Feature Selection for Survival Analysis with Competing Risks using Deep Learning

Carl Rietschel[†], **Jinsung Yoon**[*] **and Mihaela van der Schaar**[†*]
[†]University of Oxford, UK. [*]University of California Los Angeles, USA
carlrietschel@gmail.com, jsyoon0823@gmail.com, mihaela@ee.ucla.edu

Abstract

Deep learning models for survival analysis have gained significant attention in the literature, but they suffer from severe performance deficits when the dataset contains many irrelevant features. We give empirical evidence for this problem in real-world medical settings using the state-of-the-art model DeepHit. Furthermore, we develop methods to improve the deep learning model through novel approaches to feature selection in survival analysis. We propose filter methods for *hard* feature selection and a neural network architecture that weights features for *soft* feature selection. Our experiments on two real-world medical datasets demonstrate that substantial performance improvements against the original models are achievable.

1 Introduction

Recent research has produced a variety of successful new deep learning models for survival analysis. Whilst some methods [7, 16, 12] have strong parametric assumptions, more general models [13, 5, 10] have been developed. However, deep learning approaches suffer from performance deficits when there are many irrelevant features. This can certainly be the case in medical datasets, where numerous features may be recorded about a patient (e.g. the PLCO dataset we use in this work - see details in Appendix A). In this paper, we give evidence for this problem using DeepHit[1] [10] on large real-world medical datasets, and propose feature selection techniques to achieve substantial performance improvements.

2 Survival analysis with DeepHit$^+$

We consider survival analysis with K competing risks, focusing on the medical domain with right-censored data, as is frequently encountered in time-limited medical trials. Survival data for each of N patients is a tuple $(\mathbf{x}, \tau, \delta)$: *Covariates* $\mathbf{x}$ are the characteristics of each patient observed. *Time* τ is measured from when the covariates were collected until the first event or censoring. *Label* $\delta \in \{\emptyset, 1, \ldots, K\}$ indicates which event (or right censoring, denoted $\emptyset$) occurred at time τ. For each event k the cause-specific *cumulative incidence function* (CIF) gives the probability that this event occurs on or before time t for a patient with covariates $\mathbf{x}$, and is denoted $F_k(t \mid \mathbf{x}) = \mathbb{P}[\tau \leq t, \delta = k \mid \mathbf{x}]$. A machine learning model will attempt to give empirical estimates $\hat{F}_k(t \mid \mathbf{x})$ of the true cumulative incidence functions F_k.

In this paper we use DeepHit because it is very general, allows for competing risks and shows good empirical performance [10]. We furthermore develop DeepHit$^+$ to implement two improvements: We switch the early stopping criterion from validation loss to the performance

[1]We thank the authors of DeepHit for sharing their source-code implementation with us

Machine Learning for Health (ML4H) Workshop at NeurIPS 2018.

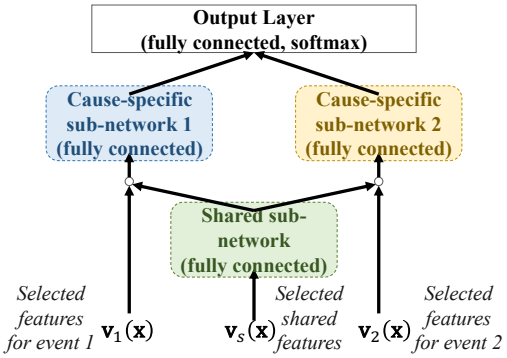

Figure 1: FilterDeepHit$^+$ architecture

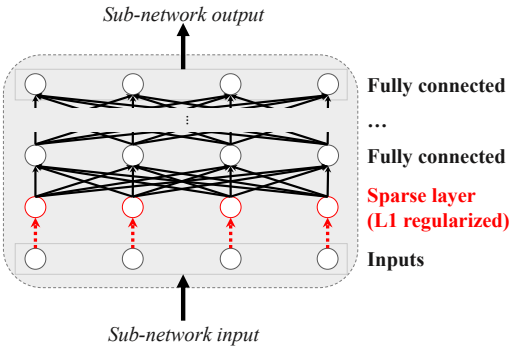

Figure 2: SparseDeepHit$^+$ sub-network

measure C-index, and allow for random search on the sizes (number of layers and hidden dimension) of the shared and cause-specific sub-networks.

3 Automatic feature selection for DeepHit$^+$

We propose models with *filter* approaches to *hard* feature selection, as well as reweighting features in *soft* feature selection techniques, using DeepHit$^+$'s architecture as a basis for our methods. We also develop a *hybrid* feature selection approach in Appendix B, but do not investigate wrapper feature selection techniques due to their computational complexity [9].

