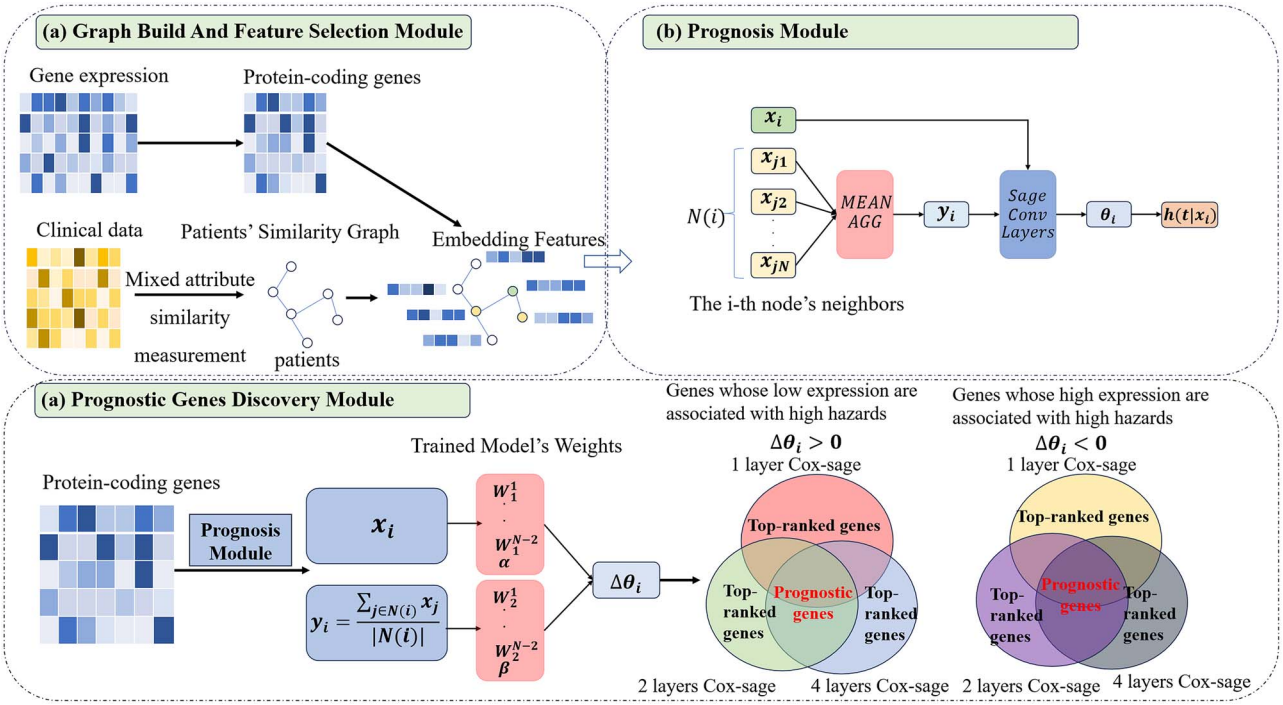

Figure 1. Data preprocessing and proposed models.

Building patients' similarity graph algorithm

In clinical data, the heterogeneity is typically manifested through four primary types of attributes: ordinal, nominal, binary, and numeric. For the kth clinical feature of the ith patient, we denote the ordinal value as r_i^k,with M_k representing the maximum ordinal value observed across all patients for the kth feature. For numeric features, normalization is essential to ensure that the distance $d(i,j)$'s falls within the range $[0,1]$. We introduce a straightforward algorithm designed to construct a Patients' Similarity Graph. The detailed content can be seen in Algorithm 1, where the algorithm traverses all pairs of nodes, calculates the distance matrix based on a mixed attribute distance measurement to obtain the similarity matrix, and finally derives the edges of the patient similarity graph through threshold processing.

Integrating graph information to Cox proportional hazards model

To aggregate information from neighbor nodes, our strategy is to compute the average features of neighboring nodes and then add this average to the Proportional Hazards model. We also add a bias, which helps the model to acquire more information about the unknown baseline hazard $h_0(t)$

$$h(t|\mathbf{X}_i) = h_0(t)e^{(\alpha^T\mathbf{x}_i+b)+\frac{1}{|N(i)|}(\beta^T\sum_{j\in N(i)}X_j)}, \tag{3}$$

where $j \in N(i)$ represents all the neighboring nodes in the Patients' Similarity Graph of the ith patient. $h_0(t)$ is a unknown baseline hazard, where $\alpha, \beta \in \mathbb{R}^{d2\times 1}$ and $b \in \mathbb{R}$ are trainable parameters. Comparing with proportional hazards model, the model's proportional hazards equation (2) is

$$\theta_i = (\alpha^T\mathbf{x}_i + b) + \frac{1}{|N(i)|}\left(\beta^T\sum_{j\in N(j)}X_j\right). \tag{4}$$

For the unknown baseline hazard $h_0(t)$, we simply assume it to be a constant 1, which essentially allows the bias of our model to learn the information of $h_0(t)$. However, in the subsequent theoretical derivations, for the sake of mathematical rigor, we still include this term in the equations, even though we set it to be a constant 1 in the experiments.

This model uses negative partial log-likelihood as the loss function, We also introduce a weight decay term to prevent overfitting, where λ is the weight decay coefficient

$$\mathbf{pl}(\alpha,\beta,\mathbf{b}) = -\sum_{C(i)=1}(\theta_i - \sum_{t_j\geq t_i}\exp(\theta_j)) + \frac{1}{2}\lambda\left(\sum_{k0=1}^{d2}(\alpha_{k0}^2+\beta_{k0}^2)+b^2\right). \tag{5}$$

By minimizing the loss function specified in equation (5), we can train the model parameters $\alpha, \beta, \mathbf{b}$.

Cox-Sage: multilayers graph convolution proportional hazards model

As a type of graph convolution in GNN, the GraphSage convolution operator is formally defined as follows [19]:

$$\mathbf{x}_i' = \mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2\cdot\oplus_{j\in\mathcal{N}(i)}\mathbf{x}_j,$$

where $\mathbf{x}_i$ represents feature vector embedded into node i. $\mathbf{x}_i$ is new feature vector after GraphSage convolution compute, and $j \in N(i)$ signifies all neighbors of node i, and ⊕ represents a differentiable aggregation function.

