DOI: 10.14529/jsfi260207

Improving Performance of HPC Systems Based on Robust Survival Machine Learning Methods

Lev V. Utkin[1], ***Andrei V. Konstantinov***[1], ***Vladimir S. Zaborovsky***[1], ***Vladimir A. Muliukha***[1]

High-performance computing (HPC) systems face growing complexity and uncertainty that limit the accuracy of classical methods for predicting computational processes and task execution. To address the problem of tasks failing to complete within allotted time windows, termed "censored" events, survival machine learning methods are proposed for predicting execution time and minimizing such losses. Survival analysis plays an important role for solving the task of time-to-event prediction. While traditional methods handle censored data effectively, they often rely on strong parametric assumptions that may limit their flexibility. This paper introduces a novel survival analysis framework, called CiSurv (Contaminated imprecise Survival model), that integrates imprecise probability theory with attention-based multi-label classification. By reformulating survival prediction as an imprecise classification problem, CiSurv provides an approach to modeling uncertainty in censored observations while improving predictive accuracy. The proposed framework incorporates the imprecise contaminated model to refine interval-valued probabilities associated with censored data. Two models are developed: CiSurvN, which uses the neural network-based attention mechanism for solving the classification task, and CiSurvG, which employs Gaussian kernel-based attention with a single kernel parameter. Experiments on real and synthetic data demonstrate that CiSurvN generally outperforms CiSurvG due to its ability to learn complex feature dependencies. Key findings reveal that intermediate contamination parameter values yield optimal performance, outperforming a special case when the parameter is 1 in most cases. Codes implementing the proposed models are publicly available.

Keywords: high-performance computing, survival analysis, censoring, attention mechanism, imprecise contaminated model.

Introduction

High-performance computing (HPC) is a powerful tool used for a wide range of scientific research requiring large volumes of data and high levels of algorithmic complexity. Although the peak performance of HPC systems continues to grow steadily, reflecting the potential of hybrid computing electronic component, the complexity of using such systems is simultaneously increasing. Furthermore, the high uncertainty in describing the characteristics of computational environment and multifunction application workflows limits the accuracy of classical methods for planning and predicting both stochastic and deterministic characteristics of computational processes. One of the important problems of a functioning supercomputer system is to successfully complete the execution of a specific applied task executed on behalf of a specific user in the time interval allotted for this by the dispatcher and to minimize losses arising when the task does not complete the execution due to various reasons. The actual execution time of tasks depends on many factors and always differs from the estimated time. When the task is not completed, then the corresponding event can be regarded as the censored event which means that the task cannot be executed in a predefined period of time. To predict the execution time of the task and to minimize the number of "censored events, the survival machine learning methods can be applied.

[1]Higher School of Artificial Intelligence Technologies, Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation

Survival analysis, commonly known as time-to-event analysis, is a core statistical methodology used to model the time until a specific event occurs [10]. Its broad applicability has made it indispensable in such fields as medicine, engineering, finance, and industrial reliability [1]. A major strength of this methodology is its capacity to incorporate censored data, where event times are incompletely observed, enabling more accurate and interpretable results for evidence-driven decisions. Survival analysis supports engineering tasks such as predicting system failures, optimizing maintenance schedules, and evaluating reliability [18].

Survival analysis has spurred the development of numerous machine learning models to address time-to-event prediction challenges [23, 27]. These models include adaptations of traditional machine learning methods, for example, random forests, neural networks, transformer-based approaches, to survival data [4, 12, 28]. Methodological perspectives of survival analysis are offered in [8].

Attention-based approaches [20] have enhanced survival analysis. Transformers capture long-range dependencies [11, 24], with applications spanning medical survival analysis [13] and competing risks scenarios [25]. These methods demonstrate improved performance in high-dimensional settings while providing feature importance insights.

Alternatively, survival analysis can be reframed as classification by discretizing time intervals [7, 19] with likelihood-based loss functions [15]. The main idea of the models is to consider the event probabilities over intervals. Another approach [14] for survival analysis with discretizing time intervals is to consider non-parametric probability distributions over time intervals for each instance in the training set. For uncensored observations, the event time distribution is deterministic, assigning probability 1 to the true interval and 0 elsewhere. For censored observations, uncertainty arises: the event could occur in any interval after censoring, yielding a set of possible distributions and reflecting partial knowledge. It is assumed that the probability of the event occurring in any interval after the censored time can range from zero to one, reflecting prior ignorance. To aggregate the probability distributions generated for each instance from the training set, it is proposed in [14] to employ kernel Nadaraya-Watson regression [17, 26] with trainable dot-product attention weights [16, 20]. A model implementing the approach is called iSurvJ.

We propose to extend the approach introduced in [14] by applying the imprecise ϵ-contaminated model [21] to reduce the set of probability distributions over the time to event intervals. A goal of incorporating the imprecise ϵ-contaminated model into the model [14] is to enhance the prediction accuracy of the model by searching for the optimal hyperparameter ϵ which controls imprecision of probabilities over the time intervals. The proposed model, called CiSurv (**C**ontaminated **i**mprecise **Surv**ival model), aims to open a door for incorporating various imprecise models into survival analysis. It illustrates that there are optimal parameters of the imprecise models which can improve a survival model analyzed. Two modifications of CiSurv are studied: CiSurvN and CiSurvG. The first model, CiSurvN, uses the neural network to compute the matrix of attention weights. The second model, CiSurvG, uses the Gaussian kernels with a single parameter for computing the weights. On the one hand, the model CiSurvN is more powerful due to using the neural network. On the other hand, when a dataset is rather small, the model CiSurvG may provide better results due to a small number of training parameters.

It should be noted that imprecise models have been proposed in survival analysis offering robust alternatives when distributional assumptions are violated. Key approaches include the Hill model framework for right-censored data [6] and the imprecise Dirichlet model [5, 22].

However, the models provide lower and upper bounds for the survival concepts, for example, for survival functions. Moreover, in contrast to the proposed approach, they do not take into account relationships between the feature vectors.

Various numerical experiments with real and synthetic datasets are provided to compare the proposed survival models with each other for different values of the hyperparameter ϵ. It is shown by numerical examples that there is an "optimal value" of ϵ which provides the best accuracy measure for a certain dataset.

The corresponding codes implementing the proposed models are publicly available at: `https://github.com/NTAILab/isurv_reducing_imprecision`.

The paper is organized as follows. Section 1 provides basic definitions of survival analysis. The model iSurvJ proposed in [14] is considered in Section 2. The approach to reduce imprecision of probabilities in iSurvJ and the model CiSurv are described in Section 3. Numerical experiments with well-known public real and synthetic data illustrating CiSurv are provided in Section 4. Concluding remarks are provided in Section 5.

1. Preliminaries: Survival Analysis

Datasets in survival analysis are represented by a set of triplets $\mathcal{A} = \{(\mathbf{x}_1, \delta_1, T_1), ..., (\mathbf{x}_N, \delta_N, T_N)\}$ [10]. Here the vector $\mathbf{x}_i \in \mathbb{R}^d$ consisting of d features characterizes the i-th object. The time T_i corresponds to one of two types of the event observations. The first type is when the event is observed. In this case, the observation is called uncensored and the censoring indicator δ_i is equal to 1. The second type is when the event is not observed, it is greater than T_i, but its 'true' value is unknown. In this case, the observation is called censored and the censoring indicator δ_i is equal to 0. A survival model is trained on the set $\mathcal{A}$ to predict probabilistic measures of an event time T for a new object represented by the vector $\mathbf{x}$.

