Artificial Intelligence Review (2024) 57:65
https://doi.org/10.1007/s10462-023-10681-3

Deep learning for survival analysis: a review

Simon Wiegrebe[1,3,4] · Philipp Kopper[2,3] · Raphael Sonabend[5] · Bernd Bischl[2,3] · Andreas Bender[2,3]

Accepted: 20 December 2023 / Published online: 19 February 2024

Abstract
The influx of deep learning (DL) techniques into the field of survival analysis in recent years has led to substantial methodological progress; for instance, learning from unstructured or high-dimensional data such as images, text or omics data. In this work, we conduct a comprehensive systematic review of DL-based methods for time-to-event analysis, characterizing them according to both survival- and DL-related attributes. In summary, the reviewed methods often address only a small subset of tasks relevant to time-to-event data—e.g., single-risk right-censored data—and neglect to incorporate more complex settings. Our findings are summarized in an editable, open-source, interactive table: https://survival-org.github.io/DL4Survival. As this research area is advancing rapidly, we encourage community contribution in order to keep this database up to date.

Keywords Survival analysis · Time-to-event analysis · Deep learning · Review

1 Introduction

Survival analysis (SA), or equivalently *time-to-event analysis*, comprises a set of techniques enabling the unbiased estimation of the distribution of outcome variables that are partially censored, truncated, or both. Usually, the outcome is given by the time until the occurrence of an event such as death, system failure, or time to remission.

Non-parametric methods like the Kaplan–Meier estimator (Kaplan and Meier 1958) are baseline tools still used today, yet semi-parametric methods received the most attention

✉ Simon Wiegrebe
simon.wiegrebe@stat.uni-muenchen.de

✉ Andreas Bender
andreas.bender@stat.uni-muenchen.de

[1] Statistical Consulting Unit StaBLab, Department of Statistics, LMU Munich, Munich, Germany

[2] Department of Statistics, LMU Munich, Munich, Germany

[3] Munich Center for Machine Learning (MCML), LMU Munich, Munich, Germany

[4] Department of Genetic Epidemiology, University of Regensburg, Regensburg, Germany

[5] MRC Centre for Global Infectious Disease Analysis, Jameel Institute, School of Public Health, Imperial College London, London, UK

historically, in particular the Cox proportional hazards regression model (Cox 1972) and its extensions. Since the early 2000s, Machine Learning (ML) methods have been successfully adapted to survival tasks: e.g., Random Survival Forest (Ishwaran et al. 2008) and boosting-based methods (Binder and Schumacher 2008). These methods often outperform traditional statistical models in terms of predictive power (Steele et al. 2018) (see Wang et al. (2019) and Sonabend (2021) for detailed discussions).

Neural networks (NNs) had already been applied to survival tasks in the 1990s (Faraggi and Simon 1995; Brown et al. 1997), but were shallow and restricted to the most standard survival settings. Most modern Deep Learning (DL) survival models have been developed only since the late 2010s, as indicated by the publication year of the methods we review; see *Main Table* (https://survival-org.github.io/DL4Survival).

Despite the large number of DL-based survival methods proposed in recent years, to the best of our knowledge, there is no general systematic review of these methods. Schwarzer et al. (2000) summarize misuses in early applications of NNs to clinical data. Lee and Lim (2019) and Deepa and Gunavathi (2022) do not explicitly focus on DL-based survival methods, do not address any survival-related specifics of DL, and are restricted to use cases involving genomics data and cancer survival prediction, respectively. The glioma-focused survey by Wijethilake (2021) as well as the benchmarking study by Zhang et al. (2022) consider only few NN-based methods and thus do not provide a general overview of DL methods for time-to-event data either.

Motivated by the above, in this paper we provide a comprehensive review of currently available DL-based survival methods, addressing theoretical dimensions, such as model class and NN architecture, as well as data-related aspects, such as outcome types and feature-related aspects (see Sect. 3 for definitions). Table 1 gives an overview of the dimensions we consider.

This paper is structured as follows. Section 2 introduces SA notation and concepts (Sect. 2.1), common data-related aspects of survival tasks (Sect. 2.2), as well as estimation of survival models (Sect. 2.3). Section 3 outlines the review methodology (Sect. 3.1), explains general NN architecture choices in SA (Sect. 3.2), and eventually provides a detailed, comprehensive overview of all methods reviewed, covering estimation and network architecture (Sect. 3.3), and supported survival tasks in terms of outcome types and feature-related aspects (Sect. 3.4); findings are summarized in the *Main Table*. Finally, Sect. 4 concludes, discusses limitations, and provides an outlook.

2 Theoretical concepts and data-related aspects

In this section, we first introduce quantities that are targets of estimation in SA and characterize the distribution of a random variable $T > 0$. Later, we describe censoring and truncation, which need to be accounted for in order to estimate these quantities (see Sect. 2.2).

2.1 Targets of estimation

Initially, assume that T is continuous. Let $f_T(t)$ and $F_T(t) := P(T \leq t)$ be its density and cumulative distribution function, respectively. Then, the survival function of T is defined as

$$S_T(t) := P(T > t) = 1 - F_T(t),$$

i.e., the probability of surviving beyond t. The hazard rate,

Table 1 Overview of theoretical and practical dimensions reviewed

Dimension	Examples	Section(s)
Estimation	Cox-based, discrete-time, PEM-, ODE-, or continuous-time ranking-based	Sect. 3.3.1
Neural network architecture	FFNN, CNN, RNN, AE, transformer, flexible, nODE	Sects. 3.2 and 3.3.2
Outcome types	Interval-, right- and left-censoring, right- and left-truncation, CR, MSM, recurrent events	Sects. 2.2.1 and 3.4.1
Feature-related aspects	TVF, TVE, multimodality, high-dimensional features	Sects. 2.2.2 and 3.4.2
Interpretability	Inherent interpretability, post-hoc methods	Sect. 3.5

PEM piecewise exponential model, *ODE* ordinary differential equation, *FFNN* feed-forward neural network, *CNN* convolutional neural network, *RNN* recurrent neural network, *AE* autoencoder, *nODE* neural ODE, *CR* competing risks, *MSM* multi-state, *TVF* time-varying features, *TVE* time-varying effects

$$h_T(t) := \lim_{\Delta t \to 0} \frac{1}{\Delta t} P(t \leq T < t + \Delta t | T \geq t) = \frac{f_T(t)}{S_T(t)}, \tag{1}$$

is the instantaneous risk of the event occurring given it has not yet occurred at time t. Finally, the cumulative hazard, defined as

$$H_T(t) := \int_0^t h_T(u)du = -\log(S_T(t)),$$

is often used as an intermediate step when calculating the survival probability.

In the above, T was assumed to be continuous. However, sometimes the time scale is discrete by nature (e.g., grade level at the time of school dropout) or a continuous time scale is discretized into intervals. With discrete event times, the discrete hazard

$$h_T(t) := P(T = t | T \geq t), \; t = 1, 2, \ldots \tag{2}$$

is the probability of the event occurring in the time interval t conditional upon the individual still being alive at the beginning of t (cf. Tutz et al. (2016) for details). This gives rise to the discrete-time survival probability $S_T(t) := P(T > t) = \prod_{j=1}^{t}(1 - h_T(j))$. Some discrete time methods on the other hand directly estimate the probability mass function (PMF), i.e. $P(T = t)$, rather than estimating the discrete hazard (2).

2.2 Data-related aspects

We now discuss different data-related aspects of time-to-event data, in terms of both outcomes and features, that are frequently encountered in real-world survival tasks. We refer to them as outcome types (Sect. 2.2.1) and feature-related aspects (Sect. 2.2.2), respectively.

In Sect. 3.4, we provide detailed information regarding which of the reviewed methods can handle these data-related aspects.

2.2.1 Outcome types

Throughout this work, we consider a sample of size n and refer to a single $i \in \{1, \ldots, n\}$ as *individual* or *subject*. Let $T_i > 0$ be the non-negative random variable representing the time until the event of interest for subject i occurs. We want to estimate the distribution of T_i given the p-dimensional feature vector $\mathbf{x}_i$. However, T_i often cannot be fully observed because the time-to-event is right-, left- or interval-censored. Let C_i^L and C_i^R be the left- and right-censoring times, and let L_i and R_i be the endpoints of the censoring interval for subject i, respectively. For an interval-censored observation, we have $T_i \in (L_i, R_i]$ as we only know that the event occurs within the interval, but not the exact time. Right-censoring $T_i \in (L_i = C_i^R, \infty]$ and left-censoring $T_i \in (L_i = 0, R_i = C_i^L]$ are special cases of interval-censoring.

Time-to-event data can also be subject to truncation. In SA, truncation implies that subjects are either not part of the dataset at all or not part of the risk set for a specific event at certain time points. Formally, let T_i^L and T_i^R be the left- and right-truncation times, respectively. Left-truncation occurs when $T_i^R = \infty$, then subjects with $T_i < T_i^L$ never enter the study. Similarly, observations are right-truncated when $T_i^L = 0$ and $T_i > T_i^R$.

Survival tasks are not restricted to single-risk scenarios. In case of *competing risks*, each individual can experience only one of at least two distinct, mutually exclusive events,

e.g., death in hospital versus hospital discharge. More generally, in a *multi-state* setting multiple (transient and terminal) events (states) are possible, as well as certain (recurring) transitions between them, e.g., transitions between different stages of an illness with death as terminal event. We denote transitions by $k \in \{1, \dots, K\}$ and episodes by $e = 1, \dots, E$. A final outcome type we consider is *recurrent events*. Often, we record a single outcome (censoring or event) for each individual. However, when conditions such as epilepsy or sports injuries are being modeled, subjects may experience the same event type repeatedly.

Table 2 provides an overview and examples of the outcome types discussed in this section.

2.2.2 Feature-related aspects

Time-varying features (TVFs) such as weight or lifestyle factors change over time, whereas others such as sex are time-constant. Similarly, *time-varying effects* (TVEs) are feature effects on the outcome (e.g., on the hazard rate) that vary over time. Both TVFs and TVEs constitute deviations from the proportional hazards (PH) assumption (see Sect. 2.3).

Another important feature-related aspect is the dimensionality of data input. Due to the prominence of SA in the life sciences, features derived from high-dimensional data—*omics* data in particular—are sometimes employed to predict and explain survival times. In order for a method to learn from a high-dimensional feature space, the model architecture needs to be adapted, usually with appropriate penalization or feature selection techniques (see, e.g., Wu 2019).

Multimodality is the final feature-related aspect we consider. In the life sciences, in particular, we are oftentimes not restricted to structured tabular data (e.g., clinical patient data), but also have access to unstructured data, such as images (e.g., CT scans)

Table 2 Overview of different outcome types

Outcome type	Example
Right-censoring	Clinical trials: exact event times are unobserved for some individuals because of dropout
Left-censoring	Age at which children learn a certain task: some children already know the task at the beginning of the study, but it is unknown at which age they learned it
Interval-censoring	Medical study with a periodic follow-up: exact event times are unknown, only the interval between two follow-ups is known
Right-truncation	Transfusion-induced AIDS onset study (Klein and Moeschberger 1997): only patients developing AIDS from transfusion before the registry sampling date are included, while patients with onset after that date are right-truncated
Left-truncation	Coumarin abortion study (Meister and Schaefer 2008): only women conscious of their pregnancy are included; women who had a spontaneous abortion before their pregnancy is recognized never enter the study
Competing risks	Study on dialysis mortality (Noordzij 2013): the event of interest, death on dialysis, is precluded by the competing event kidney transplantation
Multi-state	Study on kidney failure: for all patients, transitions between the states *healthy*, *dialysis*, *kidney transplantation* and *death* are possible, sometimes even bidirectionally (e.g., between *dialysis* and *kidney transplantation*)
Recurrent event	Incidence of pneumonia in young children (Ramjith et al. 2021): the occurrence of multiple, recurrent pneumonia episodes is possible, with episodes within a child's history not being independent

or text data (e.g., written doctor's notes); that is, the feature set is multimodal and special techniques are required to extract information from it.

2.3 Estimation

Here we summarize estimation in the SA context, focusing on the methods most frequently used among the DL-based approaches included in this review.

In SA we want to estimate the distribution of event times based on observed data, represented by tuples

$$\left(y_{i,k,e}^{entry}, y_{i,k,e}^{exit}, \delta_{i,k,e}, \mathbf{x}_{i,k,e}\right),$$

with $y_{i,k,e}^{entry}$ and $y_{i,k,e}^{exit}$ defining entry and exit times of subject $i = 1, \dots, n$ into the risk set for transition $k \in \{1, \dots, K\}$ in episode $e = 1, \dots E$, respectively, and $\delta_{i,k,e} \in \{0, 1\}$ being the indicator for whether the respective transition has been actually observed (rather than censored). Finally $\mathbf{x}_{i,k,e}$ represents the p-dimensional feature vector (for simplicity we omit that $\mathbf{x}_{i,k,e}$ could additionally vary over time between the entry and exit times for transition k in episode e). Often this notation can be simplified. For example, when all subjects enter the risk set at time point 0 and there is no truncation or interval-censoring, $y_{i,k,e}^{entry}$ is omitted. When we only consider one single event type, we can drop index k. If there are no recurrent transitions, we can additionally drop e, yielding the more common notation $(y_i, \delta_i, \mathbf{x}_i)$.