The proportional hazards equation (4) is tantamount to a single layer of the GraphSage convolution when the differentiable aggregation function ⊕ is taking average. This graph convolution operator not only aggregates information from neighboring nodes but also maintains a degree of independence, facilitating

interpretative analysis. We propose the following Cox-Sage model, where $W_k^1, W_k^2 \in \mathbb{R}^{l_k \times l_{k-1}}$ and $b_k \in \mathbb{R}$ for $k = 1, 2, \ldots, N$ are all trainable parameters, with l_k representing the node embedding dimension of the kth layer of the GCN. In particular, since the output layer dimension is 1, the parameter matrices W_N^1 and W_N^2 degenerate into parameter vectors. For the convenience of subsequent proofs and discussions, we denote W_N^1, W_N^2, and b_N as α, β, and b, respectively.

$$
\begin{cases}
\mathbf{x}_i^1 = \mathbf{W}_1^1 \mathbf{x}_i + \mathbf{b}_1 + \dfrac{1}{|N(i)|}\mathbf{W}_2^1 \cdot \sum\limits_{j \in N(i)} \mathbf{x}_j, \\
\mathbf{x}_i^2 = \mathbf{W}_1^2 \mathbf{x}_i^1 + \mathbf{b}_2 + \dfrac{1}{|N(i)|}\mathbf{W}_2^2 \cdot \sum\limits_{j \in N(i)} \mathbf{x}_j^1, \\
\quad \cdots \\
\mathbf{x}_i^k = \mathbf{W}_1^k \mathbf{x}_i^{k-1} + \mathbf{b}_k + \dfrac{1}{|N(i)|}\mathbf{W}_2^k \cdot \sum\limits_{j \in N(i)} \mathbf{x}_j^{k-1} \\
\quad \cdots \\
\mathbf{x}_i^{N-1} = \mathbf{W}_1^{N-1} \mathbf{x}_i^{N-2} + \mathbf{b}_{N-1} + \dfrac{1}{|N(i)|}\mathbf{W}_2^{N-1} \cdot \sum\limits_{j \in N(i)} \mathbf{x}_j^{N-2} \\
\theta_i = (\alpha^T \mathbf{x}_i^{N-1} + \mathbf{b}) + \dfrac{1}{|N(i)|}(\beta^T \sum\limits_{j \in N(i)} \mathbf{x_j^{N-1}}) \\
h(t|x_i) = h_0(t) e^{\theta_i}
\end{cases}
. \tag{6}
$$

Due to the good mathematical interpretability of linear models, we did not introduce an activation function. We will discuss this viewpoint in detail later. And we set the baseline hazard $h_0(t)$ to a constant value of 1. This model uses negative partial log-likelihood as the loss function. We also introduce a weight decay term to prevent overfitting, where λ is the weight decay coefficient

$$
\begin{aligned}
&\mathbf{pl}(\alpha, \beta, b, W_1^{N-1}, W_2^{N-1}, \cdots, W_1^1, W_2^1) = \\
&-\sum_{C(i)=1} \left(\theta_i - \sum_{t_j \geq t_i} \exp(\theta_j) \right) \\
&+ \frac{1}{2}\lambda \sum_{k=1}^{N} \left(\sum_{j_0=1}^{l_{k-1}} \sum_{i_0=1}^{l_k} \left(W_1^k\right)_{i_0 j_0}^2 + \sum_{j_0=1}^{l_{k-1}} \sum_{i_0=1}^{l_k} \left(W_2^k\right)_{i_0 j_0}^2 + b_k^2 \right).
\end{aligned} \tag{7}
$$

By minimizing the loss function specified in equation (7), we can train the model parameters $\alpha, \beta, \mathbf{b}, \mathbf{W_1^{N-1}}, \mathbf{W_2^{N-1}}, \cdots, \mathbf{W_1^1}, \mathbf{W_2^1}$.

Model interpretation: based on gradient and linear transformation

Parameters meaning

Let $\mathbf{x}_i = (x_i^1, x_i^2, \cdots, x_i^k, \cdots, x_i^{d2})^T$ represent the gene expression vector of the ith patient, and $y_i = \frac{1}{|N(i)|}\sum_{j \in N(i)} x_j = (y_i^1, y_i^2, \cdots, y_i^k, \cdots, y_i^{d2})$ denote the average gene expression vector of its neighboring nodes within the Patients' Similarity Graph. Taking into account the changes Δx and Δy affecting the kth gene's expression level, we define $x_i' = (x_i^1, x_i^2, \cdots, x_i^k + \Delta x, \cdots, x_i^{d2})^T$ and $\mathbf{y}_i' = (y_i^1, y_i^2, \cdots, y_i^k + \Delta y, \cdots, y_i^{d2})$. By maintaining the expression values of other genes constant, we can assess the hazard ratio before and after these changes

$$
\frac{h(t|\mathbf{x}'_i)}{h(t|\mathbf{x}_i)} = \frac{h_0(t)\exp(\theta'_i)}{h_0(t)\exp(\theta_i)} = \exp(\theta'_i - \theta_i). \tag{8}
$$

In our model, θ_i represents the multilayer GraphSage convolution. Since we hold the expression values of other genes fixed, we can treat the difference in equation (8) as a bivariate function of x_i^k and y_i^k:

$$
\Delta\theta = \theta'_i - \theta_i = \theta(x_i^k + \Delta x, y_i^k + \Delta y) - \theta(x_i^k, y_i^k). \tag{9}
$$

For convenience of discussion, we denote the two partial derivatives of this bivariate function as $\theta_{x_i^k} = \frac{\partial \theta_i}{\partial x_i^k}$ and $\theta_{y_i^k} = \frac{\partial \theta_i}{\partial y_i^k}$, respectively. According to the mean value theorem for multivariable functions, there exists at least one ξ $(0 < \xi < 1)$ such that

$$
\begin{aligned}
&\Delta\theta = \theta'_i - \theta_i = \theta(x_i^k + \Delta x, y_i^k + \Delta y) - \theta(x_i^k, y_i^k) = \\
&\theta_{x_i^k}(x_i^k + \xi\Delta x, y_i^k + \xi\Delta y)\Delta x + \theta_{y_i^k}(x_i^k + \xi\Delta x, y_i^k + \xi\Delta y)\Delta y,
\end{aligned}
$$

if and only if (x_i^k, y_i^k) belong to a two-dimensional convex set. In our case,$(x_i^k, y_i^k) \in R_+^2$,which is a convex set. Since our model is linear, this ensures that the two partial derivatives $\theta_{x_i^k}(x_i^k + \xi\Delta x, y_i^k + \xi\Delta y)$ and $\theta_{y_i^k}(x_i^k + \xi\Delta x, y_i^k + \xi\Delta y)$ take on constant values, and are also independent of the value of ξ. Furthermore, the gradient vector of