An important concept in survival analysis is the survival function (SF) $S(t \mid \mathbf{x})$, which is defined as the probability of surviving up to time t, i.e., $S(t \mid \mathbf{x}) = \Pr\{T > t \mid \mathbf{x}\}$. Most survival measures can be derived through the SF. Comparison of survival models is often carried out by means of the Brier Score (BS) [3, 9] which is defined as

$$BS(t, \hat{S}) = \mathbb{E}\left[\left(\Delta_{new}(t) - \hat{S}(t \mid \mathbf{x}_{new})\right)^2\right], \tag{1}$$

where $\Delta_{new}(t) = \mathbf{1}\left(T_{new} > t\right)$ is the true status of a new test subject; $\hat{S}(t \mid \mathbf{x}_{new})$ is the predicted survival probability.

The Integrated Brier Score (IBS) extends the BS by averaging it over a range of time points, typically from $t = 0$ to a maximum time $t_{\max}$. It is defined as:

$$IBS = \frac{1}{t_{\max}} \int_0^{t_{\max}} BS(t, \hat{S})\, \mathrm{d}t. \tag{2}$$

2. Survival Analysis as an Imprecise Multi-Label Classification Problem

First, we introduce the model iSurvJ proposed in [14] which considers the survival analysis problem as an imprecise classification task under condition of total ignorance.

Consider a partition of the time axis into discrete intervals:

$$[0,\infty) = [0,t_1] \cup [t_1,t_2] \cup ... \cup [t_{T-2},t_{T-1}] \cup [t_{T-1},\infty), \tag{3}$$

such that

$$0 = t_0 < t_1 < t_2 < ... < t_{T-1} < \infty. \tag{4}$$

The partition divides the time axis into T intervals denoted as $\tau_i = [t_{i-1}, t_i]$ for $i = 1, \ldots, T$. Let us define the probability $\pi_k^{(i)}$ that an event corresponding to the i-th object is observed in the interval τ_k. This generates a probability distribution for each subject $\pi^{(i)} = (\pi_1^{(i)}, ..., \pi_T^{(i)})$, $i = 1, ..., N$. If the observation is uncensored, then there holds

$$\pi_j^{(i)} = 0, \text{ if } j \neq k, \quad \pi_j^{(i)} = 1, \text{ if } j = k. \tag{5}$$

For censored observations, the true event time must occur in some interval after t_{k-1}, though the specific interval remains unknown. Consequently, the probability that the event is at each interval after the time t_{k-1} can be from 0 to 1, i.e., there holds

$$\pi_j^{(i)} = 0, \text{ if } j < k, \quad \pi_j^{(i)} \in [0,1], \text{ if } j \geq k. \tag{6}$$

The interval $[0, 1]$ is a form of prior ignorance. We have the constraint $\sum_{j=1}^{T} \pi_j^{(i)} = 1$ for precise probabilities. However, this condition is not fulfilled when the upper or lower probabilities are summed. Due to imprecision of probabilities $\pi_j^{(i)}$, we suppose that the i-th instance produces a set of probability distributions $\pi^{(i)}$ denoted as $\mathcal{R}^{(i)}$.

Let us consider the following *classification problem*. There is a set of feature vectors $\mathbf{x}_1, ..., \mathbf{x}_N$. The k-th interval $[t_{k-1}, t_k]$ can be viewed as the k-th class represented by the vector $\pi_k = (\pi_k^{(1)}, ..., \pi_k^{(N)})^{\mathrm{T}}$. Then the Nadaraya-Watson kernel regression [17, 26] can be applied to find the class probability distribution $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), ..., p_T(\mathbf{x}))$ of a new instance $\mathbf{x}$. Since a part of probabilities $\pi_k^{(i)}$ are interval values, then it is obvious that the predicted class probabilities $p_1, ..., p_T$ are also interval-valued. They are determined as follows:

$$p_k(\mathbf{x}) = \sum_{i=1}^{N} w(\mathbf{x}, \mathbf{x}_i)\pi_k^{(i)}, \ k = 1, ..., T, \tag{7}$$

where $w(\mathbf{x}, \mathbf{x}_i)$ is an attention weight expressed through kernels $K(\mathbf{x}, \mathbf{x}_i)$, $i = 1, ..., N$.

The attention weights satisfy the condition:

$$\sum_{i=1}^{N} w(\mathbf{x}, \mathbf{x}_i) = 1. \tag{8}$$

We extend the above weights $w(\mathbf{x}, \mathbf{x}_i)$ to a more general form accepted in transformers [16, 20]. Introduce a matrix of the input keys $\mathbf{K} \in \mathbb{R}^{N \times d}$ (the matrix of N vectors $\mathbf{x}_i$), and a matrix of the input queries $\mathbf{Q} \in \mathbb{R}^{N \times d}$ (the matrix of the same feature vectors). We also introduce the trainable matrices $\mathbf{W}_K \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_Q \in \mathbb{R}^{d \times d}$. Then the matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ of attention weights $w(\mathbf{x}_j, \mathbf{x}_i)$, $i = 1, ..., N$, $j = 1, ..., N$, can be computed as [16, 20]:

$$\mathbf{W} = \mathrm{softmax}\left(\frac{(\mathbf{Q}\mathbf{W}_Q)\,(\mathbf{K}\mathbf{W}_K)}{\sqrt{d}}\right). \tag{9}$$

Hence, (7) is rewritten as:

$$p_k(\mathbf{x}) = \mathbf{W}_k \pi_k, \ k = 1, ..., T, \tag{10}$$

where $\mathbf{W}_k$ is the k-th row of the matrix $\mathbf{W}$.

By having the probability distribution $\mathbf{p}(\mathbf{x})$, the predicted SF at time $t \in [t_{j-1}, t_j]$, $j = 1, ..., T$, can be computed as follows:

$$\hat{S}(t \mid \mathbf{x}) = \sum_{k=j+1}^{T} p_k(\mathbf{x}). \tag{11}$$

The matrix $\mathbf{W}$ can be computed in a more simple way by using the Gaussian kernel

$$w(\mathbf{x}_j, \mathbf{x}_i) = \text{softmax}\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\psi}\right), \tag{12}$$

where ψ is the kernel parameter.

On the one hand, this way simplifies the model because it learns only the parameter ψ for computing the matrix $\mathbf{W}$ and can be applied to cases of small datasets. On the other hand, this simplified model may provide worse accuracy results.

2.1. Model iSurvJ

To solve the above classification problem under condition of the interval-valued class probabilities, we have to define how to train the classifier when the class probabilities are interval-valued.