Parametric survival models, such as the Accelerated Failure Time (AFT) model (Kalbfleisch and Prentice 2011), assume event times to follow a certain statistical distribution characterized by a set of parameters. Based on the distribution-specific likelihood, parametric survival models then estimate these distributional parameters as a function of features $\mathbf{x}$. We can write the density for an event at time t as

$$f(t|\boldsymbol{\theta}), t \geq 0, \tag{3}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}(\mathbf{x}) = (\theta_1(\mathbf{x}), \theta_2(\mathbf{x}), \dots) = (g_1(\mathbf{x}, \boldsymbol{\beta}_1), g_2(\mathbf{x}, \boldsymbol{\beta}_2), \dots), \tag{4}$$

where $g_1(), g_2(), \dots$ are real-valued functions associating features $\mathbf{x}$ with the distributional parameters $\boldsymbol{\theta}(\mathbf{x})$ via parameters $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots$. That is, all distributional parameters (e.g., both shape and scale of a Weibull distribution) can be estimated as a function of $\mathbf{x}$. Estimation proceeds by maximizing the likelihood given the observed data

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} L_i(\boldsymbol{\theta}) = \prod_{o \in \mathcal{O}} f(y_o) \times \prod_{c \in \mathcal{C}} S(y_c) \times \prod_{l \in \mathcal{L}} (1 - S(y_l)) \times \cdots, \tag{5}$$

where L_i are the individual likelihood contributions, depending on the observed outcome type, and $\mathcal{O}$, $\mathcal{C}$, $\mathcal{L}$ are the sets of observed event times, right-censored, and left-censored observations, respectively. Likelihood contributions for other outcome types can be constructed similarly (e.g., Klein and Moeschberger 1997, Ch. 3.5).

Other methods exploit the relationships $f(t) = h(t)S(t)$ and $S(t) = \exp\left(-\int_0^t h(u)du\right)$, such that the likelihood can always be expressed in terms of only the hazard rate (1), and right-censoring and left-truncation are dealt with by adjusting the so-called risk set

$$\mathcal{R}(t) = \left\{ \left(y_i^{entry}, y_i^{exit}, \delta_i, \mathbf{x}_i \right) : y_i^{entry} < t \leq y_i^{exit} \right\}.$$

Most prominently, the Cox PH regression (Cox 1972) models the hazard rate at time t, conditional on features $\mathbf{x}$, as the product of a non-parametrically estimated baseline hazard $h_0(t)$ and the exponentiated log-risk $\eta = g(\mathbf{x}, \boldsymbol{\beta})$:

$$h(t|\mathbf{x}) = h_0(t) \exp(\eta = g(\mathbf{x}, \boldsymbol{\beta})). \tag{6}$$

Feature effects are multiplicative with respect to the hazard rate independently of time, yielding proportionality of hazards.

Parameters are estimated by optimizing the log-partial-likelihood

$$Pl(\boldsymbol{\theta}) = \sum_{m=1}^{M} \left(g(\mathbf{x}_{(m)}, \boldsymbol{\beta}) - \log \sum_{j \in \mathcal{R}(t_{(m)})} \exp\left(g(\mathbf{x}_j, \boldsymbol{\beta}) \right) \right), \tag{7}$$

where $t_{(m)}$ is the mth ordered event ($m \in \{1, \ldots, M\}$), $\mathcal{R}(t_{(m)})$ denotes the risk set at that time point, and $\mathbf{x}_{(m)}$ is the feature vector of the individual experiencing the event at $t_{(m)}$.

Piecewise Exponential Models (PEMs) also parametrize the hazard rate as in (6). However, by partitioning the time axis into J intervals and assuming piecewise constant hazards within each interval, the baseline hazard is parametrized and estimated alongside the feature-related coefficients. Friedman (1982) showed that the likelihood of this model is proportional to a Poisson likelihood, which implies that, after appropriate data transformation, any method capable of minimizing a negative Poisson log-likelihood can also be used for various survival tasks (Bender et al. 2021). Despite partitioning the follow-up into intervals, PEM-based approaches are methods for continuous time-to-event data as they take the full information about event times into account.

Discrete-time survival methods, such as discrete hazard methods (Tutz et al. 2016) or Multi-Task Logistic Regression (MTLR; Yu et al. 2011), consider the time-to-event data to be a succession of binary outcomes. To do so, the time axis is first partitioned into intervals, with $T = t$ implying the event occurred in interval $(a_{t-1}, a_t]$. Binary event indicators y_{it} are then defined for each time interval t and used as outcomes. For individual i, the discrete hazard $h_i(t|\mathbf{x}_i)$ in interval t is then

$$h_i(t|\mathbf{x}_i) = \phi(g(\mathbf{x}, \boldsymbol{\beta})) = P(y_{it} = 1 | T \geq t, \mathbf{x}_i), \tag{8}$$

where the real-valued function $g()$ represents feature effects and $\phi()$ maps this quantity onto $[0, 1]$ to yield the conditional event probability $P(y_{it} = 1|T \geq t, \mathbf{x}_i)$. Analogously to PEM-based approaches, any ML algorithm that is applicable to binary outcomes can be used for discrete-time survival modeling after data transformation. The logit model, for instance, uses a logistic response function to model the probability of the event taking place in t [i.e., the discrete hazard (2)], conditional on $a_{t-1} < t$ and feature values $\mathbf{x}$. Alternatively, some discrete-time methods directly estimate the probability of an event at specified time points $P(y_{it} = 1|T = t, \mathbf{x}_i)$ using a softmax output layer.

3 Deep learning in survival analysis

Early DL-based survival techniques date back to the mid-1990s (Liestbl et al. 1994; Faraggi and Simon 1995; Brown et al. 1997) and are usually NN-based extensions of classical statistical survival methods discussed in Sect. 2.3. While in the Cox model the log-risk (6) is traditionally given by $g(\mathbf{x}, \boldsymbol{\beta}) = \mathbf{x}^\top \boldsymbol{\beta}$, the model by Faraggi and Simon (1995) replaces the linear predictor by a shallow feed-forward neural network (FFNN). Liestbl et al. (1994) propose implementing the PEM as an NN, yet without any hidden layers. The *PEANN* model by Fornili et al. (2013) parametrizes the piecewise constant hazards by a shallow FFNN. *PLANN* (Biganzoli et al. 1998) is an NN-based extension of the discrete-time logit model, parametrizing the discrete hazard by an FFNN.

Many DL-based methods for SA have been developed in recent years. They usually build upon one of the aforementioned statistical survival approaches, while harnessing advantages of NNs. Furthermore, recent advances in multimodal learning and interpretability have made DL-based survival methods even more attractive for many common survival tasks.

3.1 Inclusion and exclusion criteria

For this review, we designed a two-step literature screening process. In the first step (inclusion criteria), we searched Web of Science for the topic

```
 ("survival analysis" or "time-to-event analysis" or
"survival data" or "time-to-event data") AND
("neural network" or "deep learning") AND
("model" or "method") AND
("performance" or "evaluation" or "comparison" or
"benchmark")
```

with December 31, 2022 as cutoff date. These inclusion criteria resulted in a total of 211 articles. In the second step (exclusion criteria), we excluded all articles not satisfying *all* of the following four conditions:

(a) Development of a new DL-based method beyond the mere application of an already existing method to new data or contexts.
(b) Evaluation of performance results on at least one non-private benchmark dataset.
(c) Performance evaluation using metrics designed for time-to-event data, such as C-index or Integrated Brier Score.
(d) Focus on estimation and prediction in the context of time-to-event data and learning all model parameters within the NN architecture in an end-to-end fashion.

Criteria (a), (b), and (c) aim to ensure that the paper in question develops a new method rather than applying a known method to new data or in a new context. Criterion (b) complements (a) as the predictive utility is often illustrated via benchmark experiments when new methods are proposed. Additionally, criterion (b) introduces an open science aspect and ensures that at least one empirical comparison could be replicated in theory. Criterion (c) ensures that benchmark analyses focus on methods modeling time-to-event

data, as some papers that passed the initial screening eventually ignored the time-to-event nature of the data. Finally, criterion (d) aims to exclude two-step approaches where DL is used solely for feature extraction, with survival modeling performed using non-DL approaches with the extracted features outside of the NN. Our criteria are motivated by the scope of this work—to review methods that can be used for specific time-to-event problems and to provide details on estimation-, architecture- and data-related aspects of the respective methods.

Subsequently, we combined the selected articles with additional papers that had otherwise come to our attention and fulfilled the above criteria, yielding a total of 61 articles—and thus, 61 distinct methods. The inclusion, exclusion, and screening process is visualized in the PRISMA diagram in Fig. 1.

The following naming scheme is used in the remainder of the paper to reference individual methods/papers: the method name as specified in the publication, if provided and unique; if the method name is not unique, we append a suffix (the first three letters of the first author's last name followed by the year of publication) with an underscore; if no method name is provided, we use this suffix as a name. All methods are summarized in our *Main Table* (https://survival-org.github.io/DL4Survival).

We now aim to provide a summary of the 61 methods based on a broad range of both theoretical (estimation and architecture) as well as practical model characteristics (outcome types and feature-related aspects).

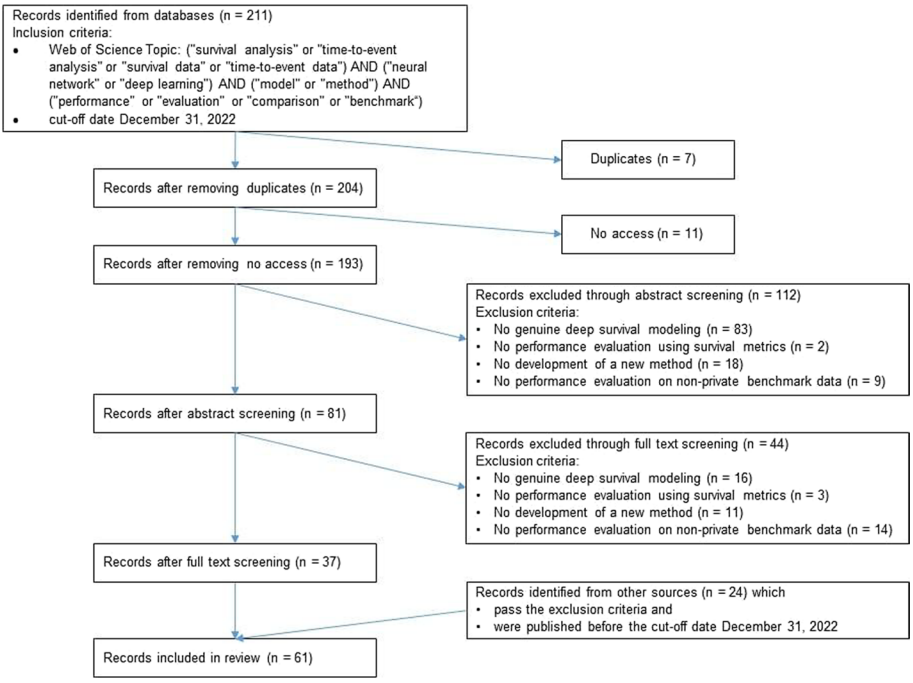

Fig. 1 PRISMA diagram for literature screening of deep learning-based survival methods

3.2 Architectural choices

FFNNs were the earliest type of NN architecture (Ivakhnenko et al. 1967; Rosenblatt 1967). Within an FFNN, information passes from the input nodes through a user-specified number of hidden layers until the output nodes. Information only flows forward as there are no cyclical patterns or loops. The main property of FFNNs is stated through the universal approximation theorem (Hornik et al. 1989), meaning that NNs are capable of approximating a very general class of functions. Practically, this allows FFNN to discover non-linear feature effects and complex interaction structures. In SA, FFNNs naturally allow for a more flexible estimation of, e.g., (semi-)parametric hazard rates, as well as for the incorporation of TVEs and TVFs (in theory); for instance, the hazard rate in (6) can be estimated more flexibly by parametrizing $g(\mathbf{x})$ through an NN. At the same time, the FFNN architecture contains multiple limitations: for example, learning from multimodal data input—in particular, image data—is not possible. FFNNs constitute the main architecture of most early DL-based survival methods and still serve as a baseline building-block within most advanced architectures.

Convolutional neural networks (CNNs) were introduced in the late 1980s (LeCun 1989) and are most successfully employed in computer vision. In time-to-event analysis, CNNs are usually applied to unstructured data, especially images. Often, CNN-based methods use large pre-trained CNNs with many parameters, such as ResNet18 (He et al. 2016), and then fine-tune them on case-specific data. This transfer learning approach enables the application of large CNNs to smaller datasets.

Recurrent neural networks (RNNs), also invented in the 1980s (Rumelhart et al. 1986), distinguish themselves from FFNNs and CNNs by being able to memorize parts of the input through a short-term memory and are thus applicable to sequential data. In SA, RNNs are hence useful when TVFs are present or to take temporal information into account in general.

The autoencoder (AE; Ballard 1987) is another common NN architecture, learning how to reduce the dimensionality of input data and subsequently reconstructing the data from the learned latent representation; extensions include stacked AEs (SAEs; Vincent 2010) and variational AEs (VAEs; Kingma and Welling 2013). General Adversarial Networks (GANs; Goodfellow et al. 2014) consist of a generator that produces synthetic data of gradually improving quality as well as a discriminator that learns how to distinguish between true data input and generator-produced data points. Transformers (Vaswani et al. 2017) use an attention mechanism to learn a representation of context in sequential (e.g., language) data and can subsequently produce output (sequences) from it. Normalizing flows (NFs; Rezende and Mohamed 2015) constitute a family of generative models which employ differentiable and invertible mappings to obtain complex distributions from a simple initial probability distribution for which sampling and density evaluation is easy. Neural Ordinary Differential Equations (nODEs; Chen et al. 2018) use NNs to parametrize the derivative of the hidden state, thus moving beyond the standard specification of a discrete sequence of hidden layers. Fuzzy neural networks (Lee and Lee 1975) use fuzzy numbers as inputs and weights within the NN. Diffusion models (Sohl-Dickstein et al. 2015) employ a Markov chain to gradually add random noise to the input data and subsequently learn to undo this diffusion, learning to generate new data from noise.