Algorithm 1 Constructing Patient Similarity Graph

Require: $\mathbf{X\_clinical}_{N \times d_1} = (\mathbf{c}_1^T; \mathbf{c}_2^T; \cdots; \mathbf{c}_N^T)$;
Feature weights $\mathbf{f} = (\delta^{(1)}, \delta^{(2)}, \cdots, \delta^{(d1)})$ where $\sum_{k=1}^{d_1} \delta^{(k)} = 1$;
Features set $\mathbf{A} = (a^1, a^2, \cdots, a^{(d1)})$.
Ensure:
Initialize $\mathbf{V} = \{v_1, v_2, \cdots, v_N\}$, $\mathbf{E} = \{\}$, *Distance* $= \mathbf{O}_{N \times N}$,
Similarity $= \mathbf{O}_{N \times N}$.
for each v_i in **V do**
  for each v_j in **V do**
    for each $a^{(k)}$ in **A do**
      if $a^{(k)}$ is nominal or binary feature **then**
        Compute distance $d_{ij}^{(k)} = \begin{cases} 0 & c_i^k = c_j^k \\ 1 & \text{else} \end{cases}$
      else if $a^{(k)}$ is ordinal feature **then**
        Normalize by $z_i^k = \frac{r_i^k - 1}{M_k - 1}$ and $z_j^k = \frac{r_j^k - 1}{M_k - 1}$.
        Compute distance $d_{ij}^{(k)} = |z_i^k - z_j^k|$.
      else if $a^{(k)}$ is numerical feature **then**
        Compute distance $d_{ij}^{(k)} = |c_i^k - c_j^k|$.
      end if
    end for
    $Distance(i,j) = \sum_{k=1}^{d1} \delta^{(k)} d_{ij}^{(k)}$.
    $Similarity(i,j) = 1 - Distance(i,j)$.
    if $similarity(i,j) > threshold$ **then**
      Add (v_i, v_j) to edges set **E**.
    end if
  end for
end for
return : $\mathbf{G} = (\mathbf{V}, \mathbf{E})$;

this bivariate function, denoted as

$$\left\langle \frac{\partial \Delta\theta}{\partial \Delta x}, \frac{\partial \Delta\theta}{\partial \Delta y} \right\rangle = \langle \theta_{x_i^k}(x_i^k + \xi\Delta x, y_i^k + \xi\Delta y), \theta_{y_i^k}(x_i^k + \xi\Delta x, y_i^k + \xi\Delta y) \rangle,$$

is a constant vector that is independent of the changes in gene expression levels Δx and Δy. Moreover, the gradient vector $\langle \frac{\partial \Delta\theta}{\partial \Delta x}, \frac{\partial \Delta\theta}{\partial \Delta y} \rangle$ points in the direction of the fastest increase in hazard. This indicates that by using a linear model, we can learn a consistent pattern of increasing hazards for each gene, which is easily interpretable.

To enable the quantification and calculation of equation (8), we will now introduce a trick for computing $\theta_{x_i^k}$ and $\theta_{y_i^k}$. We will start with the simple case derived from a one-layer model and then extend to the general case for an N-layer model.

(i) One-layer Cox-Sage model

In this case, θ_i is given by the equation described in equation (4); the gradient of θ_i with respect to $\mathbf{x}_i$ is

$$\frac{\partial \theta_i}{\partial x_i} = \alpha = (\alpha_1, \alpha_2, \cdots, \alpha_k, \cdots, \alpha_{d2})^T.$$

Similarly, we can obtain that

$$\frac{\partial \theta_i}{\partial y_i} = \beta = (\beta_1, \beta_2, \cdots, \beta_k, \cdots, \beta_{d2})^T.$$

It is obvious that $\theta_{x_i^k} = \alpha_k$ and $\theta_{y_i^k} = \beta_k$. For the one-layer Cox-Sage model, the impact of the changes Δx and Δy is

$$\theta(x_i^k + \Delta x, y_i^k + \Delta y) - \theta(x_i^k, y_i^k) = a_k \Delta x + \beta_k \Delta y. \quad (10)$$

(ii) N-layer Cox-Sage model

In this case, θ_i is given by the equations described in equation (6); based on the chain rule, the gradient of θ_i with respect to x_i is

$$\theta_{x_i^k} = \frac{\partial \theta_i}{\partial x_i} = \left(\frac{\partial x_i^{N-1}}{\partial x_i^{N-2}} \frac{\partial x_i^{N-2}}{\partial x_i^{N-3}} \cdots \frac{\partial x_i^1}{\partial x_i}\right)^T \frac{\partial \theta_i}{\partial x_i^{N-1}} = ((\mathbf{W}_1^{N-1}\mathbf{W}_1^{N-2}\cdots\mathbf{W}_1^1)^T\alpha)_k.$$

Similarly, we can obtain that

$$\theta_{y_i^k} = \frac{\partial \theta_i}{\partial y_i} = ((\mathbf{W}_2^{N-1}\mathbf{W}_2^{N-2}\cdots\mathbf{W}_2^1)^T\beta)_k.$$

For the N-layer Cox-Sage model, the impact of the changes Δx and Δy is

$$\theta(x_i^k + \Delta x, y_i^k + \Delta y) - \theta(x_i^k, y_i^k) = ((\mathbf{W}_1^{N-1}\mathbf{W}_1^{N-2}\cdots\mathbf{W}_1^1)^T\alpha)_k \Delta x + ((\mathbf{W}_2^{N-1}\mathbf{W}_2^{N-2}\cdots\mathbf{W}_2^1)^T\beta)_k \Delta y. \quad (11)$$

This reveals that parameters of the one-layer Cox-Sage model equation (3) directly determine the hazard change, while the N-layer Cox-Sage model equation (6) utilizes parameter matrices to linear transform the parameters before determining the hazard change. The more the layers, the more transformation matrices are involved, resulting in a more complex modeling ability of hazard changes. At the same time, as we discussed earlier, our model is linear, which can learn a consistent pattern of increasing hazards for each gene, making it easily interpretable.