In the model, *iSurvJ*, the class probabilities and the attention weights are jointly learned. To implement the learning, the probability logits are initially initialized as zeros. At each epoch, the softmax(logits) operation is applied to the logits to obtain probabilities of intervals $\widetilde{\pi}_k^{(i)}$ and ensuring the condition $\sum_{k=1}^{T} \widetilde{\pi}_k^{(i)} = 1$. Here $\widetilde{\pi}_k^{(i)}$ are precise analogues of interval-valued probabilities $\pi_k^{(i)}$ obtained by means of learning.

Let $c(i)$ be an index of a time interval in which the event, corresponding to the i-th instance, occurs, and $p_{c(i)}(\mathbf{x}_i)$ be the probability of the interval with the index $c(i)$. The whole loss function for learning the model is defined as

$$\mathcal{L}(\mathbf{p}) = \sum_{i=1}^{N} l(\mathbf{p}(\mathbf{x}_i)), \tag{13}$$

where $l(\mathbf{p}(\mathbf{x}_i))$ is determined as

$$l(\mathbf{p}(\mathbf{x}_i)) = -\mathbf{1}[\delta_i = 1] \cdot \log\left(p_{c(i)}(\mathbf{x}_i)\right) - \mathbf{1}[\delta_i = 0] \cdot \log\left(\sum_{j=c(i)+1}^{T} p_j(\mathbf{x}_i)\right)$$
$$- \gamma \sum_{k=1}^{T} \widetilde{\pi}_k^{(i)} \cdot \log\left(\widetilde{\pi}_k^{(i)}\right). \tag{14}$$

For the i-th instance, $\mathbf{p}(\mathbf{x}_i) = (p_1(\mathbf{x}_i), ..., p_T(\mathbf{x}_i))$ represents a precise probability distribution over time intervals, computed at each epoch using $\mathbf{W}$ and $\widetilde{\pi}_k^{(i)}$ via equation (10). The hyper-parameter γ controls the influence of the regularization term in the loss function (14), which is

defined as the entropy of the probability distribution $\widetilde{\pi}^{(i)}$. This regularization prevents iSurvJ from overconfident decision-making during the joint optimization of weights and probabilities.

An implementation of training the model iSurvJ is presented as Algorithm 1.

Algorithm 1 The training algorithm for iSurvJ [14]

Require: Training set $\mathcal{A}$; hyperparameters p_{mask}, T, d; the number of epochs E
Ensure: Weight matrix $\mathbf{W}$; probability distribution $\mathbf{p}(\mathbf{x}_i)$
1: Initialize randomly $\mathbf{W}_Q$, $\mathbf{W}_K$ and compute $\mathbf{W}$
2: Initialize logits of $\widetilde{\pi}_k$, $k = 1, ..., T$, as zeros
3: **while** Number of epochs $\leq E$ **do**
4: Compute $\widetilde{\pi}_k$, $k = 1, ..., T$, using the softmax applied to logits y
5: Compute the distribution $\mathbf{p}(\mathbf{x}_i)$ using $\mathbf{W}$ and $\widetilde{\pi}_k$, $k = 1, ..., T$
6: Learn $\mathbf{W}$ and $\widetilde{\pi}_k$, $k = 1, ..., T$ using (14) and (13)
7: Compute logits of $\widetilde{\pi}_k$, $k = 1, ..., T$
8: **end while**

It is important to point out that the constraints for some probabilities in the form of intervals $[0, 1]$ are automatically fulfilled due to the use of the softmax operation. It always produces a probability distribution.

3. Reducing Imprecision of Probabilities

3.1. Precise Estimates of Interval Probabilities

One of the problems of using the above approach is that the probability distributions $\pi^{(i)}$ are too imprecise if the i-th observation is censored. In order to reduce the imprecision, it is proposed to apply the imprecise ϵ-contaminated model. However, if we do not exploit any additional information about the i-th object behavior, it is impossible to reduce the probability intervals $[0, 1]$ for $\pi_j^{(i)}$. One of the ways to overcome this difficulty is to use a statistical model constructed from other observations available in the dataset $\mathcal{A}$. It should be noted that the probability distributions $\pi^{(i)}$ corresponds to the i-th object and does not take into account other objects because they have different feature vectors. There are several approaches to take into account the difference of the feature vectors. All the approaches are based on a common idea.

Let us consider the probabilities $\pi_j^{(i)}$ for $j \geq k$ when the corresponding event will occur after the interval τ_k (an uncensored observation). Since the observation is censored at t_k, we know that the random time to event X is satisfied to the condition $X > t_k$. Thus, the conditional probability that the event will occur in the interval $\tau_j = [t_{j-1}, t_j]$ for $j > k$ is of the form:

$$\pi_j^{(i)} = P(t_{j-1} \leq X_i \leq t_j \mid X_i > t_k) = \frac{\widetilde{S}(t_{j-1} \mid \mathbf{x}_i) - \widetilde{S}(t_j \mid \mathbf{x}_i)}{\widetilde{S}(t_k \mid \mathbf{x}_i)}, \tag{15}$$

where $\widetilde{S}(t \mid \mathbf{x}_i)$ is an estimate of the conditional SF for the i-th object.

It is important to note that $\pi_j^{(i)} = 0$ for all $j < k$ and there holds:

$$\sum_{j=1}^{T} \pi_j^{(i)} = 1. \tag{16}$$

SFs $\widetilde{S}(t_j \mid \mathbf{x}_s)$ can be obtained in several ways. One of the ways taking into account relationships between feature vectors is the Beran estimator [2] which estimates the SF as follows:

$$\widetilde{S}(t \mid \mathbf{x}) = \prod_{t_i \le t} \left\{ 1 - \frac{\alpha(\mathbf{x}, \mathbf{x}_i)}{1 - \sum_{j=1}^{i-1} \alpha(\mathbf{x}, \mathbf{x}_j)} \right\}^{\delta_i}, \tag{17}$$

the weight $\alpha(\mathbf{x}, \mathbf{x}_i)$ corresponds to location of $\mathbf{x}_i$ relative to $\mathbf{x}$ under condition that the closer $\mathbf{x}_i$ to $\mathbf{x}$, the greater the weight $\alpha(\mathbf{x}, \mathbf{x}_i)$.

Weights are defined through kernels as

$$\alpha(\mathbf{x}, \mathbf{x}_i) = \frac{K(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}_j)}. \tag{18}$$

If $K(\mathbf{x}, \mathbf{x}_i)$ is the Gaussian kernel with a parameter ω, then weights are of the form:

$$\alpha(\mathbf{x}, \mathbf{x}_i) = \text{softmax}\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\omega} \right). \tag{19}$$

It should be noted that the set $\mathcal{A}$ can be restricted to the M nearest neighbors of $\mathbf{x}$ to simplify computations. In this case, SFs $\widetilde{S}(t_j \mid \mathbf{x}_i)$ can be efficiently computed by fitting a Kaplan-Meier model to these M neighbors. This implementation corresponds precisely to the Beran estimator with uniform weighting.