3.1 FilterDeepHit$^+$: Automatic feature selection using filter method

Whilst it would be possible to feed a subset of features in place of $\mathbf{x}$ into DeepHit$^+$'s original network, we desire the ability to make different feature selections for each competing risk. FilterDeepHit$^+$ thus uses the modified architecture shown in Figure 1, where the input connections to the shared and cause-specific sub-networks are replaced with certain selected subsets of the features. We denote these $\mathbf{v}_s(\mathbf{x})$ for the shared sub-network and $\mathbf{v}_k(\mathbf{x}), k = 1, 2, ...$ for the kth cause-specific sub-network.

We cannot directly apply classical filter feature selection methods for classification or regression tasks to survival analysis: due to the right-censored data, the label is not determined for some patients. However, when fixing the time horizon to Δt, survival analysis becomes a classification task for which we can use filter techniques. In particular, if for patient i time $\tau^{(i)}$ and label $\delta^{(i)}$ are observed, the binary cause-specific time-fixed label would be

$$\delta^{(i,\Delta t,k)} = \begin{cases} 1, & \text{if } \tau^{(i)} < \Delta t \text{ and } \delta^{(i)} = k \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

For each event, we apply the filter method multiple times to score associations between features and the cause-specific label for a preselected set of evaluation time horizons. The final score for each feature is then the average of its scores across the time horizons, and is used for automatic feature selection for the survival model. Our method in FilterDeepHit$^+$ automatically selects the features for each cause-specific sub-network $\mathbf{v}_k(\mathbf{x})$ from the feature ranking. It chooses the top m_k features according to the ranking, treating the number of features to be selected $(m_k, k = 1, \ldots, K)$ as an additional hyperparameter optimized through random search. The shared sub-network's input features $\mathbf{v}_s$ are the common features that occur in all $\mathbf{v}_k$ (i.e. their intersection). If there are none, FilterDeepHit$^+$ does not have a shared sub-network.

We desire methods that are able to deal with both continuous and categorical features, and choose three common filters for three new models: Analysis of variance / t-tests (**ANOVA**) [15], weights of a trained support vector machine (**SVM**) [2] and the **ReliefF** algorithm [8].

3.2 SparseDeepHit$^+$: Automatic feature selection using sparse initial layer

Inspired by linear models that achieve low or zero weights through introducing L_1 regularization, [11] proposes adding an additional sparse layer with one-to-one connections from the input before the first hidden layer of a deep neural network. Our architecture for SparseDeepHit$^+$ applies this to the multitask learning setting of DeepHit$^+$, incorporating additional sparse layers for the shared sub-network as well as each of the cause-specific sub-networks. In Figure 2, we show this by illustrating the nodes, and connections between nodes in each layer.

Mathematically, we denote the input to the shared network $\mathbf{x}$, and the inputs to the kth cause-specific network $\mathbf{z}_k$. Then we define parameters $\mathbf{w}_s$ and $\mathbf{w}_k, k = 1, \ldots, K$, and the activations of the sparse layers are given by

$$\mathbf{z}_s^{'} = \mathbf{x} \odot \mathbf{w}_s \tag{2}$$

$$\mathbf{z}_k^{'} = \mathbf{z}_k \odot \mathbf{w}_k, \quad k = 1, \ldots, K, \tag{3}$$

where $\odot$ denotes the element-wise (*Hadamard*) product of two vectors. These outputs $\mathbf{z}_s^{'}$ and $\mathbf{z}_k^{'}$ are then the inputs to the fully connected parts of the sub-networks as before.

The loss function, originally $\mathcal{L}$, is adjusted to include additional L_1 regularization terms:

$$\mathcal{L}_{\text{Total}} = \mathcal{L} + \gamma_s \left\|\mathbf{w}_s\right\|_1 + \sum_{k=1}^{K} \gamma_k \left\|\mathbf{w}_k\right\|_1, \tag{4}$$

for additional positive hyperparameters γ_s and $\gamma_k, k = 1, \ldots, K$.

4 Experiments

We evaluate all new methods against DeepHit [10] as well as the most common survival analysis models for competing risks, Fine-Gray models [3] and Random Survival Forests (RSF) [6], on two medical datasets described below (details in Appendix A).

The Prostate, Lung, Colorectal and Ovarian Cancer (**PLCO**) Screening Trial includes baseline information and prostate cancer screening data (105 features) from 38,052 patients, for which we predict prostate cancer incidence (Event 1), with death prior to prostate cancer incidence (Event 2) a competing risk. The trial has been previously described in [14].