Prognostic genes discovery

We can simulate the scenario where the expression level of the kth gene is set to 0 by assigning $\Delta x = -x_i^k$ and $\Delta y = -y_i^k$. To further explore the impact of the expression level of the kth gene being zero on cancer prognosis risk, we conducted the analysis using Equation (9):

$$\begin{aligned}\frac{h(t|\mathbf{x}'_i)}{h(t|\mathbf{x}_i)} &= \frac{h_0(t)\exp(\theta'_i)}{h_0(t)\exp(\theta_i)} = \exp(\theta'_i - \theta_i) \\ &= \exp(\theta(x_i^k + \Delta x, y_i^k + \Delta y) - \theta(x_i^k, y_i^k)) \\ &= \exp(\theta(0,0) - \theta(x_i^k, y_i^k)) \\ &= \begin{cases} \exp\left(-\left(\alpha x_i^k + \beta_k y_i^k\right)\right), & \text{for a single layer model} \\ \exp\left(-\left(\left(\left(W_1^{N-1}W_1^{N-2}\cdots W_1^1\right)^T\alpha\right)_k x_i^k + \left(\left(W_2^{N-1}W_2^{N-2}\cdots W_2^1\right)^T\beta\right)_k y_i^k\right)\right), & \text{for a multilayer model} \end{cases}.\end{aligned}$$

When the ratio is >1 (same as $\Delta\theta_i > 0$), it indicates an increase in hazards due to the corresponding gene expression level being 0. Conversely, when the ratio is <1(same as $\Delta\theta_i < 0$), it suggests a decrease in hazards due to the corresponding gene expression level being 0. Based on this, we propose the following two metrics to measure the importance of genes.

For each gene in every sample, we can conduct a hazard ratio analysis according to the method described above. Hence, our initial evaluation metric is the MHZ, which is calculated by averaging the hazard ratios across all samples. As explained before, a higher value of this metric signifies a greater elevation in the prognostic risk of cancer associated with the expression level of the kth gene being 0. By sorting all the genes based on their MHZs, we can identify a subset of genes whose low expression levels are associated with high prognostic risk:

$$MHZ = E_{x_i \sim p(x_i)}\left[\exp(\theta'_i - \theta_i)\right]. \quad (12)$$

Conversely, our second evaluation metric is the *RMHZ*, which is obtained by calculating the inverse of *MHZ*. As previously discussed, a higher value of this metric indicates a more significant reduction in the prognostic risk of cancer associated with the expression level of the kth gene being 0. By sorting all the genes based on their *RMHZ* values, we can identify genes whose high expression levels are associated with low prognostic risk:

$$RMHZ = E_{x_i \sim p(x_i)}\left[\exp(-(\theta'_i - \theta_i))\right]. \quad (13)$$

By analyzing these two metrics, we can approach the discovery of cancer prognostic genes for cancer from two perspectives: genes whose low expression levels are associated with high prognostic risk or genes whose high expression levels are associated with high prognostic risk.

Results

Data and experiment

Datasets and preprocessing

For this research, we obtained gene expression and clinical data for seven different types of cancer from The Cancer Genome

Table 1. Samples after preprocessing: retained protein-coding genes and removed samples with incomplete data

Dataset	Samples	Death count	Gene count
LIHC	368	88	19 938
LUAD	485	117	19 938
STAD	406	79	19 938
LUSC	489	158	19 938
COAD	453	56	19 938
HNSC	520	167	19 938
ESCA	184	57	19 938

Table 2. Clinical features used in each dataset. (✓) (×)

Feature	COAD	ESCA	HNSC	LUAD	LUSC	STAD	LIHC
Age	✓	✓	✓	✓	✓	✓	✓
Gender	✓	✓	✓	✓	✓	✓	✓
Race List	✓	✓	✓	✓	✓	×	✓
Anatomic Neoplasm Subdivision	×	×	✓	✓	×	×	×
ICD-10	✓	✓	✓	✓	✓	×	✓
Neoplasm Histologic Grade	×	×	✓	×	×	✓	×
Person Neoplasm Cancer Status	✓	✓	✓	✓	×	×	✓
Stage Event	✓	✓	✓	✓	✓	×	✓
Tissue Source Site	✓	✓	✓	✓	✓	✓	✓

Atlas (TCGA) database, accessible at https://portal.gdc.cancer.gov/. They are as follows: Lung Squamous Cell Carcinoma (LUSC), Stomach Adenocarcinoma (STAD), Lung Adenocarcinoma (LUAD), Head and Neck Squamous Cell Carcinoma (HNSC), Esophageal Carcinoma (ESCA), Liver Hepatocellular Carcinoma (LIHC), and Colon Adenocarcinoma (COAD).

All of our models and data processing pipeline are showed in Fig. 1. We retain samples with complete gene expression and clinical data, specifically preserving data on protein-coding genes Fig. 1(a). Clinical features varied across datasets, and missing values in pathological data are marked as "UNKNOWN." For demographic features such as race, missing values were imputed using either the mode or a Random Forest model. Samples with excessive missing values are excluded from the analysis.

We first build the Patients' Similarity Graph using Algorithm 1. For the selection of the similarity threshold for constructing edges, in order to avoid generating overly dense graphs, we set the threshold to $Q_3 + 1.5 \times \text{IQR}$, where Q_3 is the upper quartile of the similarity matrix values and IQR is the interquartile range. This way, we can establish edges only between nodes with high similarity. Then we use transcriptome profiling gene expression quantification data as features embedded in graph. When selecting quantification data, we opt for raw counts and perform Log2 transformation, basing on the study by Zhao *et al.* [20]. We only preserve protein-coding genes. Then we embed the gene expression quantification data to the graph. The number of samples, statistics of deceased samples, and gene counts of the processed dataset are shown in Table 1. In the corresponding clinical data, heterogeneity is evident. We selected several demographic and pathological features for each dataset to construct the similarity graph of patients. The specific clinical data features used are presented in Table 2. It includes demographic features (such as age, gender, etc.) as well as pathological features (like ICD-10). A checkmark (✓) indicates that the dataset contains the feature, while a cross (×) indicates that the feature is not present.

Experiment set

Cancer prognosis module

Due to the large and dense dimensionality of the gene expression data we used, we must take computational costs into account (in our practical experiments, training a four-layer Cox-Sage model requires ∼36GB of GPU memory), which means we need to train models with fewer layers. Additionally, to avoid the over-smoothing problem caused by stacking layers in the GNN [21], we separately train a one-layer Cox-Sage model as described in equation (3), along with two-layer and four-layer Cox-Sage models as described in equation (6), across all datasets. The model parameters are obtained by minimizing the objective functions in equation (5) and equation (7).

Prognostic genes discovery module

To investigate the computational metrics *MHZ* (equation (12)) and *RMHZ* (equation (13)) we proposed for feature importance ranking, we conducted the discovery of prognostic biomarkers for liver cancer using the trained models on LIHC. Specifically, we extracted the parameters from the trained one-layer, two-layer, and four-layer Cox-Sage models, and then used these to compute the rankings of each gene's *MHZ* and *RMHZ* values according to the formulas. To maintain the stability of our method for gene importance ranking, we retain half of the samples with values greater than the median and take the intersection of all models (see Fig. 1(c)). This approach allows us to identify prognostic biomarkers for liver cancer from two different perspectives.