3.2. Imprecise Statistical Models

3.2.1. Imprecise contaminated model

The linear-vacuous mixture, or imprecise ϵ-contaminated model [21], produces a set $\mathcal{P}(\epsilon, \pi)$ of the probability distributions $\pi^* = (\pi_1^*, ..., \pi_T^*)$ such that $\pi_j^* = (1 - \epsilon)\pi_j + \varepsilon \cdot h_i$, where $\pi = (\pi_1, ..., \pi_T)$ is an elicited probability distribution, $\mathbf{h} = (h_1, ..., h_T)$ is an arbitrary probability distribution, and $\epsilon \in [0, 1]$ is a model contamination parameter which reflects imprecision of the model. The set $\mathcal{P}(\epsilon, \pi)$ is a subset of the unit simplex with the dimension d. Moreover, it coincides with the unit simplex when $\epsilon = 1$. The ϵ-contaminated model for the i-th object can be represented as

$$\pi^{*(i)} = (1 - \epsilon) \cdot \pi^{(i)} + \epsilon \cdot \mathbf{h}. \tag{20}$$

Probabilities $\pi_j^{(i)}$, $j = k, ..., T$, are defined through (15). Using the imprecise ϵ-contaminated model and the definition of $\pi_j^{(i)}$ for uncensored observations given in (15), we can reduce the interval $[0, 1]$ in (6) and control the obtained interval by means of the contamination parameter ϵ. The lower probability of $\pi_j^{(i)}$ is determined under condition $h_j = 0$ as follows:

$$\underline{\pi}_j^{*(i)} = (1 - \epsilon) \cdot \pi_j^{(i)}. \tag{21}$$

The upper probability of $\pi_j^{(i)}$ is determined under condition $h_j = 1$ as follows:

$$\overline{\pi}_j^{*(i)} = (1 - \epsilon) \cdot \pi_j^{(i)} + \epsilon. \tag{22}$$

The lower and upper predicted class probabilities $p_1, ..., p_T$ from (7) are determined as

$$\underline{p}_k(\mathbf{x}_0) = \sum_{i=1}^{N} w_{0,i} \underline{\pi}_k^{(i)}, \ \overline{p}_k(\mathbf{x}_0) = \sum_{i=1}^{N} w_{0,i} \overline{\pi}_k^{(i)} \ k = 1, ..., T. \tag{23}$$

However, according to Algorithm 1, we do not need to compute the lower and upper class probabilities. Algorithm 1 could be applied to compute the SF for a new subject without these bounds. However, it assumes that elements of the probability distributions are restricted by the interval $[0, 1]$ to simply use of the softmax operation. An advantage of the imprecise ϵ-contaminated model is that the probability distribution $\mathbf{h}$ in 20 belongs to the unit simplex. Hence, the i-th elements $\widetilde{\pi}_k^{(i)}$ of learned vectors $\widetilde{\pi}_k$, $k = 1, ..., T$, in Step 4 of Algorithm 1 are replaced with

$$\pi_k^{*(i)} = (1 - \epsilon) \cdot \pi_k^{(i)} + \epsilon \cdot \widetilde{\pi}_k^{(i)}. \tag{24}$$

Therefore, Algorithm 1 is extended by additional calculations of probabilities $\pi_k^{*(i)}$ through learned probabilities $\widetilde{\pi}_k^{(i)}$ and probabilities $\pi_k^{(i)}$ obtained by using (15) and the Beran estimator (17).

4. Numerical Experiments

The main goal of numerical experiments is to show that the proposed model can provide better results for certain values of the contamination parameter ϵ in comparison with the model iSurvJ which can be regarded as a special case of CiSurv under condition $\epsilon = 1$. The model iSurvJ has been compared with different survival models and demonstrated outperforming results [14]. Therefore, we show that the contamination parameter ϵ impacts on the model CiSurv performance.

4.1. Conditions of Experiments

For research purposes, we consider two modifications of the proposed model: CiSurvN and CiSurvG. The first model, CiSurvN, uses the neural network to compute the matrix $\mathbf{W}$. The model CiSurvG uses the Gaussian kernels with a single parameter for computing $\mathbf{W}$. On the one hand, the model CiSurvN is more powerful due to using the neural network. On the other hand, when a dataset is rather small, the model CiSurvG may provide better results due to a small number of training parameters.

20 independent experiments are conducted for models CiSurvN and CiSurvG. In each experiment, the dataset is diveded into training and test sets in a 60% to 40% ratio. The well-established metric IBS is used for comparison. The resulting metric values are averaged over 20 runs.

In all experiments, the following values of hyperparameters are used for both models: number of training iterations: 300; learning rate: 10^{-2}; regularization: $\alpha = 2 \cdot 10^{-3}$; entropy regularization coefficient: $\gamma = 2.0$; kernel parameter (temperature) in the Beran estimator: 0.4; batch size: 100% of the training set.

Prior to model training, all numerical features are standardized using z-score normalization, while categorical features are encoded using one-hot encoding to ensure compatibility with the machine learning algorithms.

4.2. Results of Experiments: Real Datasets

To evaluate the performance of the proposed model, we study several publicly available survival datasets: Veterans, Breast cancer, LND, GCD. These datasets from medical observations are used in experiments instead of real data with times to the successful execution of tasks in

Table 1. IBS obtained for real datasets by using CiSurvN for different ϵ

ϵ	Veterans	Breast cancer	LND	GCD
0.0	0.0993	0.2414	0.2574	0.1984
0.1	0.0990	0.2268	0.2491	0.1936
0.2	0.0982	0.2101	0.2290	0.1907
0.3	**0.0975**	0.2010	0.2283	0.1864
0.4	0.0980	**0.1924**	0.2273	0.1855
0.5	0.0993	0.2015	0.2175	0.1844
0.6	0.0993	0.2022	**0.2120**	0.1843
0.7	0.0993	0.2053	0.2234	0.1820
0.8	0.0993	0.2153	0.2288	**0.1785**
0.9	0.0994	0.2160	0.2300	0.1845
1.0	0.0995	0.2166	0.2303	0.1867

HPC systems due to two reasons. First, data on the success or failure of tasks on HPC systems is generally unique and typically inaccessible to many users and readers. This prevents the requirement for reproducibility of results, as the model parameters are tuned to specific data. Second, real datasets from observations of task solving on HPC systems are quite large and have complex structures, which significantly complicate or make impossible the study of features of the proposed models. Therefore, we test and investigate CiSurvN and CiSurvG on the open datasets having restricted sizes.

The *Veterans' Administration Lung Cancer Study (Veteran) Dataset* includes data on 137 instances characterized by 6 features. It is available through the R package "survival".

The *Breast Cancer Dataset* consists of 198 samples described by 80 features. It can be obtained from the Python library "scikit-survival".

The *Lupus Nephritis Dataset (LND)* consists of 87 observations. The dataset is available at `http://www.stat.rice.edu/~sneeley/STAT553/Datasets`.

The *Gastric Cancer Dataset (GCD)* includes 90 observations with 4 features. It is available through the R package "coxphw".

Table 1 shows IBS obtained for real datasets by using the CiSurvN modification for different values of ϵ from 0 to 1. The best metric values are highlighted in bold. It can be seen from Tab. 1 that there exist "optimal" values of ϵ different from 1, which correspond to the smallest values of IBS. This implies that incorporation of the imprecise ϵ-contaminated model into CiSurvN may lead to outperforming results.