Many adoptions of NNs for SA emphasize the replacement of the predictors in (4), (6), or (8) through a (deep) NN. The (DL-based) survival models can be further

extended to also include interactions, non-linear effects, stratification, time-varying effects, and even unstructured components $d(\mathbf{z})$, yielding the generalized predictor

$$\eta = g(\mathbf{x}, \mathbf{z}, t) = f(\mathbf{x}, t) + \gamma_1 d_1(\mathbf{z}_1) + \cdots + \gamma_G d_G(\mathbf{z}_G), \tag{9}$$

where $f(\mathbf{x}, t)$ denotes potentially non-linear, time-varying effects of tabular features $\mathbf{x}$ as well as their interactions. $d_g(\mathbf{z}_g)$ denotes embeddings learned in the deep part(s) of the model from unstructured data sources $\mathbf{z}_g$, $g \in \{1, \ldots, G\}$, such as images or text. That is, the predictor $g(\mathbf{x})$ from (7) can be generalized to be $g(\mathbf{x}, \mathbf{z}, t)$. Using an appropriate transformation function ψ predictor (9) can be transformed to e.g. the hazard function or cumulative incidence function, depending on the target of estimation.

Architectural choice is also closely related to parametrization. The PMF of discrete-time methods can be modeled via a softmax layer producing discrete survival probabilities at each (pre-defined) time point, as done in Lee et al. (2018). RNN architectures are particularly suitable for taking into account temporal information and sharing parameters across time, e.g., in order to estimate quantities like the hazard rate or survival probability at time t using information from time points $\tilde{t} < t$ (e.g., Giunchiglia et al. 2018). Some less frequently encountered architectures, for example GANs, incentivize the development of custom losses (Chapfuwa et al. 2018). More recent work shows that (surrogate) loss functions can be created based on scoring rules, such as a smooth C-index loss function (Huang et al. 2018) or Survival-CRPS (Avati et al. 2020), for parameter estimation without requiring traditional inner loss functions like the negative log-likelihood.

It is furthermore possible to directly integrate some time-to-event data modalities into the architecture of deep survival models. For example, shared and cause-specific subnetworks for cause- or transition-specific hazards in competing risks and multi-state modeling analysis via soft- or hard-sharing of parameters (Ruder 2017) have been adopted by many DL-based survival methods when modeling transitions between different states (cf. Fig. 2).

Additionally, many methods have shown how to integrate multimodal data, by using a separate subnetwork for each modality. For instance, one may use a CNN-based subnetwork for image data while tabular data is modeled with an FFNN. The different modalities can be fused together in different ways in the network head. If interaction between different modalities is desired, vector representations of the data are concatenated and fed through another joint FFNN. Otherwise, separate scalars are learned and added onto each other. We illustrate two common architectures that tackle competing risks and multiple data modalities—and can also be combined—in Figs. 2 and 3.

3.3 Estimation and network architecture

We now review all 61 DL-based survival methods based on theoretical and technical aspects. In Sect. 3.3.1, we aim to categorize the methods in terms of estimation-related concepts—model class, loss functions, and parametrization—and how these concepts correlate. In Sect. 3.3.2, we address the NN architecture choices of all methods reviewed.

3.3.1 Estimation

We classify DL-based survival methods in terms of three concepts related to model estimation. First, the *model class* (cf. Fig. 4) describes which type of statistical survival technique forms the basis of the DL method—usually one of the approaches introduced in Sect. 3.3.1. Second, the *loss function* is often a direct consequence of the model class

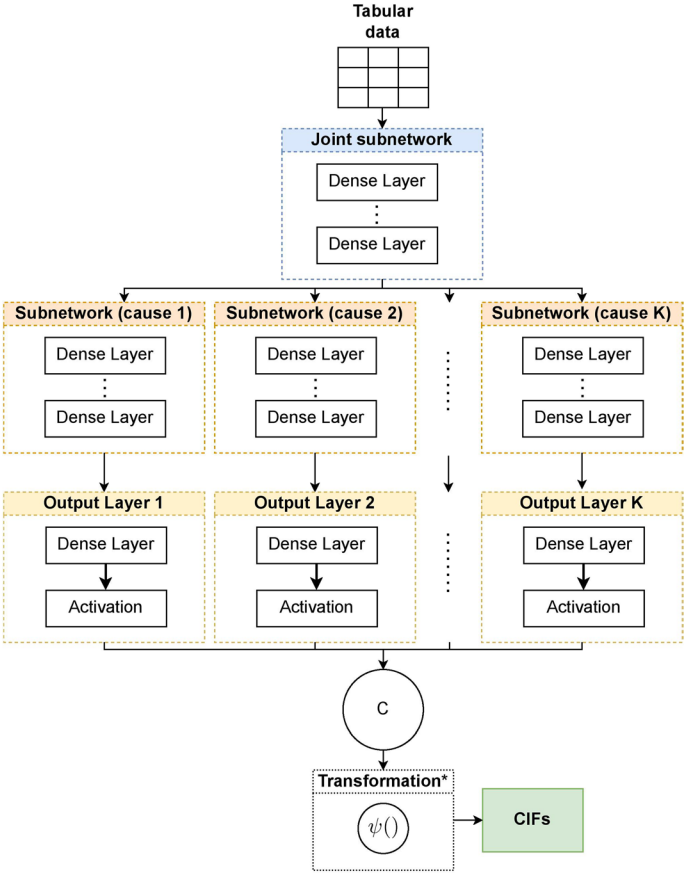

Fig. 2 Schematic neural architecture for competing risks in survival analysis using shared and cause-specific subnetworks. $\psi()$ transforms the model output (e.g., hazard rate) to the final outcome [e.g., cumulative incidence functions (CIFs)]

(i.e., its negative log-likelihood). However, as is common in DL, some methods employ multiple losses for improved performance or multi-task learning. For instance, some DL-based survival methods compute a ranking loss, in addition to a standard survival loss, for improvement of the C-index performance measure. The final loss is usually computed as the (weighted) average of all losses applied. Third, the *parametrization* determines which model component is being parametrized by an NN. The standard model parametrization is usually implied by the model class.

Almost all modern DL-based survival methods are optimized with gradient-based methods, featuring tractable loss functions yet with many parameters to be optimized. Optimizing the loss function in a batch-wise manner, which is the common approach in DL, is not always feasible, though. This holds for Cox-based methods because the partial loss (7) depends on the complete risk set. Recently, Kvamme and Borgan (2019) showed that Cox-based methods can be optimized with stochastic gradient descent methods (i.e., batch-wise) if the batch size is sufficiently large to non-parametrically

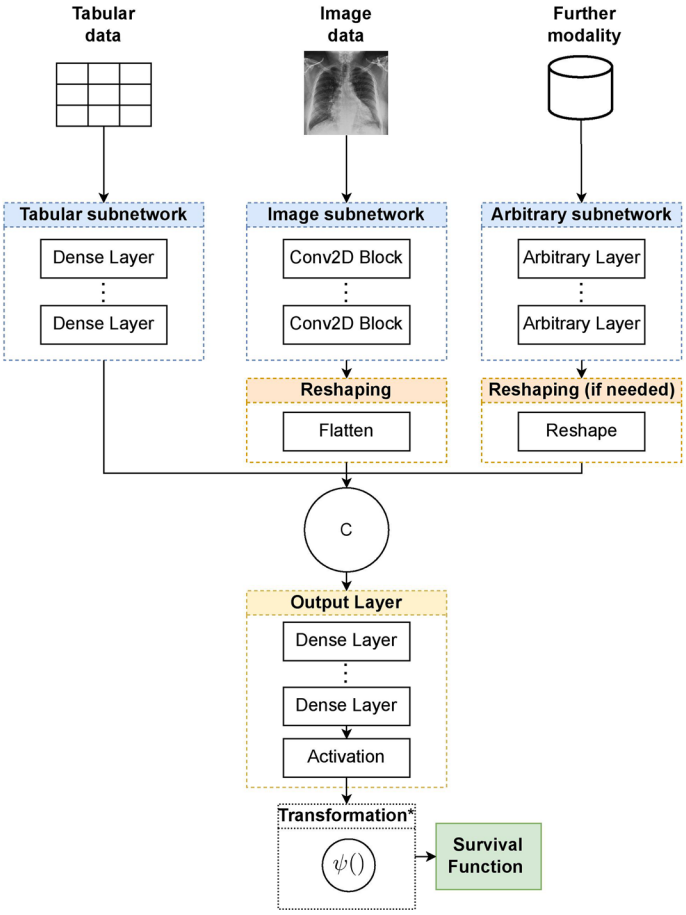

Fig. 3 Schematic neural architecture for multimodal data input in survival analysis using separate subnetworks for all modalities. Their outputs are reshaped and concatenated to align dimensions. $\psi()$ transforms the model output to the final outcome. The X-ray scan is obtained from Irvin et al. (2019)

approximate the risk set. Before that, deep Cox-based methods were optimized with full gradient descent, making them less attractive for computationally expensive tasks.

We now give a detailed description of the above estimation-related concepts as well as their interrelation for all methods reviewed.

3.3.1.1 Cox-based methods Out of the 61 methods included in this review, 26 methods are Cox-based; that is, these methods are essentially DL-based modifications and extensions of the Cox regression model. This is underlined by the fact that all of them parametrize the hazard rate—more precisely, the log-risk function $g(\mathbf{x})$ in (6)—by an NN and minimize the (sometimes slightly modified) Cox loss, i.e., the (negative logarithm of the) partial likelihood of the Cox model.

DeepSurv by Katzman (2018) extends Faraggi and Simon (1995) by using a deep FFNN as well as different non-linear hidden layer activation functions. The model by Faraggi and Simon (1995) is a simple special case of *DeepSurv*, with a single hidden layer with logistic

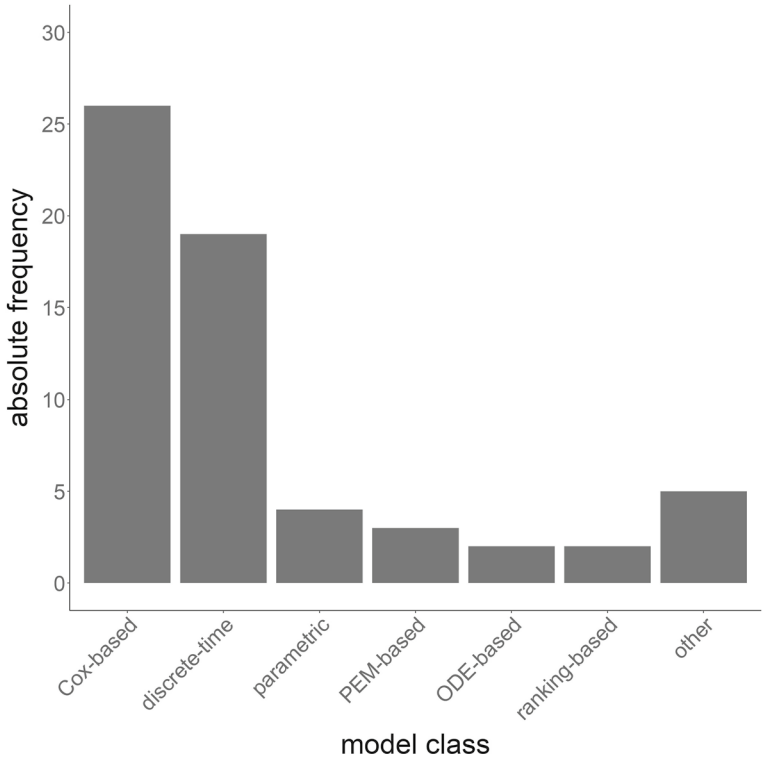

Fig. 4 Absolute frequencies of model classes among all 61 methods reviewed

activation and identity output activation. Note that the PH assumption induced by the Cox PH regression model is maintained in *DeepSurv*, as $g(\mathbf{x})$ remains time-constant despite being parametrized by a (deep) NN. *DeepSurv* uses stochastic gradient descent (SGD) for optimization. To do so, *DeepSurv* uses a restricted risk set including only individuals in the current batch since the Cox loss originally sums over the *entire* risk set, which would impede batching. *Cox-Time* (Kvamme et al. 2019) is a more flexible extension of *DeepSurv* where a time-dependent predictor allows estimation of TVEs, i.e. $h(t|\mathbf{x}) = h_0(t)\exp(g(\mathbf{x}, t))$. However, this increased flexibility would render the batching strategy as applied by *DeepSurv* (and most other PH-restricted Cox-based methods) computationally expensive. Therefore, the *Cox-Time* loss function is modified to approximate the risk set by a sufficiently large subset of all individuals at risk, which enables mini-batching and thus scalability to large datasets. *NN-DeepSurv* (Tong and Zhao 2022) is another extension of *DeepSurv*, employing a nuclear norm for imputation of missing features.