Training setup

We utilize the Adam optimizer [22] of PyTorch [23] to optimize the negative log-likelihood function mentioned in equation (5) and equation (7) to learn models' parameters. For each dataset, 20% is reserved as a test set and is not involved in the model training/tuning process. We perform model tuning on the

Table 3. Performance Comparison of Different Models. ±

Dataset	Graph-Surv	LAG-Prog	GGNN	Cox-EN	Cox-AE	Auto-Surv	Cox-KAN	Cox-Sage (One layer)	Cox-Sage (Two layers)	Cox-Sage (4 layers)
COAD	–	0.774	0.592	0.548 ±0.065	0.574 ± 0.021	0.734 ± 0.027	0.617 ± 0.019	0.768 ± 0.034	**0.779 ± 0.027**	0.771 ± 0.009
LUAD	0.646	0.667	0.638	0.611 ± 0.004	0.621 ± 0.037	0.715 ± 0.038	0.628 ± 0.024	0.721 ± 0.046	0.809 ± 0.037	**0.835 ± 0.010**
HNSC	0.637	0.654	0.615	0.546 ± 0.022	0.571 ± 0.015	0.698 ± 0.013	0.583 ± 0.002	0.690 ± 0.023	0.858 ± 0.094	**0.865 ± 0.070**
STAD	0.592	0.710	–	0.612 ± 0.011	0.568 ± 0.008	0.687 ± 0.017	0.613 ± 0.011	0.646 ± 0.060	0.744 ± 0.138	**0.812 ± 0.068**
ESCA	–	0.729	–	0.693 ± 0.006	0.682 ± 0.071	0.729 ± 0.022	0.684 ± 0.014	0.706 ± 0.118	0.934 ± 0.023	**0.959 ± 0.012**
LIHC	0.684	0.756	0.672	0.551 ± 0.019	0.563 ± 0.018	0.710 ± 0.021	0.627 ± 0.012	0.735 ± 0.044	**0.782 ± 0.027**	0.738 ± 0.018
LUSC	0.529	0.625	–	0.545 ± 0.017	0.593 ± 0.015	0.678 ± 0.018	0.603 ± 0.014	0.749 ± 0.047	0.774 ± 0.047	**0.815 ± 0.022**

remaining 80% of the dataset. Specifically, we further randomly split this portion into an 80% training set and a 20% validation set (which corresponds to 64% and 16% of the total data, respectively). This random splitting process is repeated five times (i.e. five-fold cross-validation) to find the optimal model. Additionally, we use early stopping mechanism, at each epoch; if the C-index value of the validation set do not improve for a given patience times, the training is terminated early. We select the model with the best performance in five cross-validation and test its performance on the validation set. For all our models, the learning rate is set to 0.00001, weight decay to 0.001, and the number of training epochs to 20, with an early stopping tolerance of 3. To better demonstrate the stability of our model, we conducted experiments using five different random seeds and calculated the mean and standard deviation of our model's performance, as shown in Table 3. For the Cox-Sage, CoxKAN, AutoSurv, Cox-EN, and Cox-AE models, we report the mean ± standard deviation based on multiple repeated experiments. For other benchmark methods, we present the best performance reported in the respective papers.

Discussion

C-index comparison across benchmarks

In seven TCGA datasets, we compared our method with the latest approaches, including GraphSurv, LAGProg, GGNN, CoxKAN, and AutoSurv [12–14, 16, 17]. We also compared with some classical methods, including Cox-EN and Cox-AE [24, 25]. For GraphSurv, LAGProg, and GGNN, we directly utilized the best experimental results reported on the overlapping datasets in their respective papers. For CoxKAN, AutoSurv, Cox-EN, and Cox-AE, we reproduced the results using the experimental protocols or code provided in their respective papers, and performed systematic hyperparameter tuning to ensure a fair comparison. Additionally, we applied five-fold cross-validation for each model and reported the best results. We also repeated the experiments with five different random seeds and computed the mean and standard deviation. The reproduction code and prediction output results of these four benchmark methods can be found in the GitHub repository: https://github.com/beeeginner/benchmarks_compare.

The performance comparison of our three models with other models across seven datasets is presented in Table 3, with the evaluation metric being Harrell's C-index. We have bolded the C-index values of the best-performing models. It can be observed that our models achieve the most advanced performance.

Kaplan–Meier survival analysis: a case study on the LIHC dataset

Due to space limitations, we only conduct survival analysis on the LIHC dataset as a primary research. We choose a high-performing model and divide the patients into two groups of high and low risk based on the median of the model's outputs. Then we perform Kaplan–Meier survival analysis. Subsequently, we plot survival curves for each group separately and conducted log-rank tests. The survival curves significantly separate on this dataset ($c-index > 0.6, P-value < 0.1$), as showed in Fig. 2. We group based on the median of the model's predicted outputs. Samples with risk values above the median are categorized into the high-risk group, while those below the median are placed in the low-risk group. In the figure, blue represents low-risk samples, while orange represents high-risk samples. The 1 layer Cox-Sage model has a C-index of 0.672 (log-rank $P < 0.005$), the 2 layers Cox-Sage model has a C-index of 0.778 (log-rank $P < 0.005$), and the 4 layers Cox-Sage model has a C-index of 0.742 (log-rank $P < 0.005$).

Cancer prognostic genes discovery: a case study on the LIHC dataset

We can rank the importance of genes by calculating the *MHZ* (equation (12)) and the *RMHZ* (equation (13)) using the trained model's parameters combined with the dataset. However, since we train three models for each of the seven datasets and perform five-fold cross-validation for each model, the storage cost for all these parameters is considerable. Therefore, we use the LIHC dataset to evaluate the feature importance ranking ability of our model, where the features represent genes. We extracted the parameters of the trained one-layer, two-layer, and four-layer Cox-Sage models, substituted them into equation (12) and equation (13), and obtained the gene importance rankings separately. Finally, we took the intersection of the genes with MHZ or RMHZ values greater than the median from the three models. Finally, among the 19 938 protein-coding genes, our model identified 2456 genes whose low expression levels are associated with high prognostic risk, and 2487 genes whose high expression levels are associated with high prognostic risk. We have uploaded the code for our method, the processed dataset, prediction results, and benchmark results to GitHub. Detailed instructions are provided in the README file at https://github.com/beeeginner/Cox-sage.

Additionally, basing on equation (10), equation (11), we select some representative prognostic genes from each category identified by our model to draw contour plot of hazards changes, which are presented in Fig. 3.Our model reveals that high expression levels of CD69 correlate with high risk, whereas low expression levels of GLUL also lead to high risk.