Figure 1 illustrates how IBS depends on the parameter ϵ for some datasets when the CiSurvN modification of CiSurv is used.

Table 2 shows the IBS metrics obtained for real datasets by using the CiSurvG modification of CiSurv for different values of ϵ from 0 to 1. It is interesting to point out that the results differ from the results obtained by using CiSurvG and presented in Tab. 1. First of all, one can see from Tab. 2 that CiSurvG provides worse results in comparison with CiSurvN because only one training parameter of the attention weight is used (the parameter of the Gaussian kernel). Second, we can see that some datasets show the best IBS in "extreme" cases when the contamination parameter ϵ is 1 or close to 1. Figure 2, where the dependence of IBS on the parameter ϵ is shown for real datasets using CiSurvG, illustrates this peculiarity of results.

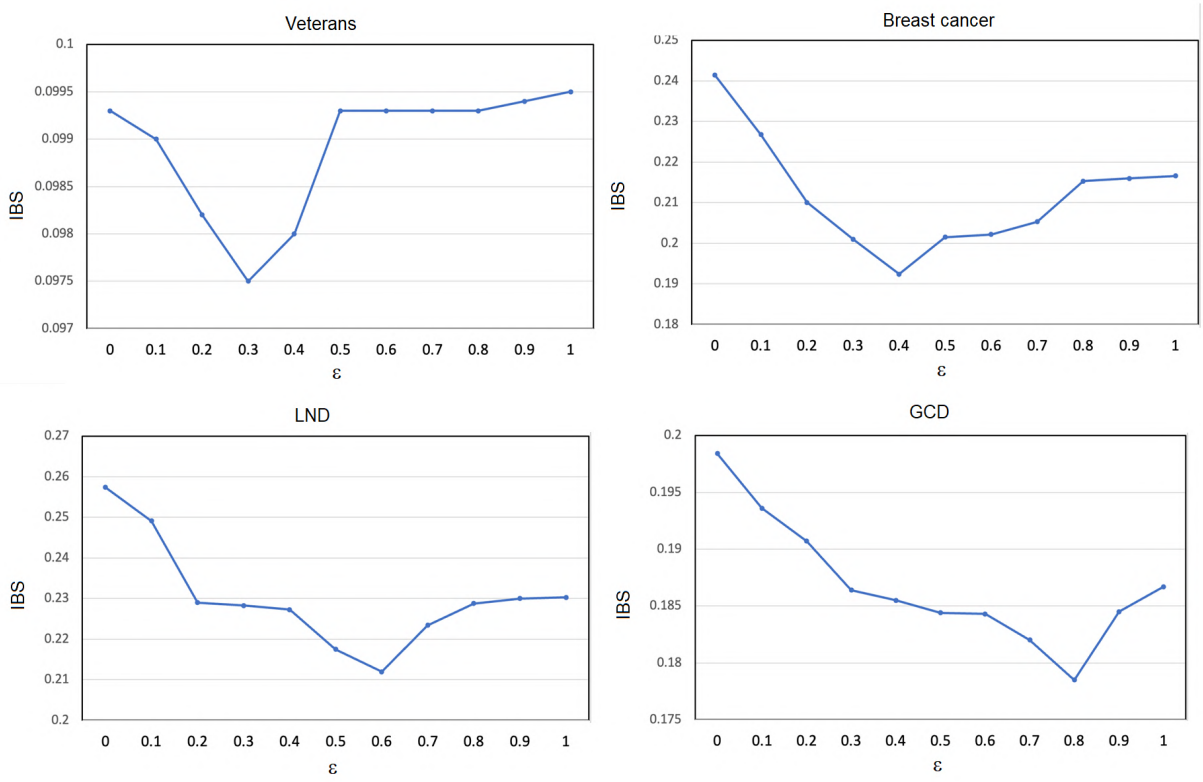

Figure 1. IBS metrics as functions of the contamination parameter ϵ for some real datasets obtained by using CiSurvN

Table 2. IBS obtained for real datasets by using CiSurvG for different ϵ

ϵ	Veterans	Breast cancer	LND	GCD
0.0	0.1585	0.3378	0.2867	0.2089
0.1	0.1571	0.3285	0.2802	0.2037
0.2	0.1560	0.3216	0.2750	0.2004
0.3	0.1550	0.3167	**0.2706**	0.1980
0.4	0.1542	0.3136	**0.2690**	0.1964
0.5	0.1536	**0.3124**	0.2698	0.1952
0.6	0.1531	0.3128	0.2706	0.1944
0.7	0.1527	0.3152	0.2714	0.1932
0.8	0.1524	0.3193	0.2723	0.1926
0.9	0.1522	0.3245	0.2730	**0.1925**
1.0	**0.1521**	0.3297	0.2734	0.1928

4.3. Synthetic Datasets

To rigorously benchmark the capabilities of the proposed survival analysis models under diverse and adversarially designed conditions, we construct synthetic datasets Linear, Quadratic, Friedman 1, Noisy. These datasets incorporate structured complexity, including non-linear de-