More than half of all Cox-based methods (14) focus on the applicability to high-dimensional data, usually omics data. *MCAP* (Chai et al. 2022) and *VAECox* (Kim et al. 2020) both use multiple losses, the latter one within a transfer learning approach. *Cox-nnet* (Ching et al. 2018), *Cox-PASNet* (Hao et al. 2018) and its multimodal extension *PAGE-Net* (Hao et al. 2019), *GDP* (Xie 2019), *DNNSurv_Sun2020* (Sun et al. 2020), *Qiu2020* (Qiu et al. 2020), *DeepOmix* (Zhao 2021), and *CNT* (Fan et al. 2022) use simple FFNNs and only a single Cox loss, thus being very similar to *DeepSurv* and *Cox-Time*. *SALMON*

(Huang 2019) and *CNN-Cox* (Yin et al. 2022) distinguish themselves through their architecture (see Sect. 3.3.2), *Haa2019* (Haarburger et al. 2019) and *ConcatAE/CrossAE* (Tong et al. 2020) through additionally being multimodal (see below).

Eight Cox-based methods focus on unstructured or multimodal input (see also Sect. 3.4.2). *WideAndDeep* (Pölsterl et al. 2020) combines a linear predictor of tabular features (wide part) with a 1D embedding $d(\mathbf{z})$ learned from a point cloud, which is a latent representation learned from 3D shapes (deep part); subsequently both parts are fused by linearly aggregating the learned weights as in (9). The model uses the *DeepSurv* loss and thus preserves the PH assumption. *Haa2019* employs a pre-trained CNN of type ResNet18 for subsequent fine-tuning on CT scans, using a Cox loss. Both *DeepConvSurv* (Zhu et al. 2016) and *CapSurv* (Tang et al. 2019) can learn from image data—yet without incorporating structured (tabular) data—by using CNN and CapsNet architectures (see Sect. 3.3.2), respectively. *DeepConvSurv* uses a single Cox loss, while *CapSurv* additionally employs the CapsNet margin and reconstruction losses. Both *ConcatAE/CrossAE* and *PAGE-Net* can process high-dimensional data as well as multimodal data; *ConcatAE/CrossAE* use classification and reconstruction losses in addition to the Cox loss to do so, while *PAGE-Net* introduces biologically interpretable pathology, genome-, and a demography-specific layers. *Xie2021* (Xie and Yu 2021) can learn from unstructured data for cure rate classification. *DAFT* (Wolf et al. 2022) employs CNNs and a single Cox loss to learn from both structured and unstructured data.

SurvNet (Wang et al. 2021) and *DCM* (Nagpal et al. 2021c) do not accommodate any of the additional outcome types or feature-related aspects defined above (see also Sect. 3.4), yet they use multiple losses. In addition to a Cox regression module, *SurvNet* employs an input construction module and a survival classification module (with corresponding losses) for handling missing values and high- versus low-risk profile classification, respectively. *DCM* employs an approximate Monte Carlo Expectation Maximization (EM) algorithm for the estimation of a mixture of Cox models, the total loss also including an Evidence Lower Bound (ELBO) component. *ELMCoxBAR* (Wang and Li 2019) and *San2020* (Sansaengtham et al. 2020) are standard Cox-based methods in terms of estimation, their architectures being extensions of FFNNs (see Sect. 3.3.2).

3.3.1.2 Discrete-time methods Another 19 methods can be categorized as discrete-time approaches. They consider time to be discrete and usually employ classification techniques, with the outcome being binary event indicators for each discrete time point or interval. The standard loss function of discrete-time DL-based survival methods is the negative log-likelihood (NLL), while typically the discrete hazard (2) is parametrized by an NN—just like in the early *PLANN* model. However, as compared to the Cox-based methods which are rather homogeneous methodologically, discrete-time methods are much more heterogeneous in terms of loss functions and architecture.

DeepHit (Lee et al. 2018) is a discrete-time DL-based survival method. It aims to learn first-hitting times directly by not making any assumptions about the underlying stochastic process and parametrizing the discrete PMF directly. *DeepHit* combines two loss functions: first, the log-likelihood derived from the joint distribution of first hitting time and the corresponding event, adjusted for right-censoring and taking into account competing risks; and second, a combination of ranking losses. *Dynamic-DeepHit* (Lee et al. 2019), an RNN-based extension of *DeepHit* which can handle longitudinal input data and thus TVFs, additionally employs a so-called prediction loss for the auxiliary task of step-ahead prediction of TVFs. The transformer-based *TransformerJM* (Lin and Luo 2022) also parametrizes

the PMF, focusing on modeling survival data and longitudinal data jointly and training on a combination of NLL- and MSE-based losses.

RNN-SURV (Giunchiglia et al. 2018) uses both features and time as inputs, and outputs the survival probability at each discrete time point, employing RNN architecture to use information from previous time points to inform prediction of subsequent time points; the model combines the estimated survival probabilities to a risk score via a weighted sum and employs both an NLL loss (based on the survival probabilities) and a C-index-based (ranking) loss (based on the risk score) for model training.

Nnet-survival (Gensheimer and Narasimhan 2019) parametrizes the discrete hazard (8) by an NN, using an NLL loss as well as mini-batch SGD for rapid convergence and scalability to large datasets. Mini-batch SGD is easily applicable to discrete-time methods because the loss only depends on individuals in the current mini-batch. The specific architecture—in particular, the number of neurons per hidden layer and the connectedness of layers—determines whether TVEs can be modeled or whether the PH restriction is upheld. Another four methods—*CNN-Survival* (Zhang 2020), *MultiSurv* (Vale-Silva and Rohr 2021), *SurvCNN* (Kalakoti et al. 2021), and *Tho2022* (Thorsen-Meyer 2022)—use the same loss and parametrization as *Nnet-survival*. *CNN-Survival* uses a CNN along with transfer learning to learn from CT data (without incorporating tabular data). The multimodal *MultiSurv* first extracts feature representations for each data modality separately, then fuses them, and finally outputs predictions of conditional survival probabilities. *SurvCNN* creates an image representation of multiple omics data types using CNNs and can combine this with clinical data for prediction. *Tho2022* can embed data from multiple modalities, such as electronic health records, and feeds these embedded representations into an RNN which in turn produces survival predictions.

The competing-risk and recurrent-event method *CRESA* (Gupta et al. 2019) is an RNN-based approach that parametrizes the discrete hazard and uses a loss based on recurrent cumulative incidence functions, which also contains a ranking component as in *DeepHit*. *DRSA* (Ren 2019) also employs an RNN and also parametrizes the discrete hazard, yet as compared to *CRESA* it uses multiple log-likelihood-based losses to predict the likelihood of uncensored events as well as survival rates for censored cases. *Kam2021* (Kamran and Wiens 2021) uses the same network architecture as *DRSA*, but proposes a novel training scheme to directly estimate the survival probability: a combination of Rank Probability Score (RPS) loss, emphasizing calibration, and a kernel loss, emphasizing discrimination through penalization of wrongly ordered uncensored individuals. *DCS* (Fuhlert et al. 2022) extends the architecture from *DRSA* and then produces survival probability estimates by employing the same loss function from *Kam2021*, yet modifying the kernel loss component by not only comparing uncensored-uncensored pairs.

N-MTLR (Fotso 2018) builds upon MTLR and parametrizes the corresponding logistic regression parameters. *DNNSurv_Zha2019* (Zhao and Feng 2019) first computes individual-level pseudo (conditional) probabilities, defined as the difference between the estimated survival function with and without individual i and computed on a regular grid of time points, thus reducing the survival task to a regression task, and consequently uses a standard regression loss. *su-DeepBTS* (Lee et al. 2020) discretizes the time axis but then uses a Cox loss for each time interval, summing up the losses across intervals. *DeepComp* (Li et al. 2020) combines distinct losses for censored and uncensored observations with an additional penalty. *SSMTL* (Chi et al. 2021) transforms the survival task into a multitask setting with binary outcome for all time points (or multi-class in case of competing risks), then predicting survival probabilities for each of the time points. *SSMTL* also employs a custom loss made up of a classification loss for uncensored data, a so-called

semi-supervised loss for censored data, regularization losses (L1 and L2) as well as a ranking loss in order to ensure monotonicity of predicted survival probabilities.

Hu2021 (Hu et al. 2021), a transformer-based method, uses an entropy-based loss as well as a discordant-pair penalization loss, parametrizing the discrete hazard. *SurvTRACE* (Wang and Sun 2022), another transformer-based method, also parametrizes the discrete hazard, but additionally performs two auxiliary tasks on the survival data: classification and regression; accordingly, the final model loss is a combination of a *PC-Hazard* survival loss (see below), an entropy-based classification loss, as well as a Mean Squared Error (MSE)-based regression loss.

3.3.1.3 Parametric methods The two methods *DeepWeiSurv* (Bennis et al. 2020) and *DPWTE* (Bennis et al. 2021)—the latter one building on the former—are Weibull-based deep survival methods. Neither of them addresses any of the outcome types or feature-related aspects presented in Sect. 2.2. *DeepWeiSurv* parametrizes a mixture of Weibull models, as well as both Weibull distribution parameters (see (4) with θ_1, θ_2 the scale and shape parameters of the Weibull distribution), by an FFNN and uses an NLL-based loss function. *DPWTE* employs classification and regression subnetworks to learn an optimal mixture of Weibull distributions, using the same loss function as *DeepWeiSurv* with additional sparsity regularization with respect to the number of mixtures. *Ava2020* (Avati et al. 2020) parametrizes the parameters of a log-normal distribution, while being flexible in terms of model architecture. The method introduces the Survival-CRPS loss, a survival adaptation of the Continuous Ranked Probability Score (CRPS). This loss results in well-calibrated survival probabilities and furthermore provides the flexibility to handle both right- and interval-censored data. *DSM* (Nagpal et al. 2021a) is a hierarchical generative model based on a finite mixture of parametric primitive distributions similar to the well-known approach by Ranganath et al. (2016), using a (mixture) likelihood-based loss as well as an additive loss based on ELBO for uncensored and censored observations; the choice of the parametric survival distribution—either Weibull or Log-Normal—is a hyperparameter and can thus be tuned. Its RNN-based extension *RDSM* (Nagpal et al. 2021b), is furthermore capable of handling TVFs.

3.3.1.4 PEM-based methods Three methods rely on the PEM framework to develop a deep survival approach. *PC-Hazard* (Kvamme and Borgan 2021) addresses the right-censored single-risk survival task by parametrizing the hazard rate through an FFNN and using the standard likelihood-based PEM loss. Support for other outcome types or feature-related aspects, as introduced in Sects. 2.2.1 and 2.2.2, is not discussed. Similarly, *DeepPAMM* (Kopper et al. 2021, 2022) uses a penalized Poisson NLL as a loss function and also parametrizes the hazard rate by an NN. This method combines a Piecewise Exponential Additive Mixed Model (PAMM; Bender et al. 2018) with Semi-structured Deep Distributional Regression (Rügamer et al. 2023), which embeds the structured predictor in an NN and further learns from other (unstructured) data types [see (9)].

Finally, *IDNetwork* (Cottin et al. 2022) implements an illness-death model, which uses a PEM-based approach to estimate probabilities for transitions between different states and utilizes FFNNs with shared and transition-specific subnetworks. *IDNetwork* then uses a penalized NLL loss based on the transition probabilities.

3.3.1.5 ODE-based methods *DeepCompete* (Aastha et al. 2021) consists of an FFNN shared across all risks as well as an FFNN and a neural ordinary differential equation (ODE)

block for each specific risk, using an NLL-based loss. *survNODE* (Groha et al. 2021) is based on a Markov process and aims to directly solve the Kolmogorov forward equations by using neural ODEs to achieve flexible multi-state survival modeling, with the transition rates parametrized by a nODE architecture (see Sects. 3.2 and 3.3.2).

3.3.1.6 Ranking-based methods As can be seen in the section above, multiple discrete-time methods (*DeepHit*, *CRESA*, *DCS*, *Kam2021*, *RNN-Surv*, *SSCNN*, and *SSMTL*) use ranking losses as auxiliary losses. Beyond that, there are two continuous-time methods—*RankDeepSurv* and *SSCNN*—that are built upon ranking losses. Here, we refer to these continuous-time ranking loss-based methods simply as ranking-based methods. *RankDeepSurv* (Jing 2019) combines ranking losses with an extended MSE loss to augment the number of training samples, without advanced NN architecture or handling of non-standard survival data modalities. *SSCNN* (Agarwal et al. 2021b) is a multimodal method that reduces histopathology images to whole slide feature maps and uses them, in addition to clinical features, as input of a Siamese Survival CNN; model training with a custom loss—a combination of a ranking loss with a loss to improve model convergence and pairwise differentiation between survival predictions—is built directly on the outputs of the Siamese NN.

3.3.1.7 Other methods As for the remaining five methods, *DASA* (Nezhad et al. 2019) is a framework introducing a novel sampling strategy based on DL and active learning. The GAN-based *DATE* (Chapfuwa et al. 2018) seeks to learn the event time distribution non-parametrically by using adversarial learning and a custom loss function made up of an uncensored-data component, a censored-data component, as well as a distortion loss component. *Hua2018* (Huang et al. 2018) uses a CNN architecture and correlational layers for multimodal learning to produce person-specific risks, which are then directly fed into a smooth C-index loss function for model training. *Aus2021* (Ausset et al. 2021) employs normalizing flows in order to estimate the density of time-to-event data and predict individual survival curves via a transformation model, using an NLL-based loss augmented by an intermediary loss for regularization. Finally, *rcICQRNN* (Qin et al. 2022) is a deep survival method based on a quantile regression NN, parametrizing the quantile regression coefficients by means of an FFNN and using an inverse-probability-of-censoring weighted log-linear quantile regression loss.