To demonstrate the effectiveness of our method, based on our method, we selected 20 genes closely associated with the prognosis risk of liver hepatocellular carcinoma (HCC) from the genes identified by our model, which are presented in Table 4. Additionally, we conducted a literature review and found that 17 of these genes are related to liver cancer, while the remaining

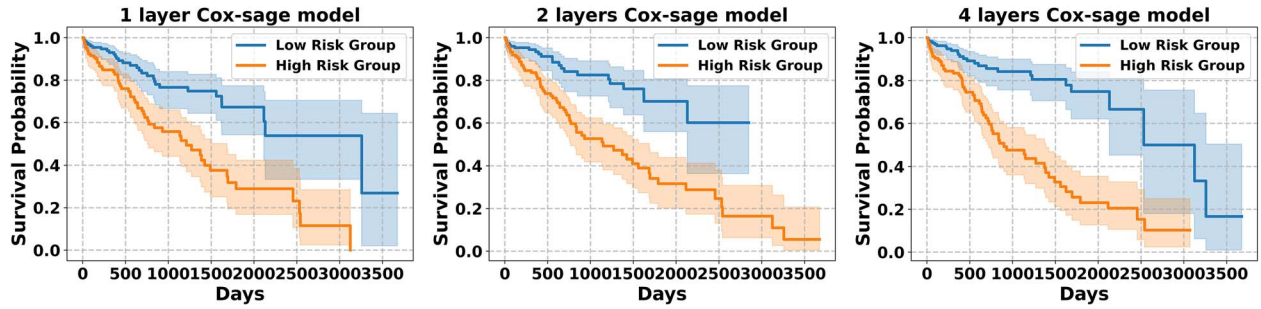

Figure 2. The Kaplan–Meier Survival Analysis Curves of LIHC dataset.

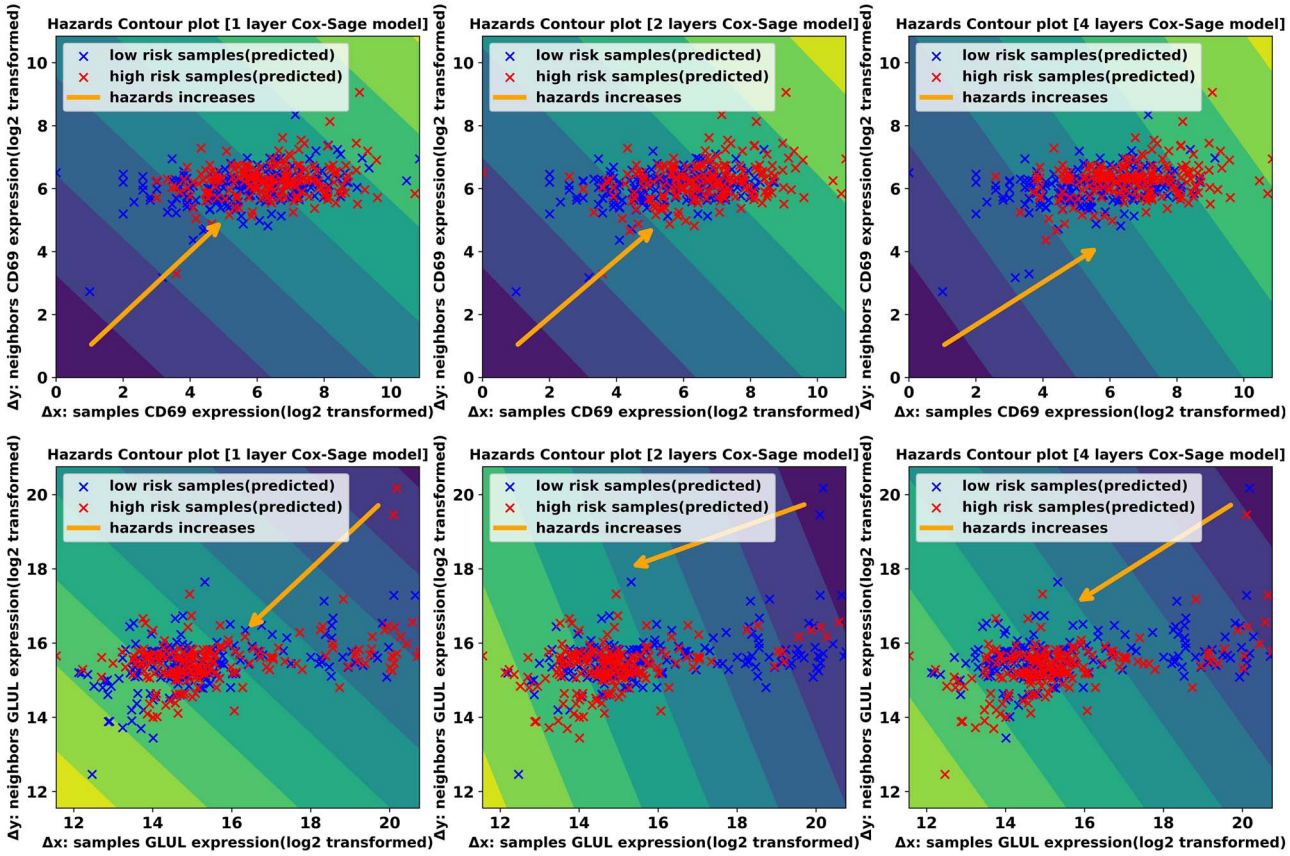

Figure 3. Hazards contour plot for three genes associated with LIHC (CD69,GLUL).

three are associated with other cancers: BET1L (cancer related [26]), CBR4 (cancer related [27]), AHCY (cancer related [28]), IGFBP5 (HCC related [29]), RPS7 (HCC related [30]), PRKAG2 (HCC related [31]), TNFRSF1A (HCC related [32]), HNRNPUL2 (HCC related [33]), APOH (HCC related [34]), MMP11 (HCC related [35]), ANTXR1 (HCC related [36]), CD69 (HCC related [37]), EEF1E1 (HCC related [38]), FOXP4 (HCC related [39]), MAP3K2 (HCC related [40]), CTSZ (HCC related [41]), CWF19L1 (HCC related [42]), DPH2 (HCC related [43]), GLUL (HCC related [44]), ISOC1 (HCC related [45]). Due to length constraints, we did not review the remaining prognostic genes. We encourage readers to explore the literature or conduct their own research on these genes.