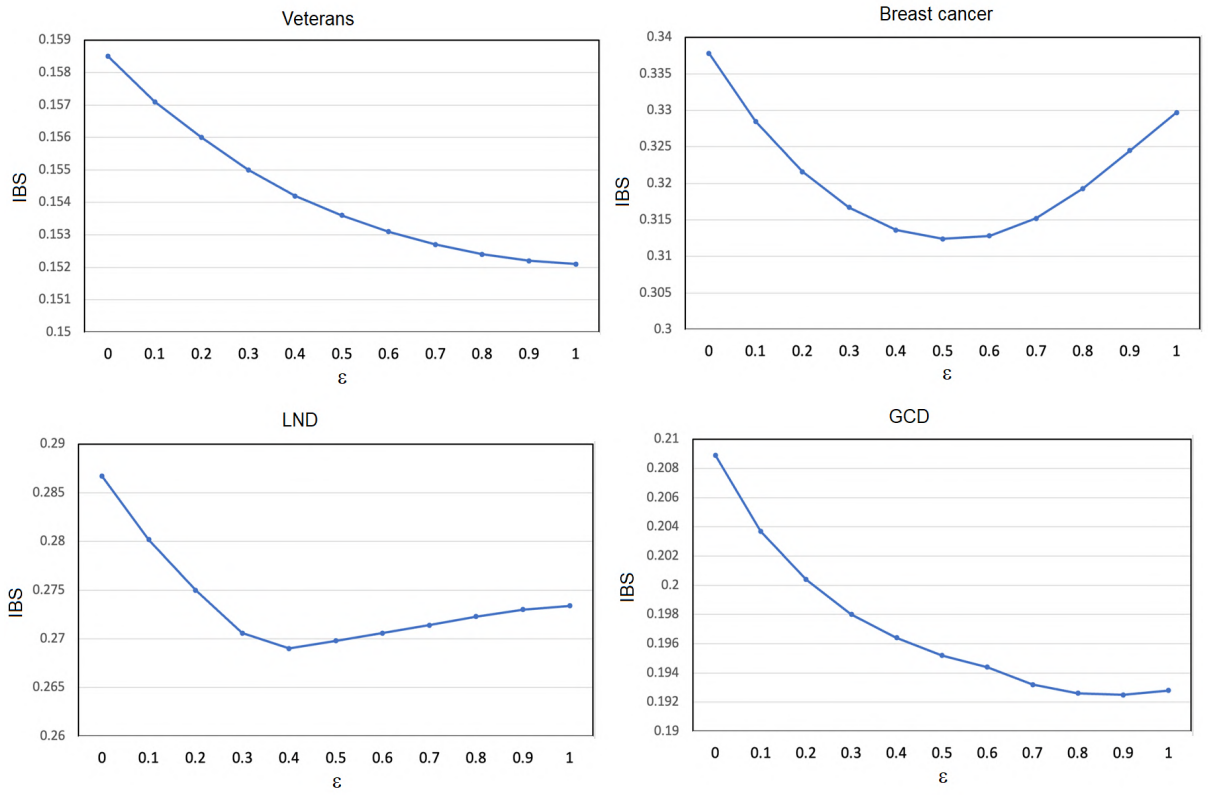

Figure 2. IBS metrics as functions of the contamination parameter ϵ for some real datasets obtained by using CiSurvG

pendencies, feature interactions, and noise regimes. Let $\mathbf{x} = (x_1, x_2, \ldots, x_d)^\top \in \mathbb{R}^d$ be a feature vector.

For all synthetic datasets, censoring indicators $\delta_i \in \{0, 1\}$ are generated as independent Bernoulli random variables with fixed probabilities $\Pr\{\delta_i = 0\}$ (censored) and $\Pr\{\delta_i = 1\}$ (observed). Event times T are sampled from a Weibull distribution with shape parameter $k > 0$ and are of the form:

$$T = \frac{y}{\Gamma\left(1 + \frac{1}{k}\right)} \cdot (-\log(u))^{\frac{1}{k}}, \tag{25}$$

where $u \in \mathcal{U}(0, 1)$ is a random number uniformly distributed in the interval $[0, 1]$; y is a function of features, which is defined for each dataset and shown below; $\Gamma(\cdot)$ is the gamma function.

The *Friedman1* dataset defines y as:

$$y = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5, \tag{26}$$

where $x_i \sim \mathcal{U}(0, 1)$ for $i \leq 5$, the remaining $d - 5$ features (if $d > 5$) are considered as noise variables and do not influence y.

The *Noisy* dataset generates T by applying the function $y = X\mathbf{w}$ and $k \leq 1$. The parameter k leads to high-variance noise. Here $X \in \mathbb{R}^{n \times d}$ is a feature matrix with elements $x_{ij} \sim \mathcal{U}(0, 1)$; $\mathbf{w} \in \mathbb{R}^d$ is a coefficient vector whose elements are generated as: $w_j \sim \mathcal{U}(0, 1)$.

Table 3. IBS obtained for synthetic datasets by using CiSurvN for different ϵ

ϵ	Linear	Quadratic	Friedman1	Noisy
0.0	0.2021	0.2892	0.1336	0.1208
0.1	0.1924	0.2783	0.1310	0.1175
0.2	0.1870	0.2741	0.1240	0.1161
0.3	0.1892	0.2601	0.1202	0.1162
0.4	0.1848	0.2622	0.1160	0.1152
0.5	0.1824	**0.2538**	0.1142	0.1120
0.6	0.1839	0.2561	**0.1139**	**0.1107**
0.7	0.1826	0.2541	0.1144	0.1103
0.8	0.1811	0.2580	0.1158	0.1136
0.9	**0.1780**	0.2569	0.1164	0.1177
1.0	0.1825	0.2574	0.1175	0.1190

The *Linear* dataset uses the linear function:

$$y = \sum_{i=1}^{d} w_i x_i = \mathbf{x}^\top \mathbf{w}, \tag{27}$$

where $x_i \sim \mathcal{U}(0,1)$; $\mathbf{w} = (w_1, w_2, \ldots, w_d)^\top \in \mathbb{R}^d$ is a vector of weights, $w_i \sim \mathcal{U}(0,1)$.

The *Quadratic* dataset implements the quadratic dependency between features such that $y = \mathbf{x}^\top Q \mathbf{x}$, where $x_i \sim \mathcal{U}(0,1)$, $Q \in \mathbb{R}^{d\times d}$ is a symmetric positive semi-definite matrix defined as $Q = A^\top A$, $A \in \mathbb{R}^{d\times d}$ is a matrix with normally distributed elements $a_{ij} \sim \mathcal{N}(0,1)$.

In each experiment, a new training set of 500 samples with 5 features and the censoring rate 0.2 is generated. The test set is the same across all experiments and consisted of 300 samples with the same censoring rate. In all other aspects, the experimental procedure is identical to that used for the real datasets.

Numerical results provided by CiSurvN with synthetic data are given in Tab. 3. The results are depicted in Fig. 3. One can see that from the results that there exist "optimal" values of ϵ between 0 and 1, corresponding to the smallest values of IBS. Numerical results provided by the model CiSurvG with synthetic data are given in Tab. 4. The same results are depicted in Fig. 4. It can be seen from Tab. 4 and Fig. 4 that, for half of the datasets, experimental results obtained using the CiSurvG model show that the optimal value of ϵ is 1. This indicates that with a small number of trainable parameters, the model attempts to relax constraints on the probability distribution set in the ϵ-contaminated model to allow greater freedom for the parameter of the Gaussian kernel.

Experiments conducted on real and synthetic datasets demonstrate the effectiveness of the proposed CiSurv model and its two modifications, CiSurvN and CiSurvG. For the CiSurvN model, which utilizes a neural network to compute the weight matrix $\mathbf{W}$, experiments showed that values of the parameter ϵ (typically between 0.3 and 0.7) yielded the best performance across most datasets. This confirms a hypothesis that the ϵ-contaminated model can indeed improve upon the special case of iSurvJ for which $\epsilon = 1$.

The improvement in the IBS scores across multiple datasets when using optimal values of ϵ provides strong evidence for the value of the contamination model framework in survival analysis. The numerical results demonstrate that the proposed CiSurv framework, particularly the