3.3.2 Network architecture

Most DL-based survival methods in this review use FFNNs, often in combination with some other, more advanced architecture. Still, 20 methods—as well as all early DL-based methods such as the one by Faraggi and Simon (1995)—exclusively rely on FFNNs. Still, architectural choices among these FFNN-based methods differ. For instance, *DeepHit* uses a softmax outcome layer to produce survival probabilities for each discrete time point and, thus, to model the PMF.

Out of a total of 10 CNN-based methods in this review, eight are multimodal methods that can work with image data: *DeepConvSurv*, *Hua2018*, *Haa2019*, *CNN-Survival*, *PAGE-Net*, *SSCNN*, *Xie2021*, and *DAFT*. For instance, *Hua2018* employs CNN and FFNN subnetworks, along with correlational layers, in order to learn from both pathological images and molecular profiles. The CNN-based method *SurvCNN* is not multimodal per se, but transforms high-dimensional omics data into an image

representation in order to feed them into a CNN. *CNN-Cox* combines cascaded Wx (Shin 2019), an NN-based algorithm selecting features based on how well they distinguish between high- and low-risk groups, with a 1D CNN architecture applied to gene expression data. Note that the choice of architecture for CNN- and AE-based methods is usually motivated by the objective of extracting information from data input (e.g., from images via CNNs or from omics data via AEs with auxiliary losses), without being very relevant to the target of estimation. This is in contrast to, e.g., RNNs, where the architecture choice is driven by the learning objective.

Nine methods reviewed here use RNN architectures. Six of them—*RNN-Surv*, *CRESA*, *DRSA*, *Kam2021*, *DCS*, and *Tho2022*—use a Long Short-Term Memory (LSTM), while the remaining one, *DeepComp*, does not state the RNN architecture it employs. Out of these methods, *RNN-Surv*, *DRSA*, *Kam2021*, and *DCS* do not go beyond the setting of single-risk, right-censored tabular data. For example, *RNN-Surv* uses the RNN to carry forward information from previous time steps, employing a sigmoid output layer activation. *Tho2022* employs the RNN architecture for multimodal learning from text, medical history, and high-frequency data, while *DeepComp* uses it for competing risk modeling. *CRESA* models both recurrent events and competing risks by means of its RNN architecture. The final two RNN-based methods, *Dynamic-DeepHit* and *RDSM*, are actually extensions of the simpler FFNN-based methods *DeepHit* and *DSM*, respectively, enabling the incorporation of TVFs.

Four methods—*DASA*, *DCM*, *ConcatAE/CrossAE*, and *VAECox*—use some form of AEs. Another four methods—*Nnet-survival*, *Ava2020*, *MultiSurv*, and *Deep-PAMM*—do not require a specific architecture, which can instead be flexibly chosen based on application requirements; for instance, a CNN for handling image data (as in *MultiSurv*) or an RNN for incorporating TVFs (as in *Ava2020*). Three recent methods, *Hu2021*, *SurvTRACE*, and *TransformerJM*, use a transformer architecture, while another two novel methods, *DeepCompete* and *survNode*, use a nODE architecture.

Only a single method, *DATE*, uses a GAN architecture (along with a custom loss). *ElmCoxBAR* uses an Extreme Learning Machine (ELM) architecture, which is similar to an FFNN but does not require backpropagation for optimization. *SALMON*, *San2020*, *DPWTE*, and *SurvNet* all use FFNNs, but in a modified manner. *SALMON* adds so-called eigengene modules, using eigengene matrices of gene co-expression modules (Zhang and Huang 2014) instead of raw gene expression data as NN input. *San2020* uses a Stacked Generalization Ensemble Neural Network (Wolpert 1992), which takes a combination of *DeepSurv* sub-models and concatenates them for improved hazard prediction. *DPWTE* adds a Sparse Weibull Mixture (SWM) layer to learn the optimal number of Weibull distributions for the mixture model, through an element-wise multiplication of its weights by the previous layer's output. *SurvNet* adds a context-gating mechanism, which is similar to the attention mechanism used in transformers, by adjusting log hazard ratios by survival probabilities from the survival classification module. *WideAndDeep* employs a PointNet (Qi et al. 2017) architecture to learn a latent representation of 3D shapes of the human brain while additionally learning from regular tabular data, subsequently fusing both parts. *CapSurv* modifies the CapsNet architecture (Sabour et al. 2017), developed for image classification, by adding a Cox loss and thus making it amenable to SA tasks.

Figure 5 depicts the absolute frequencies of NN architectures among all 61 methods included in this review.

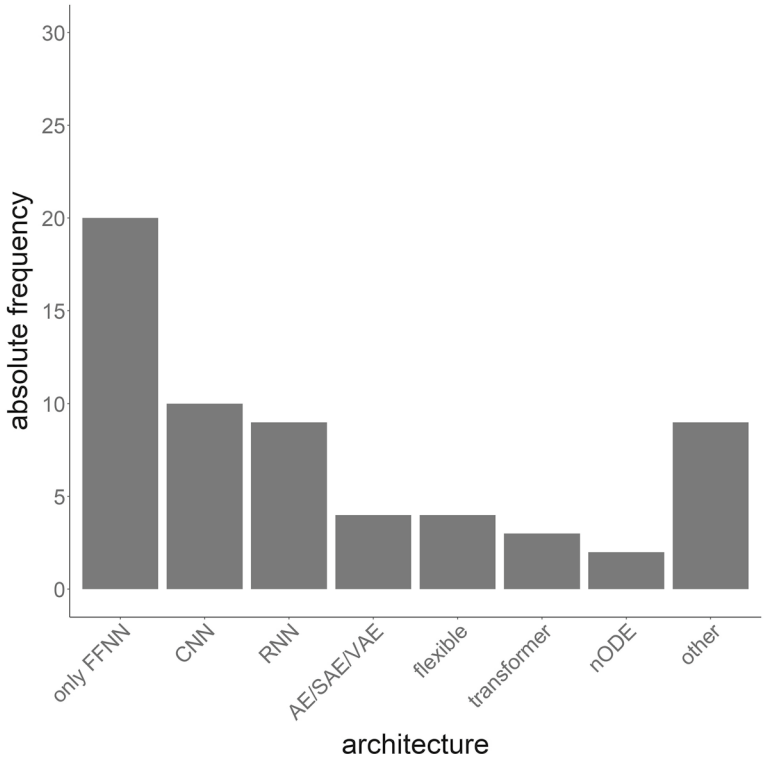

Fig. 5 Absolute frequencies of neural network architectures among all 61 methods reviewed

3.4 Supported survival tasks

In this section, we discuss which methods can handle the data-related aspects introduced in Sect. 2.2. We start by considering *outcome* types and subsequently *feature*-related aspects. Finally, we summarize which methods offer (which kind of) interpretability of results.

3.4.1 Supported outcome types

Regarding censoring and truncation of event times, left-censoring and right-truncation are not explicitly addressed by any of the methods reviewed. *Ava2020* is capable of handling interval-censored data thanks to the flexibility of the Survival-CRPS loss. *survNode* briefly addresses interval-censoring and left-truncation by stating how they would affect likelihood computations. *DSM* mentions that the modeling framework is amenable to these two output modalities. In *DeepPAMM* left-truncation is accounted for in the data pre-processing step.

Nine methods are designed to deal with competing risks; interestingly, none of these methods is Cox-based, and four of them are discrete-time. *DeepHit*, *CRESA*, and *DeepComp* all assume time to be discrete and employ cause-specific subnetworks, with *DeepHit* using FFNNs to generate a final distribution over all competing causes for each

individual; both *CRESA* and *DeepComp* use RNN architectures, yet while *CRESA* also generates a final distribution over all competing causes, *DeepComp* outputs cause-specific discrete hazards for each time interval. *SSMTL*, also discrete-time, uses an FFNN architecture, views competing risk SA as a multiclass problem, and creates a custom loss with separate components for non-censored and censored individuals, as well as a ranking component. *DeepCompete* is a continuous-time method that employs nODE blocks within each of its cause-specific subnetworks in order to output a cumulative hazard function. *DSM* first learns a common representation of all competing risks by passing through a single FFNN. Based on this representation, and treating all other events as censoring, the event distribution for a single risk is then learned using cause-specific Maximum Likelihood Estimation (MLE); the ELBO loss is also adjusted to treat competing events as censoring. Both *survNode* and *IDNetwork* are based on Markov processes—illness-death process and Markov jump process, respectively—and thus naturally handle competing risks and even the more general case of multi-state outcomes. Being PEM-based, *DeepPAMM* parametrizes the hazard rate, which is a transition rate by definition; *DeepPAMM* can further specify multiple transitions and therefore model competing risks as well as multi-state outcomes. Finally, two methods discuss handling of recurrent events: *CRESA* employs an RNN architecture with time steps representing recurrent events, while *DeepPAMM* uses random effects inspired by statistical mixed models. Figure 6 summarizes which outcome types beyond right-censoring the methods reviewed explicitly mention.

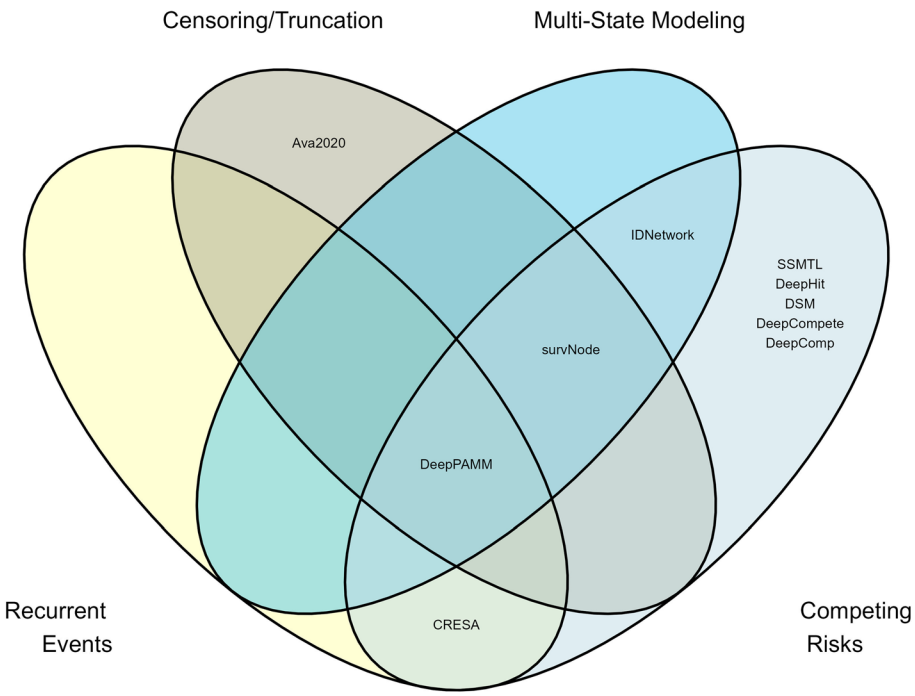

Fig. 6 Venn diagram illustrating which methods can handle the distinct survival outcome types

3.4.2 Supported feature-related aspects

One important feature-related aspect is time dependence, a deviation from the PH assumption imposed by traditional survival models such as Cox regression or Weibull AFT. Seven methods address TVFs: *DeepHit*'s and *DSM*'s RNN-based extensions *Dynamic-DeepHit* and *RDSM*, as well as *CRESA*, *Ava2020* (by choosing an RNN architecture), *survNode*, *DeepPAMM*, and *TransformerJM*. The technical incorporation of TVFs is, for example, achieved by converting tabular time-varying feature input into long format (*DeepPAMM*) or by employing RNNs prior to each new feature measurement (*survNode*).

TVEs constitute another deviation from the PH assumption: Seven methods are capable of modeling effects that might not be constant over time, with four of them being time-discrete approaches. *Nnet-survival* and *MultiSurv* incorporate TVEs modeling by using a fully connected NN to connect the final hidden layer's neurons with the output nodes, while *RNN-Surv* captures TVEs through its RNN architecture. *Cox-Time* accommodates TVEs by making the Cox-style relative risk—which it parametrizes by an NN—time-dependent and *DeepPAMM* can address TVEs through the interaction of the follow-up time (represented as a feature) with other features. *DSM* and *SSMTL* do not provide further detail about how TVEs are being estimated.

Another feature-related aspect is the integrability of high-dimensional (usually omics) data, which implies learning from a high-dimensional predictor space. While all DL-based methods are generally capable of handling high-dimensional feature inputs, here we focus on the 18 DL-based survival methods that are explicitly designed to work with high-dimensional data, usually by applying specialized regularization techniques. 14 of these methods—*Cox-nnet*, *Cox-PASNet* and *PAGE-Net*, *Haa2019*, *GDP*, *SALMON*, *ConcatAE/CrossAE*, *DNNSurv_Sun2020*, *Qiu2020*, *VAECox*, *DeepOmix*, *CNN-Cox*, *CNT*, and *MCAP*—are (partially) Cox-based. As for the remaining four methods, *CNN-Survival*, *MultiSurv*, and *SurvCNN* are discrete-time methods, while *rcICQRNN* is quantile regression-based.

Finally, a total of 16 methods can (hypothetically) extract information from unstructured or multimodal features. Eight of them are (partially) CNN-based, underlining the focus on processing mostly medical image data. *DeepConvSurv*, *CapSurv*, and *CNN-Survival* (the last one employing transfer learning) exclusively work with imaging data without incorporating any tabular information, which is why these methods are not truly *multi*modal. Similarly, *Nnet-survival*, being flexible in terms of NN architecture, can learn from image data by choosing a CNN, yet again at the cost of discarding tabular data as only a single data modality can be handled. *Hua2018* incorporates both image and molecular data yet without making any mention of tabular data. *Haa2019* fine-tunes a pre-trained ResNet18, optionally concatenating it with radiomics features, and additionally leverages clinical data.