Table 4. Discovery of 20 genes closely related to hepatocellular carcinoma prognosis by our model

Low expression levels associated with a higher risk
TNFRSF1A, AHCY, PRKAG2, APOH, MMP11, CTSZ, CWF19L1, DPH2, GLUL, ISOC1
High expression levels associated with a higher risk
CBR4, RPS7, HNRNPUL2, IGFBP5, BET1L, ANTXR1, CD69, EEF1E1, FOXP4, MAP3K2

Conclusion

In this paper, we investigate enhancing cancer prognosis performance and model interpretability. Specifically, we introduce model Cox-Sage, which extracts graph information from clinical data, constructs proportional hazards patterns using the multilayer GraphSage convolution, and proposes a mathematical method for interpreting model parameters. Based on this method, we also propose two metrics for cancer prognostic genes discovery. Finally, the experimental results demonstrate that our model

outperforms other prognostic models and effectively identifies potential prognostic biomarkers.

Key Points

- We proposed the Cox-Sage model, an interpretable GNN designed for cancer prognosis, which can be utilized for the identification of cancer prognostic biomarkers.
- Our model achieved the state-of-art C-index scores at seven different datasets, demonstrating its ability to effectively distinguish cancer patient risk.
- Our model can be more effective to identify potential cancer prognostic biomarkers at different expression levels.

Conflict of interest

No competing interest is declared.

Funding

This work was supported by the Basic Research Programs of Shanxi Province (202303021211069, 202403021222070).

Code availability

The GitHub link for the code of our method can be found at https://github.com/beeeginner/Cox-sage, and the code for other benchmark methods that we reproduced can be found at https://github.com/beeeginner/benchmarks_compare.

Data availability

We have uploaded the dataset to Kaggle and Zendo, where the Zendo link includes all the reproduced benchmark prediction results. The specific links and access methods are provided in the README file at the GitHub link: https://github.com/beeeginner/Cox-sage.

References

1. Menyhárt O, Gyoőrffy B. Multi-omics approaches in cancer research with applications in tumor subtyping, prognosis, and diagnosis. *Comput Struct Biotechnol J* 2021;**19**:949–60. https://doi.org/10.1016/j.csbj.2021.01.009
2. Zhu W, Xie L, Han J. *et al.* The application of deep learning in cancer prognosis prediction. *Cancer* 2020;**12**:603. https://doi.org/10.3390/cancers12030603
3. Cox DR. Regression models and life-tables. *J R Stat Soc B Methodol* 1972;**34**:187–202. https://doi.org/10.1111/j.2517-6161.1972.tb00899.x
4. Abadi A, Yavari P, Dehghani-Arani M. *et al.* Cox models survival analysis based on breast cancer treatments. *Iran J Cancer Prev* 2014;**7**:124–9.
5. Chai H, Zhou X, Zhang Z. *et al.* Integrating multi-omics data through deep learning for accurate cancer prognosis prediction. *Comput Biol Med* 2021;**134**:104481. https://doi.org/10.1016/j.compbiomed.2021.104481
6. Lopez-Garcia G, Jerez JM, Franco L. *et al.* Transfer learning with convolutional neural networks for cancer survival prediction using gene-expression data. *PloS One* 2020;**15**:e0230536. https://doi.org/10.1371/journal.pone.0230536
7. Lim GB, Kim Y-A, Seo J-H. *et al.* Prediction of prognostic signatures in triple-negative breast cancer based on the differential expression analysis via nanostring ncounter immune panel. *BMC Cancer* 2020;**20**:1–13. https://doi.org/10.1186/s12885-020-07399-8
8. Huang Z, Johnson TS, Han Z. *et al.* Deep learning-based cancer survival prognosis from rna-seq data: Approaches and evaluations. *BMC Med Genomics* 2020;**13**:1–12.
9. Xie G, Dong C, Kong Y. *et al.* Group lasso regularized deep learning for cancer prognosis from multi-omics and clinical features. *Genes (Basel)* 2019;**10**:240. https://doi.org/10.3390/genes10030240
10. Zhan Z, Jing Z, He B. *et al.* Two-stage Cox-nnet: Biologically interpretable neural-network model for prognosis prediction and its application in liver cancer survival using histopathology and transcriptomic data. *NAR Genomics Bioinformatics* 2021;**3**:lqab015. https://doi.org/10.1093/nargab/lqab015
11. Sanchez-Lengeling B, Reif E, Pearce A. *et al.* A gentle introduction to graph neural networks. *Distill* 2021;**6**:e33. https://doi.org/10.23915/distill.00033
12. Wang Y, Zhang Z, Chai H. *et al.* Multi-omics cancer prognosis analysis based on graph convolution network. In: *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1564–8. Houston, TX, USA: IEEE, 2021.
13. Zhang Y, Xiong S, Wang Z. *et al.* Local augmented graph neural network for multi-omics cancer prognosis prediction and analysis. *Methods* 2023;**213**:1–9. https://doi.org/10.1016/j.ymeth.2023.02.011
14. Zhu J, Jung Hun O, Simhal AK. *et al.* Geometric graph neural networks on multi-omics data to predict cancer survival outcomes. *Comput Biol Med* 2023;**163**:107117. https://doi.org/10.1016/j.compbiomed.2023.107117
15. Kim SY. Gnn-surv: Discrete-time survival prediction using graph neural networks. *Bioengineering* 2023;**10**:1046. https://doi.org/10.3390/bioengineering10091046
16. Knottenbelt W, Gao Z, Wray R. *et al.* Coxkan: Kolmogorov-Arnold networks for interpretable, high-performance survival analysis. arXiv preprint arXiv:2409.04290, 2024.
17. Jiang L, Chao X, Bai Y. *et al.* Autosurv: Interpretable deep learning framework for cancer survival analysis incorporating clinical and multi-omics data. *npj Precis Oncol* 2024;**8**:4. https://doi.org/10.1038/s41698-023-00494-6
18. Longato E, Vettoretti M, Di Camillo B. A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models. *J Biomed Inform* 2020;**108**:103496. https://doi.org/10.1016/j.jbi.2020.103496
19. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. *Advances in neural information processing systems* 2017;**30**.
20. Zhao Y, Li M-C, Konaté MM. *et al.* Tpm, fpkm, or normalized counts? A comparative study of quantification measures for the analysis of rna-seq data from the nci patient-derived models repository. *J Transl Med* 2021;**19**:269. https://doi.org/10.1186/s12967-021-02936-w
21. Li Q, Han Z, Xiao-Ming W. Deeper insights into graph convolutional networks for semi-supervised learning. In: *Proceedings of the AAAI conference on artificial intelligence*, New Orleans, USA, Vol. **32**, 2018. 10.1609/aaai.v32i1.11604.
22. Kingma DP, Ba J. "Adam: A Method for Stochastic Optimization." In: Bengio Y, LeCun Y, (eds), *Proceedings of the 3rd International*