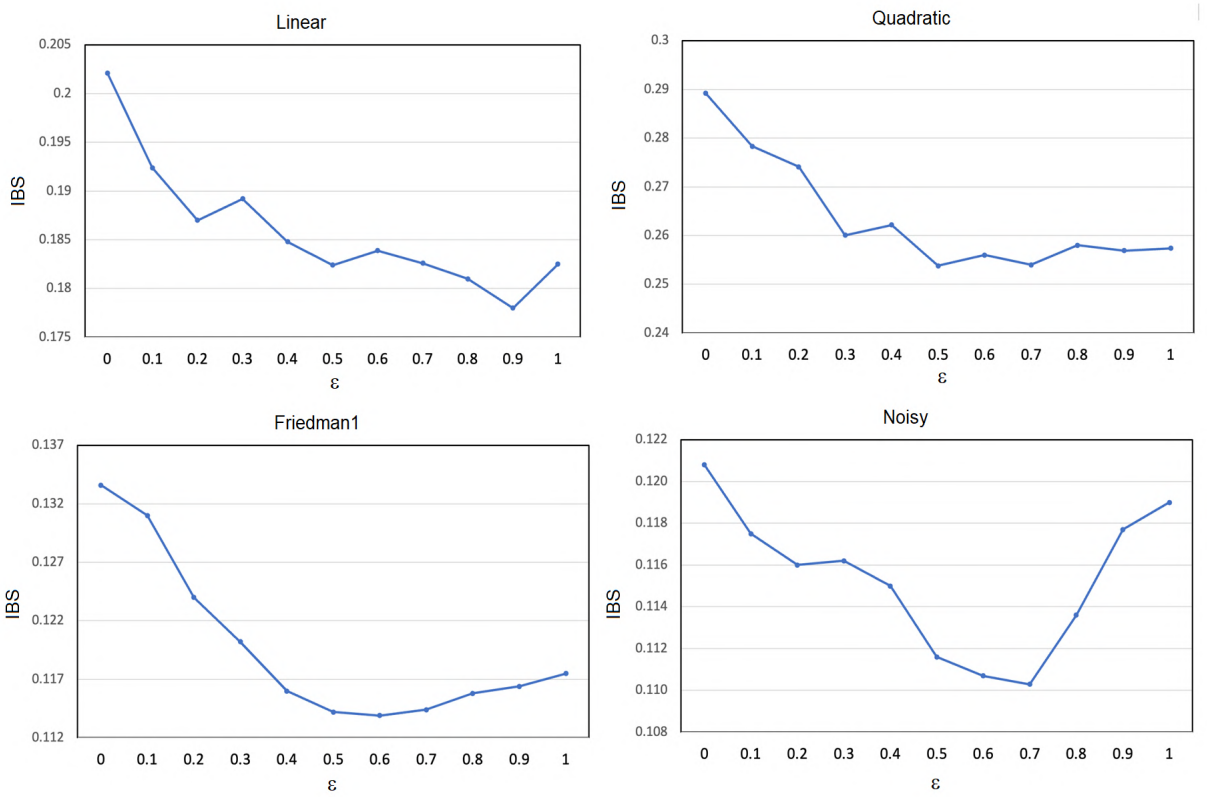

Figure 3. IBS metrics as functions of the contamination parameter ϵ for some synthetic datasets obtained by using CiSurvN

Table 4. IBS obtained for synthetic datasets by using CiSurvG for different ϵ

ϵ	Linear	Quadratic	Friedman1	Noisy
0.0	0.2032	0.3264	0.1222	0.1322
0.1	0.2018	0.2961	0.1210	0.1263
0.2	0.2008	0.2741	0.1194	0.1231
0.3	0.1998	0.2670	0.1175	0.1213
0.4	0.1989	0.2650	0.1166	0.1209
0.5	0.1982	0.2632	0.1157	0.1207
0.6	0.1976	0.2601	**0.1151**	0.1205
0.7	0.1971	0.2584	0.1154	0.1203
0.8	0.1968	0.2578	0.1157	0.1202
0.9	0.1967	**0.2569**	0.1161	0.1202
1.0	**0.1965**	0.2574	0.1170	**0.1201**

CiSurvN variant, offers an approach to survival analysis that can adapt to different data characteristics through appropriate selection of the contamination parameter ϵ. The findings suggest that relaxing the strict assumptions of traditional survival models through the ϵ-contaminated approach can lead to improved predictive performance.

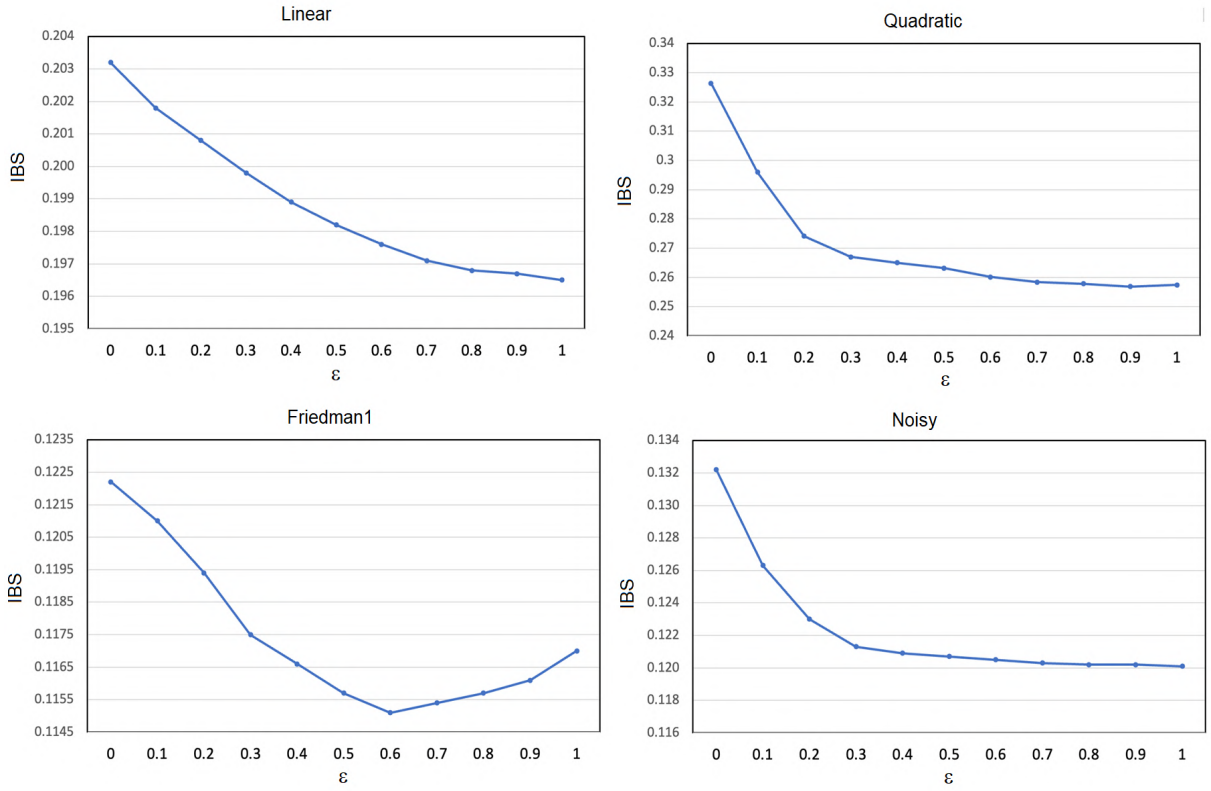

Figure 4. IBS metrics as functions of the contamination parameter ϵ for some synthetic datasets obtained by using CiSurvG

Conclusion

The proposed CiSurv framework introduces a novel approach to survival analysis by integrating the imprecise ϵ-contaminated model into an attention-based multi-label classification method. This reformulation of survival analysis as an imprecise probability problem provides a flexible and robust solution for time-to-event prediction, particularly in scenarios involving censored data.

The main contributions of this work include the successful integration of imprecise probability theory through the ϵ-contaminated model, which reduces uncertainty in censored observations by refining interval-valued probabilities. This refinement leads to more accurate survival predictions compared to the existing iSurvJ approach, where $\epsilon = 1$.