We also use an extracted cohort of 72,809 breast cancer patients from the Surveillance, Epidemiology, and End Results (**SEER**) Program. The patients have baseline medical information (23 features) to predict death from breast cancer (Event 1). Death from cardiovascular disease (Event 2) and death from other causes (Event 3) are treated as competing risks. In order to compare performance when there are a large number of irrelevant features, we also artificially extended the SEER dataset with 20 to 100 synthetic binary features to form the **SEERsynth** dataset. The additional features are constructed for each patient independently from both other features and the survival distribution.

We use an extension of the C-index [4] to evaluate discriminative performance at various time horizons, and adapt it to the competing risks setting. Given estimates $\hat{F}_k$ of cumulative incidence functions, patient features $\mathbf{x}$ and survival times τ, the C-index for event k is

$$C_k(\Delta t) = \mathbb{P}\left[\hat{F}_k(\Delta t \mid \mathbf{x}^i) > \hat{F}_k(\Delta t \mid \mathbf{x}^j) \middle| \tau^i < \tau^j, k^i = k, \tau^i < \Delta t\right]. \tag{5}$$

We evaluate performance at time horizons Δt of 12, 36, 60 and 120 months. The results are presented as averages over 5 train/test splits from 5-fold cross-validation. For DeepHit-based models we run 50 iterations of random search for hyperparameters on the first cross-validation (except for DeepHit, where 20 iterations suffice due to the smaller search space). The optimal parameters are then applied to all other train/test splits (see Appendix C for details).

Table 1 shows the C-index results of all methods for the first and third evaluation horizons (see Appendix D for full results). We find that DeepHit significantly underperforms the

Table 1: C-index performance of all models on PLCO and SEERsynth. Bold figures indicate the best performing method for each event and evaluation horizon.

Horizon	Algorithm	PLCO Event 1	PLCO Event 2	SEER (100 synth. features) Event 1	SEER (100 synth. features) Event 2	SEER (100 synth. features) Event 3
$\Delta t = 012$	**SparseDeepHit$^+$**	0.930	**0.760**	0.846	0.698	0.763
	FilterDeepHit$^+$ (Anova)	**0.935**	0.753	0.840	0.662	0.760
	FilterDeepHit$^+$ (SVM)	0.933	0.741	0.809	0.696	0.748
	FilterDeepHit$^+$ (ReliefF)	0.931	0.747	0.836	**0.709**	**0.769**
	DeepHit$^+$	0.927	**0.760**	0.822	0.687	0.750
	DeepHit	0.906	0.746	0.798	0.660	0.736
	RSF	0.934	0.716	**0.852**	0.616	0.747
	Fine-Gray	0.710	**0.760**	0.793	0.642	0.701
$\Delta t = 060$	**SparseDeepHit$^+$**	0.869	**0.757**	0.758	0.678	**0.697**
	FilterDeepHit$^+$ (Anova)	**0.871**	0.751	0.754	0.651	**0.697**
	FilterDeepHit$^+$ (SVM)	0.870	0.744	0.741	0.695	0.685
	FilterDeepHit$^+$ (ReliefF)	0.866	0.746	0.749	**0.698**	0.696
	DeepHit$^+$	0.845	0.756	0.717	0.665	0.682
	DeepHit	0.807	0.744	0.695	0.648	0.682
	RSF	0.866	0.748	**0.770**	0.680	0.693
	Fine-Gray	0.647	0.755	0.667	0.662	0.672

traditional algorithm RSF on both datasets. DeepHit$^+$ delivers improvements through more adaptive layer sizing. SparseDeepHit$^+$ achieves a 1.2% improvement of average C-index over DeepHit$^+$ on PLCO, and 3.1% on SEERsynth. FilterDeepHit$^+$ methods outperform DeepHit$^+$ on PLCO by 0.3-0.9%, and 1.6-3.3% on SEERsynth.