PAGE-Net employs a novel patch aggregation strategy to integrate unstructured Whole Slide Images (WSIs) and structured demographic and genomic data. *SSCNN* creates feature maps from WSIs and employs a Siamese CNN to learn from both these feature maps as well as clinical features. Liu and Kurc (2022) also use DL to extract features from WSIs in the context of survival analysis, however, not in an end-to-end approach within the network.

ConcatAE/CrossAE integrates information from multiple modalities, either through modality-specific autoencoders or cross-modality translation; the integration of tabular

data is, however, not explicitly mentioned. *survNode* can conceptually account for multimodal features by encoding initial values with, e.g., CNN or NLP layers. The cure rate model *Xie2021* only allows for (single-modality) unstructured data for determining the cure rate probability through a CNN. *DAFT* uses a ResNet CNN architecture as its backbone, feeding tabular data into it through a novel Dynamic Affine Feature Map Transform (DAFT) module, which in turn enables a bidirectional information flow between image and tabular data. Finally, *Tho2022* employs an RNN architecture to create an embedding for electronic patient record data (such as medical history and free text) and further fuses tabular clinical features into the model before generating survival predictions. *WideAndDeep*, using a Alzheimer's Disease (AD) dataset, learns a latent representation of 3D shapes of the human brain while additionally learning from regular tabular data, subsequently fusing both parts. *MultiSurv*, a multimodal extension of *Nnet-survival*, and *DeepPAMM* both provide flexibility in terms of architecture choice so that, for example, image data could be incorporated by employing CNNs for the NN part; they also fuse information from the different data modalities.

Figure 7 illustrates which of the methods incorporate the different types of feature-related aspects.

3.5 Interpretability

By construction, DL methods (as well as ML methods) are more complex than the survival models considered in Sect. 2.3 and thus usually do not provide the same degree of

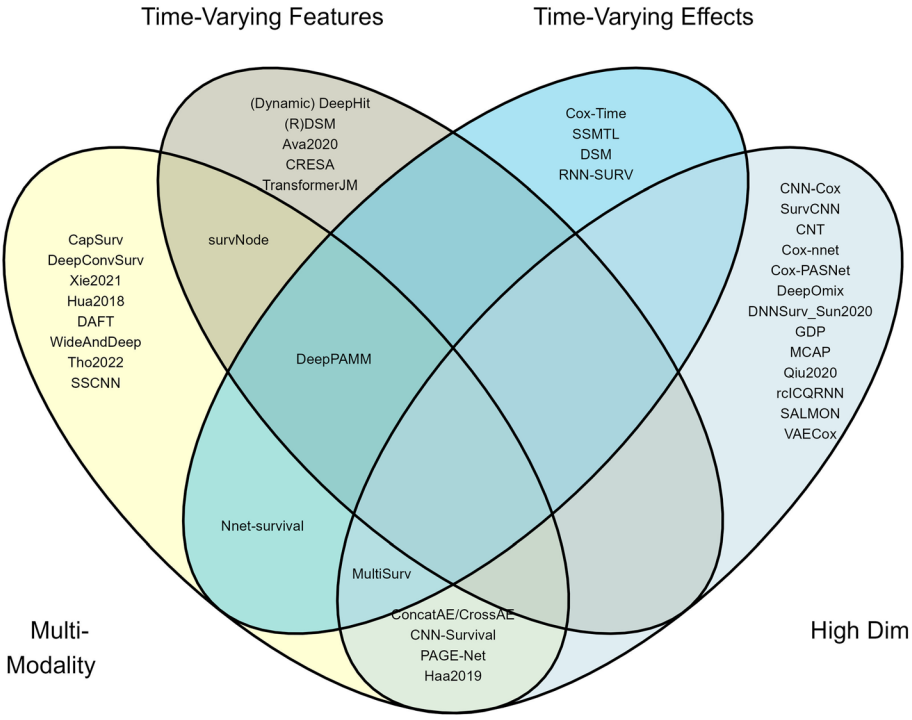

Fig. 7 Venn diagram illustrating which methods can handle the distinct survival feature-related aspects

interpretability. At the same time, in fields such as the life sciences, results and model outputs must be interpretable to provide a solid basis for highly sensitive decision-making (Vellido 2020). Here, we summarize which of the methods provide (inherently) interpretable results.

Cox-nnet, *Cox-PASNet*, *PAGE-Net* and *DeepOmix* provide some interpretability by assigning biological meaning to the nodes of their NNs. *Cox-nnet* obtains biologically relevant hidden nodes, as the most variable nodes can be viewed as surrogate features for discriminating patient survival and, in addition, these nodes correlate strongly with significantly enriched pathways. *Cox-PASNet* and *PAGE-Net* both possess genome-specific layers, which include a gene input layer, a pathway layer embedding prior pathway-related knowledge for biological interpretation, and data integration layers. The two methods then rank the node values of features by the average absolute partial derivatives (with respect to the data integration layers) for a pathway-based interpretation of genomic data: explicitly, pathway nodes each represent a biological pathway. *PAGE-Net* additionally possesses pathology-specific layers, which identify features relevant to SA from histopathological images via pre-trained CNNs; at the patient-level, these survival-discriminatory features eventually represent a histopathological WSI. *DeepOmix* incorporates multi-omics data via a gene input layer and prior biological and pathway knowledge via functional module layers (low-dimensional representations), guided by the idea that genes do not work in isolation but rather function as functional modules. With each node representing a non-linear function of the genes' different attributes (e.g., mutations), *DeepOmix* obtains biological interpretability because it captures the (non-linear) effects of biological pathways onto survival time.

By fusing the output of an NN for image data with the output of a Cox PH model for tabular data, *WideAndDeep* retains the interpretability of a standard Cox regression for structured features. *Xie2021* also provides standard Cox model interpretability, because survival prediction is performed through non-deep Cox regression. *DeepPAMM* provides classical statistical interpretability of the structured effects by its architecture, with identifiability, if necessary, ensured through orthogonalization (Rügamer 2023). *survNode* introduces a latent variable extension providing aspects of feature interpretability. The transformer-based *SurvTrace* method makes use of attention maps, comparing attention scores of different features across selected individuals to provide some interpretability of feature effects.

It is worth noting that post-hoc methods from the field of Interpretable Machine Learning and explainable AI, such as Permutation Feature Importance (Breiman 2001), Local Interpretable Model-agnostic Explanations (LIME Ribeiro et al. 2016), Shapley Additive exPlanations (SHAP Lundberg and Lee 2017), attention maps (Jetley et al. 2018), Layer-Wise Relevance Propagation (LRP Montavon et al. 2019), and Neural Additive Models (NAMs Agarwal 2021a), are potentially applicable to DL-based survival methods. However, this is subject to current research and it is not always clear if and how such methods need to be adjusted to account for different types of censoring, truncation or other outcome types.

Several survival-specific adaptations of the abovementioned post-hoc interpretability methods have already been developed; for instance, *SurvLIME* (Kovalev et al. 2020), *SurvSHAP(t)* (Krzyziński et al. 2022), and *SurvNAM* (Utkin et al. 2022) are based on *LIME*, *SHAP*, and *NAMs*, respectively. Cho et al. (2023) use meta-learning and the *DeepLIFT* (Shrikumar et al. 2017) method to make the integration of multi-omics data in SA more interpretable.

Among the papers reviewed here, *SALMON* explores feature importance of individual inputs, *DNNSurv_Sun2020* employs LIME, *Tho2022* uses SHAP, and *SSMTL* computes post-hoc feature importance and plots feature effects on cumulative incidence curves. *Qiu2020* uses a risk propagation technique called *SurvivalNet* (Yousefi 2017), which is an explanation method specific to SA.

3.6 Model evaluation and comparison

Model evaluation is an important aspect of any machine learning pipeline, and SA in particular. Typical metrics in benchmark experiments of survival models are the C-index [usually Harrell's et al. (1982) or Uno's et al. (2007)] for assessing risk predictions, and the Brier/Graf score (Graf et al. 1999) for evaluation of distribution predictions, with the C-index being by far the most popular metric among the methods reviewed here. Typically underreported are the right-censored logloss (Avati et al. 2020) and calibration measures such as D-calibration (Haider et al. 2020). Recent work also suggests that most of the previously used evaluation measures in SA do not constitute proper scoring rules (Sonabend 2022). Proper alternatives have been proposed recently (Rindt et al. 2022; Sonabend 2022), but have not been widely adopted yet.

Interpretation and comparison of the self-reported benchmark experiments in different articles is often not meaningful for various reasons: The datasets used, their pre-processing, and handling of missing values is not the same. Even if the same data sets are used, the definition of resampling strategy, the exact definition of the respective evaluation metrics (e.g. different variants of C-Index, integration window of the integrated brier score, etc.) and their use [e.g. transformation of survival distribution predictions for measures of discrimination (Sonabend et al. 2022)] are often not clearly specified or not identical. Further general issues that hinder direct interpretation or reported results are potential issues of selective reporting and researchers degrees of freedom (selection of data sets, choice of evaluation metrics, decisions about budget and hyperparameter space for tuning of the proposed as well as competing algorithms, etc.) that have plagued applied sciences but have also been bemoaned in methodological research (e.g. Boulesteix et al. 2020; Nießl et al. 2022).

For all these reasons, direct comparison of the performance of different methods reviewed in this article is not possible. This calls for future research to conduct neutral benchmark studies (Boulesteix et al. 2013). Such an investigation has been for example conducted for some non-DL-based ML methods on omics data (Herrmann et al. 2020). However, such studies are generally hard to conduct and require substantial effort, in particular for DL-based methods with high computational requirements, and because general purpose implementations of most of the methods reviewed here are not available and code repositories are missing for almost half of the methods (cf. Sect. 3.8).

3.7 Sample size requirements

Sample size considerations are an equally important topic that needs further research in the context of DL-based survival analysis.

In general, the sample size required for training a DL-based method crucially depends on the model architecture (such as the choice of network architecture and hyperparameters or the use of transfer learning) as well as on the input data modalities (e.g., whether images or high-dimensional omics data are being used), and the assumed data generating

process. In addition, sample size calculations are very task-specific: Fang (2021) show that the required sample size for organ auto-segmentation critically depends on the organ to be segmented. Overall, sample size calculation in DL is still quite rare, being an active field of research itself (Shahinfar et al. 2020; Fang 2021). For instance, in ML-based medical imaging analysis, a systematic review of methodologies for sample size calculation by Balki (2019) identified only four such methods, highlighting the need for future work in this area.

This is particularly true for DL-based SA, as, to our knowledge, there is currently no research published on sample size calculation in this specific area. Generally, in SA, the power for detection of effects does not depend on the overall sample size but rather on the number of events (for a specific transition). As a consequence, censoring, truncation and other outcome-related specifics need to be taken into account. For example, effects on the development of a rare condition could be hard to detect if there is a competing event with high prevalence. Additionally taking into account imaging data will generally make the assumed data generating process and therefore sample size calculation more complex. As for more complex statistical models, simulation-based sample size calculation could be a way to go in the future (Snell 2021).

The papers reviewed in this work do not explicitly address sample size requirements. In our *Main Table* we included a column that indicates the minimum dataset size among all benchmarked datasets used for each method. However, this answers a different question about applicability. Most of the methods reviewed will be applicable to rather small data sets, however, their ability to learn anything and outperform simpler baseline models will usually decrease with diminishing sample size.

3.8 Reproducibility

Code and data accessibility foster open and reproducible research. The availability of code can indicate a method's maturity and its general applicability to new use cases. However, the code of algorithms and benchmark experiments is not publicly accessible for 25 methods. Furthermore, the accompanying codes of 28 methods are one-shot implementations and have not yet been processed into easy-to-use packages. Data availability ensures that the reported results can be reproduced and are available for future benchmark experiments. The *Main Table* summarizes reproducibility aspects (in terms of code and data) for all methods.

For usability and reproducibility, new methods should ideally be packaged and also integrated within one of the general purpose suits for machine learning and benchmarking for survival analysis such as `auton-survival` (Nagpal et al. 2022), `mlr3proba` (Sonabend et al. 2021), `pycox` (Kvamme et al. 2019), `scikit-survival` (Pölsterl 2020), or similar.

4 Conclusion

SA is concerned with modeling the time until an event of interest occurs while accounting for censoring, truncation, and other aspects of time-to-event data (cf. Sect. 2.2).

In this paper, we provide a structured, comprehensive review of DL-based survival methods, from a theoretical as well as practical perspective. In doing so, we aim to enable practitioners to quickly gauge the methods available for their specific use case as

well as to help researchers to identify the most promising areas for future research. The main results are summarized in an open-source, interactive, editable table (https://survival-org.github.io/DL4Survival). All data, figures, and code scripts used in this work can be found in the corresponding repository (https://github.com/survival-org/DL4Survival).

We conclude that most methodologically innovative DL-based survival methods are survival-specific applications of novel methods developed in other areas of DL, such as computer vision or NLP. This usually yields a more flexible estimation of associations of (structured and unstructured) features with the outcome, rather than solving problems of time-to-event data not addressed by, e.g., statistical approaches. Outcome types beyond right-censoring and competing risks are rarely addressed, potentially due to a limited number of application cases.