Conference on Learning Representations (ICLR) 2015, San Diego, CA, USA, May 7-9, 2015, arXiv. http://arxiv.org/abs/1412.6980.
23. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L. *et al.* PyTorch: An imperative style, high-performance deep learning library. *Adv Neural Inf Process Syst*. 2019;**32**.
24. Simon N, Friedman J, Hastie T. *et al.* Regularization paths for Cox's proportional hazards model via coordinate descent. *J Stat Softw* 2011;**39**:1–13. https://doi.org/10.18637/jss.v039.i05
25. Lee T-Y, Huang K-Y, Chuang C-H. *et al.* Incorporating deep learning and multi-omics autoencoding for analysis of lung adenocarcinoma prognostication. *Comput Biol Chem* 2020;**87**:107277. https://doi.org/10.1016/j.compbiolchem.2020.107277
26. Li S, Mulong D, Kaili X. *et al.* Genetic modulation of bet1l confers colorectal cancer susceptibility by reducing mirna binding and m6a modification. *Cancer Res* 2023;**83**:2142–54. https://doi.org/10.1158/0008-5472.CAN-22-0065
27. Ye M, Lin X, Feiyu L. *et al.* Hypoxia drives cbr4 down-regulation promotes gastroenteropancreatic neuroendocrine tumors via activation mammalian target of rapamycin mediated by fatty acid synthase. *J Cell Commun Signaling* 2024;**18**:e12041. https://doi.org/10.1002/ccs3.12041
28. Budnik N, Leroux A, Cooke M. *et al.* The role of s-adenosylhomocysteine hydrolase-like 1 in cancer. *Biochim Biophys Acta, Mol Cell Res* 2024;**1871**:119819. https://doi.org/10.1016/j.bbamcr.2024.119819
29. Umemura A, Itoh Y, Itoh K. *et al.* Association of gankyrin protein expression with early clinical stages and insulin-like growth factor-binding protein 5 expression in human hepatocellular carcinoma. *Hepatology* 2008;**47**:493–502. https://doi.org/10.1002/hep.22027
30. Zhou Y-J, Yang M-L, He X. *et al.* Rna-binding protein rps7 promotes hepatocellular carcinoma progression via loxl2-dependent activation of itgb1/fak/src signaling. *J Exp Clin Cancer Res* 2024;**43**:45. https://doi.org/10.1186/s13046-023-02929-1
31. Yanjiao O, Deng Y, Wang H. *et al.* Targeting antisense lncrna prkag2-as1, as a therapeutic target, suppresses malignant behaviors of hepatocellular carcinoma cells. *Front Med* 2021;**8**:649279. https://doi.org/10.3389/fmed.2021.649279
32. Chui Y-L, Ching AK-K, Chen S. *et al.* Bre over-expression promotes growth of hepatocellular carcinoma. *Biochem Biophys Res Commun* 2010;**391**:1522–5. https://doi.org/10.1016/j.bbrc.2009.12.111
33. Alsagaby SA. Transcriptomics-based validation of the relatedness of heterogeneous nuclear ribonucleoproteins to chronic lymphocytic leukemia as potential biomarkers of the disease aggressiveness. *Saudi Med J* 2019;**40**:328–38. https://doi.org/10.15537/smj.2019.4.23380
34. Liu Y, Zeyi W, Zhao Y. *et al.* Apolipoprotein h-based prognostic risk correlates with liver lipid metabolism disorder in patients with hbv-related hepatocellular carcinoma. *Heliyon* 2024;**10**:e31412. https://doi.org/10.1016/j.heliyon.2024.e31412
35. Yang L, Si H, Meng Ma Y. *et al.* Linc00221 silencing prevents the progression of hepatocellular carcinoma through let-7a-5p-targeted inhibition of mmp11. *Cancer Cell Int* 2021;**21**:1–15. https://doi.org/10.1186/s12935-021-01819-w
36. Yuqiang X, Ge K, Junhao L. *et al.* Microrna-493 suppresses hepatocellular carcinoma tumorigenesis through down-regulation of anthrax toxin receptor 1 (antxr1) and r-spondin 2 (rspo2). *Biomed Pharmacother* 2017;**93**:334–43. https://doi.org/10.1016/j.biopha.2017.06.047
37. You H, Wang Y, Wang X. *et al.* CD69+ Vδ1$\gamma\delta$ T cells are anti-tumor subpopulations in hepatocellular carcinoma. *Mol Immunol* 2024;**172**:76–84. https://doi.org/10.1016/j.molimm.2024.06.006
38. Han R, Feng P, Pang J. *et al.* A novel hcc prognosis predictor eef1e1 is related to immune infiltration and may be involved in eef1e1/atm/p53 signaling. *Front Oncol* 2021;**11**:700972. https://doi.org/10.3389/fonc.2021.700972
39. Zhang G, Zhang G. Upregulation of foxp4 in hcc promotes migration and invasion through regulation of emt. *Oncol Lett* 2019;**17**:3944–51. https://doi.org/10.3892/ol.2019.10049
40. Wang M, Lv G, Jiang C. *et al.* Mir-302a inhibits human hepg2 and smmc-7721 cells proliferation and promotes apoptosis by targeting map3k2 and pbx3. *Sci Rep* 2019;**9**:2032. https://doi.org/10.1038/s41598-018-38435-0
41. Wang J, Chen L, Li Y. *et al.* Overexpression of cathepsin z contributes to tumor metastasis by inducing epithelial-mesenchymal transition in hepatocellular carcinoma. *PloS One* 2011;**6**:e24967. https://doi.org/10.1371/journal.pone.0024967
42. Tongjun G, Fu AQ, Bolt MJ. *et al.* Clinical relevance of noncoding adenosine-to-inosine rna editing in multiple human cancers. *JCO clinical. Cancer Informatics* 2019;**3**:1–8. https://doi.org/10.1200/CCI.18.00151
43. Gao X, He K, Zeng Z. *et al.* Integrative analysis of the role of the dph gene family in hepatocellular carcinoma and expression validation. *Transl Cancer Res* 2024;**13**:4062–84. https://doi.org/10.21037/tcr-24-147
44. Lin Y-Y, Ming-Whei Y, Lin S-M. *et al.* Genome-wide association analysis identifies a glul haplotype for familial hepatitis b virus-related hepatocellular carcinoma. *Cancer* 2017;**123**:3966–76. https://doi.org/10.1002/cncr.30851
45. Xiang J, Chen GX-Q, Yin-Ying L. Isoc1 is a novel potential tumor suppressor in hepatocellular carcinoma. *Neoplasma* 2022;**69**:174–82. https://doi.org/10.4149/neo_2021_210815N1157

Briefings in Bioinformatics, 2025, **26(2)**, bbaf108
https://doi.org/10.1093/bib/bbaf108
Problem Solving Protocol