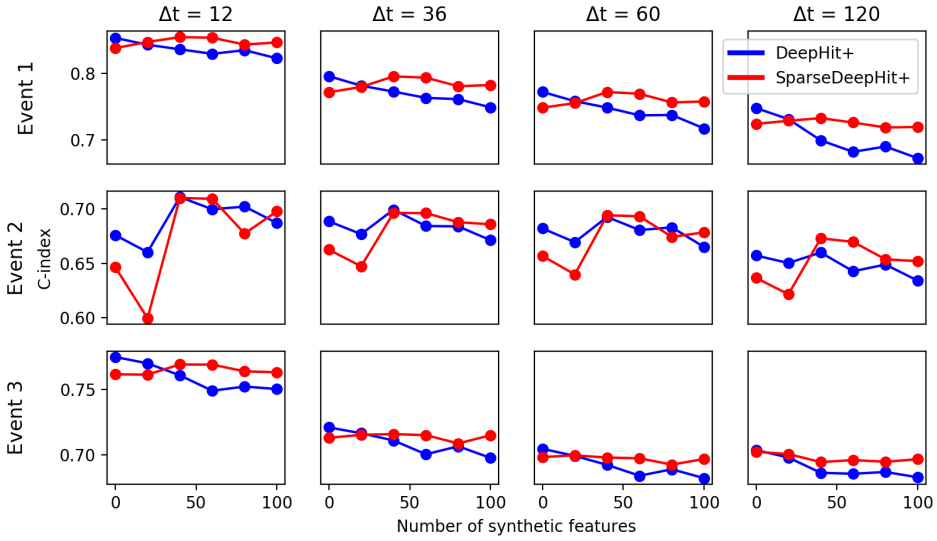

Figure 3: DeepHit$^+$ and SparseDeepHit$^+$ on SEER with varying number of synthetic features

Figure 3 shows how the performance of DeepHit$^+$ decreases when synthetic features are added. This gives evidence that the performance of deep learning survival analysis methods can significantly suffer in settings where the number of potentially irrelevant features becomes large, motivating the need for feature selection applications. In Figure 3 we also observe that the performance with feature selection, given by SparseDeepHit$^+$, remains more stable as synthetic features are added. We note that for Event 2 performance is variable due to the small number of observations (1% of the entire dataset), which results in small samples particularly for validation and testing (see Appendix A for dataset details).

Another benefit of feature selection is the output of feature rankings that contribute to the model's interpretability. We present these results in Appendix E, where it is shown that on PLCO, the medically relevant PSA level and DRE result are ranked top, whilst on SEERsynth irrelevant synthetic features are mostly correctly filtered out.

References

[1] Leo Breiman. Random forests. *Machine Learning*, 2001.

[2] E. E. Bron, M. Smits, W. J. Niessen, and S. Klein. Feature selection based on the svm weight vector for classification of dementia. *IEEE Journal of Biomedical and Health Informatics*, 2015.

[3] Jason P. Fine and Robert J. Gray. A proportional hazards model for the subdistribution of a competing risk. *Journal of the American Statistical Association*, 1999.

[4] Thomas A Gerds, Michael W Kattan, Martin Schumacher, and Changhong Yu. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Stat Med*, 2013.

[5] Eleonora Giunchiglia, Anton Nemchenko, and Mihaela van der Schaar. Rnn-surv: A deep recurrent model for survival analysis. In *International Conference on Artificial Neural Networks*, 2018.

[6] Hemant Ishwaran, Thomas A. Gerds, Udaya B. Kogalur, Richard D. Moore, Stephen J. Gange, and Bryan M. Lau. Random survival forests for competing risks. *Biostatistics*, 2014.

[7] Jared Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deep survival: A deep cox proportional hazards network. 2016.

[8] Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 1997.

[9] Mirosław Kordos et al. Data selection for neural networks. *Schedae Informaticae*, 2017.

[10] Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *AAAI*, 2018.

[11] Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. Deep feature selection: Theory and application to identify enhancers and promoters. *J Comput Biol*, 2016.

[12] Margaux Luck, Tristan Sylvain, Héloïse Cardinal, Andrea Lodi, and Yoshua Bengio. Deep learning for patient-specific kidney graft survival analysis. *CoRR*, 2017.

[13] Anton Nemchenko, Trent Kyono, and Mihaela Van Der Schaar. Siamese survival analysis with competing risks. In *International Conference on Artificial Neural Networks*, 2018.

[14] P C Prorok, G L Andriole, R S Bresalier, S S Buys, D Chia, E D Crawford, R Fogel, E P Gelmann, F Gilbert, M A Hasson, R B Hayes, C C Johnson, J S Mandel, A Oberman, B O'Brien, M M Oken, S Rafla, D Reding, W Rutt, J L Weissfeld, L Yokochi, and J K Gohagan. Design of the prostate, lung, colorectal and ovarian (plco) cancer screening trial. *Control Clin Trials*, 2000.

[15] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 2007.

[16] X. Zhu, J. Yao, and J. Huang. Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2016.