Further, little attention has been paid to optimization (e.g., choice of optimizers, tuning of hyperparameters, or neural architecture search) among the methods reviewed here, as they usually focus on network architecture, data modalities, and specific use cases. Among those articles that did elaborate on optimization, the Adam optimizer (Kingma and Ba 2014) appears to be the most common choice.

There are also some challenges specific to DL-based SA. In the parametric setting, many common log-likelihood-based losses for survival analysis are poorly conditioned. For example, modeling a Weibull distribution that assumes errors from an extreme value distribution (with standardized density $f(t) = \exp(-t)\exp(-\exp(-t))$) may be particularly challenging when being optimized with gradient descent and low precision. Similarly, Avati et al. (2020) recommend the log-normal distribution since optimization of other distributions that are suitable for time-to-event data suffers from numerical instability, as their densities have forms of type $(t\theta_1)^{\theta_2}$ (where θ_1 and θ_2 are parameters of interest) or contain the Gamma function. Their optimization will be particularly challenging, when all parameters of a distribution are learned depending on features [cf. (4)]. Batching is another issue specific to DL-based optimization. In semi-parametric models like the Cox model, batching might become problematic, as already discussed in Sect. 3.3.1. More generally, batching might need to be adapted, depending on the survival task. In recurrent events settings, for example, batching might need to be set up differently, depending on whether one wants to predict next recurrence for all subjects (given previous recurrences) or the entire process for a new subject. Finally, the lack of openly accessible, high-dimensional, potentially multimodal datasets remains a major challenge to the development and training of novel DL-based survival methods.

Missing values are rarely discussed within the methods we reviewed; indeed, most methods implicitly require missing values to be taken care of during data preprocessing. Explicit handling of missing values in the time-to-event setting is done only by *MultiSurv* and *SurvNet*.

In terms of their application, DL-based survival methods have been deployed in estimating patient survival based on medical images (usually CT scans of a particular anomaly) or (multi-)omics data—as evidenced by the large majority of multimodal or high-dimensional methods in this review. Moreover, some methods are explicitly motivated by a specific clinical use case: *DASA* by prostate cancer; *Haa2019*, *su-DeepBTS*, and *SurvNet* by lung cancer; *SALMON*, *ConcatAE/CrossAE*, and *Liu2022* by breast cancer; and *MCAP* by ovarian cancer. Other areas of application of DL-based survival methods include improved estimation of prognostic indices (Bice 2020) and of recurrence after cancer surgery (Lee et al. 2020). The choice of datasets used for benchmarking (see *Main Table*) provides further information about the application cases for each method.

In summary, deep survival methodology has advanced substantially in recent years and will certainly continue to benefit from developments in ML/DL, with big methodological advances being likely to swap over. In particular, generative DL techniques like diffusion are promising candidates for adaptation to survival tasks. The rapid progress in this area of research is also why any overview work can never be fully exhaustive or up-to-date. Therefore, we actively encourage the research community to contribute to our open-source interactive table (https://survival-org.github.io/DL4Survival).

Author contributions AB proposed the research idea. SW and AB developed the analysis plan. SW performed the initial literature search and methods inclusion and performed most of the screening, supported by PK and AB. SW wrote the initial draft, supported by AB. AB and PK contributed to and reviewed the manuscript. SW and PK created the figures and tables. SW created the open-source interactive table. RS and BB reviewed and edited the manuscript and provided valuable feedback.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was funded by the Munich Center for Machine Learning (MCML).

Data availability All data, figures, and code scripts used in this work can be found in the corresponding repository (https://github.com/survival-org/DL4Survival).

Declarations

Competing interests PK, SW, BB, and AB are authors of the *DeepPAMM* method (Kopper et al. 2022). No further competing interest is declared.

References

Aastha, Huang P, Liu Y (2021) DeepCompete: a deep learning approach to competing risks in continuous time domain. In: AMIA annual symposium proceedings, vol 2020. pp 177–186. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8075516/

Agarwal R et al (2021a) Neural additive models: interpretable machine learning with neural nets. Adv Neural Inf Process Syst 34:4699–4711

Agarwal S, Eltigani Osman Abaker M, Daescu O (2021b) Survival prediction based on histopathology imaging and clinical data: a novel, whole slide CNN approach. In: Medical image computing and computer assisted intervention–MICCAI 2021: 24th international conference, Strasbourg, France, September 27–October 1, 2021, proceedings, part V 24. Springer, pp 762–771

Ausset G, Ciffreo T, Portier F, Clémençon S, Papin T (2021) Individual survival curves with conditional normalizing flows. In: 2021 IEEE 8th international conference on data science and advanced analytics (DSAA). pp 1–10. https://doi.org/10.1109/DSAA53316.2021.9564222

Avati A et al (2020) Countdown regression: sharp and calibrated survival predictions. In: Uncertainty in artificial intelligence. PMLR, pp 145–155

Balki I et al (2019) Sample-size determination methodologies for machine learning in medical imaging research: a systematic review. Can Assoc Radiol J 70:344–353

Ballard DH (1987) Modular learning in neural networks. AAAI 647:279–284

Bender A, Groll A, Scheipl F (2018) A generalized additive model approach to time-to-event analysis. Stat Model 18:299–321
Bender A, Rügamer D, Scheipl F, Bischl B (2021) A general machine learning framework for survival analysis. In: Hutter F, Kersting K, Lijffijt J, Valera I (eds) Machine learning and knowledge discovery in databases. Springer International Publishing, pp 158–173. https://doi.org/10.1007/978-3-030-67664-3_10
Bennis A, Mouysset S, Serrurier M (2020) Estimation of conditional mixture Weibull distribution with right censored data using neural network for time-to-event analysis. In: Advances in knowledge discovery and data mining: 24th Pacific-Asia conference, PAKDD 2020, Singapore, May 11–14, 2020, proceedings, part I 24. Springer, pp 687–698
Bennis A, Mouysset S, Serrurier M (2021) DPWTE: a deep learning approach to survival analysis using a parsimonious mixture of Weibull distributions. In: Farkaš I, Masulli P, Otte S, Wermter S (eds) Artificial neural networks and machine learning—ICANN 2021. Lecture notes in computer science. Springer International Publishing. pp 185–196. https://doi.org/10.1007/978-3-030-86340-1_15
Bice N et al (2020) Deep learning-based survival analysis for brain metastasis patients with the national cancer database. J Appl Clin Med Phys 21:187–192
Biganzoli E, Boracchi P, Mariani L, Marubini E (1998) Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med 17:1169–1186
Binder H, Schumacher M (2008) Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. BMC Bioinform 9:1–10
Boulesteix A-L, Lauer S, Eugster MJA (2013) A plea for neutral comparison studies in computational sciences. PLoS ONE 8:e61562. https://doi.org/10.1371/journal.pone.0061562
Boulesteix A-L, Hoffmann S, Charlton A, Seibold H (2020) A replication crisis in methodological research? Significance 17:18–21. https://doi.org/10.1111/1740-9713.01444
Breiman L (2001) Random forests. Mach Learn 45:5–32
Brown SF, Branford AJ, Moran W (1997) On the use of artificial neural networks for the analysis of survival data. IEEE Trans Neural Netw 8:1071–1077
Chai H, Guo L, He M, Zhang Z, Yang Y (2022) A multi-constraint deep semi-supervised learning method for ovarian cancer prognosis prediction. In: Advances in swarm intelligence: 13th international conference, ICSI 2022, Xi'an, China, July 15–19, 2022, proceedings, part II. Springer, pp 219–229
Chapfuwa P et al (2018) Adversarial time-to-event modeling. In: International conference on machine learning. PMLR, pp 735–744
Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK (2018) Neural ordinary differential equations. Adv Neural Inf Process Syst 31:6572–6583
Chi S et al (2021) Deep semisupervised multitask learning model and its interpretability for survival analysis. IEEE J Biomed Health Inform 25:3185–3196
Ching T, Zhu X, Garmire LX (2018) Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. PLoS Comput Biol 14:e1006076
Cho HJ, Shu M, Bekiranov S, Zang C, Zhang A (2023) Interpretable meta-learning of multi-omics data for survival analysis and pathway enrichment. Bioinformatics 39:btad113
Cottin A, Pecuchet N, Zulian M, Guilloux A, Katsahian S (2022) IDNetwork: a deep illness-death network based on multi-state event history process for disease prognostication. Stat Med 41:1573–1598
Cox DR (1972) Regression models and life-tables. J R Stat Soc Ser B (Methodol) 34:187–202
Deepa P, Gunavathi C (2022) A systematic review on machine learning and deep learning techniques in cancer survival prediction. Prog Biophys Mol Biol. https://doi.org/10.1016/j.pbiomolbio.2022.07.004
Fan Y, Zhang S, Ma S (2022) Survival analysis with high-dimensional omics data using a threshold gradient descent regularization-based neural network approach. Genes 13:1674
Fang Y et al (2021) The impact of training sample size on deep learning-based organ auto-segmentation for head-and-neck patients. Phys Med Biol 66:185012
Faraggi D, Simon R (1995) A neural network model for survival data. Stat Med 14:73–82
Fornili M, Ambrogi F, Boracchi P, Biganzoli E (2013) Piecewise exponential artificial neural networks (PEANN) for modeling hazard function with right censored data. In: International meeting on computational intelligence methods for bioinformatics and biostatistics. Springer, pp 125–136
Fotso S (2018) Deep neural networks for survival analysis based on a multi-task framework. arXiv:1801.05512 [cs, stat]
Friedman M (1982) Piecewise exponential models for survival data with covariates. Ann Stat 10:101–113
Fuhlert P et al (2022) Deep learning-based discrete calibrated survival prediction. In: 2022 IEEE international conference on digital health (ICDH). IEEE, pp 169–174
Gensheimer MF, Narasimhan B (2019) A scalable discrete-time survival model for neural networks. PeerJ 7:e6257

Giunchiglia E, Nemchenko A, van der Schaar M (2018) RNN-SURV: a deep recurrent model for survival analysis. In: Artificial neural networks and machine learning—ICANN 2018. Lecture notes in computer science. Springer International Publishing, pp 23–32. https://doi.org/10.1007/978-3-030-01424-7_3

Goodfellow IJ et al (2014) Generative adversarial nets. In: NIPS

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

Groha S, Schmon SM, Gusev A (2021) A general framework for survival analysis and multi-state modelling. arXiv:2006.04893 [cs, stat]

Gupta G, Sunder V, Prasad R, Shroff G (2019) CRESA: a deep learning approach to competing risks, recurrent event survival analysis. In: Advances in knowledge discovery and data mining: 23rd Pacific-Asia conference, PAKDD 2019, Macau, China, April 14–17, 2019, proceedings, part II 23. Springer, pp 108–122

Haarburger C, Weitz P, Rippel O, Merhof D (2019) Image-based survival prediction for lung cancer patients using CNNS. In: 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019). pp 1197–1201. ISSN: 1945-8452

Haider H, Hoehn B, Davis S, Greiner R (2020) Effective ways to build and evaluate individual survival distributions. J Mach Learn Res 21:3289–3351

Hao J, Kim Y, Mallavarapu T, Oh JH, Kang M (2018) Cox-PASNet: pathway-based sparse deep neural network for survival analysis. pp 381–386

Hao J, Kosaraju SC, Tsaku NZ, Song DH, Kang M (2019) PAGE-Net: interpretable and integrative deep learning for survival analysis using histopathological images and genomic data. In: Pacific symposium on biocomputing 2020. World Scientific, pp 355–366

Harrell FE, Califf RM, Pryor DB (1982) Evaluating the yield of medical tests. J Am Med Assoc 247:2543–2546. https://doi.org/10.1001/jama.1982.03320430047030

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778

Herrmann M, Probst P, Hornung R, Jurinovic V, Boulesteix A-L (2020) Large-scale benchmark study of survival prediction methods using multi-omics data. arXiv:2003.03621 [cs, stat]

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2:359–366

Hu S, Fridgeirsson E, Wingen GV, Welling M (2021) Transformer-based deep survival analysis. In: Proceedings of AAAI spring symposium on survival prediction—algorithms, challenges, and applications. PMLR, pp 132–148. https://proceedings.mlr.press/v146/hu21a.html. ISSN 2640-3498

Huang C, Zhang A, Xiao G (2018) Deep integrative analysis for survival prediction. Biocomputing. https://doi.org/10.1142/9789813235533_0032

Huang Z et al (2019) Salmon: survival analysis learning with multi-omics neural networks on breast cancer. Front Genet 10:166

Irvin J et al (2019) CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence, vol 33. pp 590–597

Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS et al (2008) Random survival forests. Ann Appl Stat 2:841–860

Ivakhnenko AG, Lapa VG, Lapa VG (1967) Cybernetics and forecasting techniques, vol 8. American Elsevier Publishing Company, New York

Jetley S, Lord NA, Lee N, Torr PH (2018) Learn to pay attention. arXiv Preprint. https://arxiv.org/1804.02391

Jing B et al (2019) A deep survival analysis method based on ranking. Artif Intell Med 98:1–9

Kalakoti Y, Yadav S, Sundar D (2021) SurvCNN: a discrete time-to-event cancer survival estimation framework using image representations of omics data. Cancers 13:3106

Kalbfleisch JD, Prentice RL (2011) The statistical analysis of failure time data. Wiley, Hoboken

Kamran F, Wiens J (2021) Estimating calibrated individualized survival curves with deep learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 35. pp 240–248

Kaplan EL, Meier P (1958) Nonparametric estimation from incomplete observations. J Am Stat Assoc 53:457–481

Katzman JL et al (2018) DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Med Res Methodol 18:24