Numerical experiments confirmed that intermediate values of ϵ often yield the best predictive performance. The results have illustrated that modeling uncertainty explicitly via the ϵ-contaminated framework offers a promising pathway for developing robust, flexible survival analysis tools capable of managing real-world data heterogeneity and uncertainty. It should be noted that the neural network-based CiSurvN tends to outperform simpler kernel-based variants, underscoring the benefit of increased model capacity in complex data regimes.

The model capacity to handle imprecision in probabilistic estimates makes it well-suited for real-world applications characterized by data uncertainty, limited sample sizes, or complex censoring patterns. Extending CiSurv to incorporate other imprecise probability models, for example, the pari-mutuel model, the constant odds-ratio model [21] could enhance its applicability.

This is a direction for further research. It is interesting to extend the approach to competing risks and time-varying covariates. These are also directions for further research.

The proposed survival analysis model CiSurv and its variants CiSurvN and CiSurvG can be employed to predict system reliability and proactively schedule maintenance in HPC environments minimizing downtime. By modeling task execution times and failure probabilities, CiSurv enhances resource allocation and workload management in high-performance computing systems. The imprecise probabilistic framework enables more accurate failure prediction under uncertain and incomplete HPC system monitoring data. Finally, applying CiSurv can improve fault-tolerance strategies in HPC, ensuring more resilient operation and optimal utilization of computational resources.

Acknowledgements

The research is partially funded by the Ministry of Science and Higher Education of the Russian Federation (topic FSEG-2024-0027).

References

1. Abbasi, A., Asim, M., Ahmed, S., *et al.*: Survival prediction landscape: an in-depth systematic literature review on activities, methods, tools, diseases, and databases. Frontiers in Artificial Intelligence 7, 1428501 (2024). https://doi.org/10.3389/frai.2024.1428501

2. Beran, R.: Nonparametric regression with randomly censored survival data. Tech. rep., University of California, Berkeley (1981).

3. Brier, G.: Verification of forecasts expressed in terms of probability. Monthly Weather Review 78(1), 1–3 (1950). https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2

4. Chen, G.H.: An introduction to deep survival analysis models for predicting time-to-event outcomes. Foundations and Trends® in Machine Learning 17(6), 921–1100 (2024). https://doi.org/10.1561/2200000114

5. Coolen, F.: An imprecise Dirichlet model for Bayesian analysis of failure data including right-censored observations. Reliability Engineering and System Safety 56, 61–68 (1997). https://doi.org/10.1016/S0951-8320(96)00131-7

6. Coolen, F., Yan, K.: Nonparametric predictive inference withright-censored data. Journal of Statistical Planning and Inference 126, 25–54 (2004). https://doi.org/10.1016/j.jspi.2003.07.004

7. Craig, E., Zhong, C., Tibshirani, R.: A review of survival stacking: a method to cast survival regression analysis as a classification problem. The International Journal of Biostatistics 21(1), 37–51 (2025). https://doi.org/10.1515/ijb-2022-0055

8. Emmert-Streib, F., Dehmer, M.: Introduction to survival analysis in practice. Machine Learning & Knowledge Extraction 1, 1013–1038 (2019). https://doi.org/10.3390/make1030058

9. Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M.: Assessment and comparison of prognostic classification schemes for survival data. Statistics in Medicine 18(17-18), 2529–2545 (1999). https://doi.org/10.1002/(SICI)1097-0258(19990915/30)18:17/18<2529::AID-SIM274>3.0.CO;2-5

10. Hosmer, D., Lemeshow, S., May, S.: Applied Survival Analysis: Regression Modeling of Time to Event Data. John Wiley & Sons, New Jersey (2008).

11. Hu, S., Fridgeirsson, E., van Wingen, G., Welling, M.: Transformer-based deep survival analysis. In: Survival Prediction-Algorithms, Challenges and Applications. pp. 132–148. PMLR (2021).

12. Ishwaran, H., Kogalur, U.: Random survival forests for r. R News 7(2), 25–31 (2007).

13. Jiang, S., Suriawinata, A.A., Hassanpour, S.: MHAttnSurv: Multi-head attention for survival prediction using whole-slide pathology images. Computers in Biology and Medicine 158, 106883 (2023). https://doi.org/10.1016/j.compbiomed.2023.106883

14. Konstantinov, A., Utkin, L., Efremenko, V., *et al.*: Survival analysis as imprecise classification with trainable kernels. Mathematics 13(18), 3040 (2025). https://doi.org/10.3390/math13183040

15. Kvamme, H., Borgan, Ø.: Continuous and discrete-time survival prediction with neural networks. Lifetime Data Analysis 27(4), 710–736 (2021). https://doi.org/10.1007/s10985-021-09532-6

16. Luong, T., Pham, H., Manning, C.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1412–1421. Association for Computational Linguistics, Lisbon, Portugal (2015). https://doi.org/10.18653/v1/d15-1166

17. Nadaraya, E.: On estimating regression. Theory of Probability & Its Applications 9(1), 141–142 (1964). https://doi.org/10.1137/1109020

18. Softic, S., Hrnjica, B.: Case studies of survival analysis for predictive maintenance in manufacturing. International Journal of Industrial Engineering and Management 15(4), 320–337 (2024). https://doi.org/10.24867/IJIEM-2024-4-366

19. Suresh, K., Severn, C., Ghosh, D.: Survival prediction models: an introduction to discrete-time modeling. BMC Medical Research Methodology 22(1), 207 (2022). https://doi.org/10.1186/s12874-022-01679-6

20. Vaswani, A., Shazeer, N., Parmar, N., *et al.*: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017).

21. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991).

22. Walley, P.: Inferences from multinomial data: Learning about a bag of marbles. Journal of the Royal Statistical Society, Series B 58, 3–34 (1996). https://doi.org/10.1111/j.2517-6161.1996.tb02065.x

23. Wang, P., Li, Y., Reddy, C.: Machine learning for survival analysis: A survey. ACM Computing Surveys (CSUR) 51(6), 1–36 (2019). https://doi.org/10.1145/3214306

24. Wang, Y., Kong, X., Bi, X., *et al.*: ResDeepSurv: A survival model for deep neural networks based on residual blocks and self-attention mechanism. Interdisciplinary Sciences: Computational Life Sciences 16(2), 405–417 (2024). https://doi.org/10.1007/s12539-024-00617-y

25. Wang, Z., Sun, J.: SurvTRACE: Transformers for survival analysis with competing events. In: Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. pp. 1–9 (2022). https://doi.org/10.1145/3535508.3545521

26. Watson, G.: Smooth regression analysis. Sankhya: The Indian Journal of Statistics, Series A, 359–372 (1964).

27. Wiegrebe, S., Kopper, P., Sonabend, R., *et al.*: Deep learning for survival analysis: a review. Artificial Intelligence Review 57(65), 1–34 (2024). https://doi.org/10.1007/s10462-023-10681-3

28. Zhang, X., Mehta, D., Hu, Y., *et al.*.: Adaptive transformer modelling of density function for nonparametric survival analysis. Machine Learning 114(2), 31 (2025). https://doi.org/10.1007/s10994-024-06686-w