Kim S, Kim K, Choe J, Lee I, Kang J (2020) Improved survival analysis by learning shared genomic information from pan-cancer data. Bioinformatics 36:i389–i398

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv Preprint. https://arxiv.org/1412.6980
Kingma DP, Welling M (2013) Auto-encoding variational Bayes. arXiv Preprint. https://arxiv.org/abs/1312.6114
Klein JP, Moeschberger ML (1997) Survival analysis: techniques for censored and truncated data. Springer, New York
Kopper P et al (2021) Semi-structured deep piecewise exponential models. arXiv:2011.05824 [cs, stat]
Kopper P, Wiegrebe S, Bischl B, Bender A, Rügamer D (2022) DeepPAMM: deep piecewise exponential additive mixed models for complex hazard structures in survival analysis. In: Advances in knowledge discovery and data mining: 26th Pacific-Asia conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, proceedings, part II. Springer, pp 249–261
Kovalev MS, Utkin LV, Kasimov EM (2020) SurvLIME: a method for explaining machine learning survival models. Knowl Based Syst 203:106164
Krzyziński M, Spytek M, Baniecki H, Biecek P (2022) SurvSHAP(t): time-dependent explanations of machine learning survival models. Knowl Based Syst 262:110234
Kvamme H, Borgan Ø (2019) Continuous and discrete-time survival prediction with neural networks. arXiv:1910.06724 [cs, stat]
Kvamme H, Borgan Ø (2021) Continuous and discrete-time survival prediction with neural networks. Lifetime Data Anal 27:710–736
Kvamme H, Borgan Ø, Scheel I (2019) Time-to-event prediction with neural networks and Cox regression. J Mach Learn Res 20:1–30
LeCun Y et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551
Lee SC, Lee ET (1975) Fuzzy neural networks. Math Biosci 23:151–177
Lee S, Lim H (2019) Review of statistical methods for survival analysis using genomic data. Genomics Inform 17:e41
Lee C, Zame WR, Yoon J, van der Schaar M (2018) DeepHit: a deep learning approach to survival analysis with competing risks. In: AAAI. pp 2314–2321
Lee C, Yoon J, Van Der Schaar M (2019) Dynamic-DeepHit: a deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. IEEE Trans Biomed Eng 67:122–133
Lee B et al (2020) DeepBTS: prediction of recurrence-free survival of non-small cell lung cancer using a time-binned deep neural network. Sci Rep 10:1952
Li Y et al (2020) DeepComp: which competing event will hit the patient first? In: 2020 IEEE international conference on bioinformatics and biomedicine (BIBM). pp 629–636. https://doi.org/10.1109/BIBM49941.2020.9313333
Liestbl K, Andersen PK, Andersen U (1994) Survival analysis and neural nets. Stat Med 13:1189–1200
Lin J, Luo S (2022) Deep learning for the dynamic prediction of multivariate longitudinal and survival data. Stat Med 41:2894–2907
Liu H, Kurc T (2022) Deep learning for survival analysis in breast cancer with whole slide image data. Bioinformatics 38:3629–3637
Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Advances in neural information processing systems, vol 30
Meister R, Schaefer C (2008) Statistical methods for estimating the probability of spontaneous abortion in observational studies-analyzing pregnancies exposed to coumarin derivatives. Reprod Toxicol 26:31–35
Montavon G, Binder A, Lapuschkin S, Samek W, Müller K-R (2019) Layer-wise relevance propagation: an overview. In: Explainable AI: interpreting, explaining and visualizing deep learning. pp 193–209
Nagpal C, Li X, Dubrawski A (2021a) Deep survival machines: fully parametric survival regression and representation learning for censored data with competing risks. IEEE J Biomed Health Inform 25:3163–3175
Nagpal C, Jeanselme V, Dubrawski A (2021b) Deep parametric time-to-event regression with time-varying covariates. In: Survival prediction-algorithms, challenges and applications. PMLR, pp 184–193
Nagpal C, Yadlowsky S, Rostamzadeh N, Heller K (2021c) Deep Cox mixtures for survival regression. In: Machine learning for healthcare conference. PMLR, pp 674–708
Nagpal C, Potosnak W, Dubrawski A (2022) Auton-survival: an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data. arXiv Preprint. https://arxiv.org/abs/2204.07276
Nezhad MZ, Sadati N, Yang K, Zhu D (2019) A deep active survival analysis approach for precision treatment recommendations: application of prostate cancer. Expert Syst Appl 115:16–26

Nießl C, Herrmann M, Wiedemann C, Casalicchio G, Boulesteix A-L (2022) Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. WIREs Data Min Knowl Discov 12:e1441. https://doi.org/10.1002/widm.1441
Noordzij M et al (2013) When do we need competing risks methods for survival analysis in nephrology? Nephrol Dial Transplant 28:2670–2677
Pölsterl S (2020) Scikit-survival: a library for time-to-event analysis built on top of scikit-learn. J Mach Learn Res 21:1–6
Pölsterl S, Sarasua I, Gutiérrez-Becker B, Wachinger C (2020) A wide and deep neural network for survival analysis from anatomical shape and tabular clinical data. In: Cellier P, Driessens K (eds) Machine learning and knowledge discovery in databases. Springer International Publishing, pp 453–464. https://doi.org/10.1007/978-3-030-43823-4_37
Qi CR, Su H, Mo K, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 652–660
Qin X, Yin D, Dong X, Chen D, Zhang S (2022) Survival prediction model for right-censored data based on improved composite quantile regression neural network. Math Biosci Eng 19:7521–7542
Qiu YL, Zheng H, Devos A, Selby H, Gevaert O (2020) A meta-learning approach for genomic survival analysis. Nat Commun 11:6350
Ramjith J, Roes KC, Zar HJ, Jonker MA (2021) Flexible modelling of risk factors on the incidence of pneumonia in young children in South Africa using piece-wise exponential additive mixed modelling. BMC Med Res Methodol 21:1–13
Ranganath R, Perotte A, Elhadad N, Blei D (2016) Deep survival analysis. In: Artificial intelligence and statistics. pp 101–114
Ren K et al (2019) Deep recurrent survival analysis. In: Proceedings of the AAAI conference on artificial intelligence, vol 33. pp 4798–4805
Rezende D, Mohamed S (2015) Variational inference with normalizing flows. In: International conference on machine learning. PMLR, pp 1530–1538
Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp 1135–1144
Rindt D, Hu R, Steinsaltz D, Sejdinovic D (2022) Survival regression with proper scoring rules and monotonic neural networks. In: International conference on artificial intelligence and statistics. PMLR, pp 1190–1205
Rosenblatt F (1967) Recent work on theoretical models of biological memory. In: Computer and information sciences II. pp 33–56
Ruder S (2017) An overview of multi-task learning in deep neural networks. arXiv Preprint. https://arxiv.org/abs/1706.05098
Rügamer D (2023) A new PHO-rmula for improved performance of semi-structured networks. In: Proceedings of the 40th international conference on machine learning. PMLR, pp 29291–29305. https://proceedings.mlr.press/v202/rugamer23a.html
Rügamer D, Kolb C, Klein N (2023) Semi-structured distributional regression. Am Stat. https://doi.org/10.1080/00031305.2022.2164054
Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536
Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Advances in neural information processing systems, vol 30
Sansaengtham B, Barroso VC, Phunchongharn P (2020) Survival analysis for computing systems using a deep ensemble network. In: 2020 IEEE 6th international conference on control science and systems engineering (ICCSSE). IEEE, pp 57–62
Schwarzer G, Vach W, Schumacher M (2000) On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. Stat Med 19:541–561
Shahinfar S, Meek P, Falzon G (2020) how many images do i need? Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. Eco Inform 57:101085
Shin B et al (2019) Cascaded Wx: a novel prognosis-related feature selection framework in human lung adenocarcinoma transcriptomes. Front Genet 10:662
Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In: International conference on machine learning. PMLR, pp 3145–3153
Snell KIE et al (2021) External validation of clinical prediction models: simulation-based sample size calculations were more reliable than rules-of-thumb. J Clin Epidemiol 135:79–89

Sohl-Dickstein J, Weiss EA, Maheswaranathan N, Ganguli S (2015) Deep unsupervised learning using non-equilibrium thermodynamics. In: ICML
Sonabend REB (2021) A theoretical and methodological framework for machine learning in survival analysis: enabling transparent and accessible predictive modelling on right-censored time-to-event data. PhD, University College London (UCL). https://discovery.ucl.ac.uk/id/eprint/10129352/
Sonabend R (2022) Scoring rules in survival analysis. arXiv Preprint. https://arxiv.org/2212.05260
Sonabend R, Király FJ, Bender A, Bischl B, Lang M (2021) mlr3proba: an r package for machine learning in survival analysis. Bioinformatics. https://doi.org/10.1093/bioinformatics/btab039
Sonabend R, Bender A, Vollmer S (2022) Avoiding C-hacking when evaluating survival distribution predictions with discrimination measures. Bioinformatics 38:4178–4184. https://doi.org/10.1093/bioinformatics/btac451
Steele AJ, Denaxas SC, Shah AD, Hemingway H, Luscombe NM (2018) Machine learning models in electronic health records can outperform conventional survival models for predicting patient mortality in coronary artery disease. PLoS ONE 13:e0202344
Sun T, Wei Y, Chen W, Ding Y (2020) Genome-wide association study-based deep learning for survival prediction. Stat Med 39:4605–4620
Tang B, Li A, Li B, Wang M (2019) CapSurv: capsule network for survival analysis with whole slide pathological images. IEEE Access 7:26022–26030
Thorsen-Meyer H-C et al (2022) Discrete-time survival analysis in the critically ill: a deep learning approach using heterogeneous data. npj Digit Med 5:142
Tong J, Zhao X (2022) Deep survival algorithm based on nuclear norm. J Stat Comput Simul 92:1964–1976
Tong L, Mitchel J, Chatlin K, Wang MD (2020) Deep learning based feature-level integration of multi-omics data for breast cancer patients survival analysis. BMC Med Inform Decis Mak 20:225. https://doi.org/10.1186/s12911-020-01225-8
Tutz G, Schmid M et al (2016) Modeling discrete time-to-event data. Springer, New York
Uno H, Cai T, Tian L, Wei LJ (2007) Evaluating prediction rules for t-year survivors with censored regression models. J Am Stat Assoc 102:527–537
Utkin LV, Satyukov ED, Konstantinov AV (2022) SurvNAM: the machine learning survival model explanation. Neural Netw 147:81–102
Vale-Silva LA, Rohr K (2021) Long-term cancer survival prediction using multimodal deep learning. Sci Rep 11:13505
Vaswani A et al (2017) Attention is all you need. Adv Neural Inf Process Syst 30:6000–6010
Vellido A (2020) The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Comput Appl 32:18069–18083
Vincent P et al (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408
Wang H, Li G (2019) Extreme learning machine cox model for high-dimensional survival analysis. Stat Med 38:2139–2156
Wang Z, Sun J (2022) 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
Wang P, Li Y, Reddy CK (2019) Machine learning for survival analysis: a survey. ACM Comput Surv (CSUR) 51:1–36
Wang J et al (2021) SurvNet: a novel deep neural network for lung cancer survival analysis with missing values. Front Oncol 10:3128. https://doi.org/10.3389/fonc.2020.588990
Wijethilake N et al (2021) Glioma survival analysis empowered with data engineering—a survey. IEEE Access 9:43168–43191
Wolf TN, Pölsterl S, Wachinger C, Initiative ADN et al (2022) Daft: a universal module to interweave tabular data and 3d images in CNNs. NeuroImage 260:119505
Wolpert DH (1992) Stacked generalization. Neural Netw 5:241–259
Wu C et al (2019) A selective review of multi-level omics data integration using variable selection. High-Throughput 8:4
Xie Y, Yu Z (2021) Mixture cure rate models with neural network estimated nonparametric components. Comput Stat 36:2467–2489. https://doi.org/10.1007/s00180-021-01086-3
Xie G et al (2019) Group lasso regularized deep learning for cancer prognosis from multi-omics and clinical features. Genes 10:240
Yin Q, Chen W, Zhang C, Wei Z (2022) A convolutional neural network model for survival prediction based on prognosis-related cascaded Wx feature selection. Lab Investig 102:1064–1074
Yousefi S et al (2017) Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. Sci Rep 7:11707

Yu C-N, Greiner R, Lin H-C, Baracos V (2011) Learning patient-specific cancer survival distributions as a sequence of dependent regressors. Adv Neural Inf Process Syst 24:1845–1853

Zhang J, Huang K (2014) Normalized ImQCM: an algorithm for detecting weak quasi-cliques in weighted graph with applications in gene co-expression module discovery in cancers. Cancer Inform 13:CIN-S14021

Zhang Y et al (2020) CNN-based survival model for pancreatic ductal adenocarcinoma in medical imaging. BMC Med Imaging 20:1–8

Zhang Y, Wong G, Mann G, Muller S, Yang JY (2022) SurvBenchmark: comprehensive benchmarking study of survival analysis methods using both omics data and clinical data. GigaScience 11:giac071

Zhao L, Feng D (2019) DNNSurv: deep neural networks for survival analysis using pseudo values. arXiv: 1908.02337 [cs, stat]

Zhao L et al (2021) DeepOmix: a scalable and interpretable multi-omics deep learning framework and application in cancer survival analysis. Comput Struct Biotechnol J 19:2719–2725

Zhu X, Yao J, Huang J (2016) Deep convolutional neural network for survival analysis with pathological images. In: 2016 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp 544–547. https://doi.org/10.1109/BIBM.2016.7822579. http://ieeexplore.ieee.org/document/7822579/