Appendix

A Datasets

A.1 PLCO

The Prostate, Lung, Colorectal and Ovarian (PLCO) Cancer Screening Trial provides a comprehensive dataset to analyse its prostate cancer screening, incidence and mortality results. Its design has been previously described by [14]. Between 1993 and 2001, participants aged between 55 and 74 were enrolled in 10 US study centers and randomised into a screening and control group. Those in the screening group were offered annual PSA testing for 6 years and annual digital rectal examination for 4 years, whilst those in the control group received normal care. For both groups, all diagnosed cancers and deaths were collected and monitored, whilst screening results were recorded only for the screening group. In this analysis we restrict our dataset to the screening group in order to target predictions based on screening data.

The event of interest is prostate cancer incidence (Event 1). Furthermore, the participant could have been right-censored prior to prostate cancer diagnosis, or have died, which is included as a separate event (Event 2). The number of participants for each event are as follows:

- **Total patients:** 38,052
  - **Event 1:** Prostate cancer incidence: 4,425 (12%)
  - **Event 2:** Death prior to prostate cancer incidence: 6,032 (16%)
  - **Censored:** 27,595 (73%)

There are 66 original features in the dataset. After one-hot encoding of categorical variables we obtained 105 feature columns used in the models:

- Trial entry / year 0 PSA level (1 feature) and DRE result (1 feature - 6 features after one-hot encoding)
- PLCO background questionnaire (64 features - 99 features after one-hot encoding) (demographics, smoking, (prostate) cancer family history, body type, NSAIDS, diseases, male specifics, prostate surgery)

The dataset had 13% missing values (prior to one-hot encoding), however all but 15 features had missingness of 5% or below. PSA level at year 0 had missingness of 10%. The remaining features with high missingness were mostly questions that were left out for large parts of the cohort, such as details around smoking and medical conditions if the patient had indicated that these conditions apply to him.

Missing values were imputed using the mode for binary/categorical features, and the mean for numerical features. All features were furthermore normalised to mean 0 and variance 1 prior to training the models.

For survival analysis, we set time 0 to be the time of the first PSA or DRE screen, whichever occurred later. The average exit months for the trial were 122 months.

A.2 SEER

The Surveillance, Epidemiology, and End Results Program (SEER)[2] is a publicly available dataset. We use an extracted cohort of 72,809 breast cancer patients from 1992 to 2007. The event of interest is death, which occured either from breast cancer, cardiovascular disease or other causes. Patients living at the end of the study were right-censored. The number of patients by event are as follows:

- **Total patients:** 72,809

[2]https://seer.cancer.gov/causespecific/

- **Event 1:** Death from breast cancer: 10,634 (15%)
- **Event 2:** Death from cardiovascular disease (CVD): 903 (1%)
- **Event 3:** Death from other causes: 4,484 (6%)
- **Censored:** 56,788 (78%)

SEER: Original features The SEER dataset contains 23 features:

- Demographics, including age, race, gender (8 features)
- Morphology information (lymphoma subtype, histological type, etc.), diagnostic information, therapy information, tumor size and tumor type (15 features)

Missing values were imputed using the mean value for real-valued features and the mode for categorical features. All features were furthermore normalised to mean 0 and variance 1 prior to training the models.

SEERsynth: Additional synthetic features In order to compare the performance of methods when there are a large number of irrelevant features, we also artificially extended the SEER dataset with up to 100 synthetic features. The additional features are constructed for each patient independently from both other features and the survival distribution (time and event). For each feature j we initially pick a parameter p_j, where

$$p_j \sim \text{Unif}[0, 1], j = 1, \ldots, 100. \tag{6}$$

Finally, feature values for each patient are drawn from the distribution Bernoulli(p_j). They therefore represent additional binary random noise irrelevant to the prediction problem. We choose binary features due to their common occurence in medical datasets either as indicators of medical conditions, or one-hot encodings of categorical features.

The SEERsynth dataset thus includes up to 123 features:

- Original SEER features (23 features)
- Synthetic features (up to 100 features)

The average exit months for the SEER and SEERsynth datasets were 127 months.

B HybridDeepHit$^+$

We develop a hybrid feature selection approach that uses the machine learning model itself to extract feature relevance values from the data as an alternative to filter methods. The procedure, using an adaptation of Breiman and Cutler's permutation importance [1], works as follows:

1. We initially train DeepHit$^+$ using all features.
2. For each feature we then randomly permute its validation set values, and run the pre-trained model to predict using the new permuted validation set. The importance of the feature is defined to be the difference in true model C-index performance (using original data) and new model performance (using the new data with the permuted feature column). Feature importances are therefore event-specific, depending on the event chosen for the performance calculation. We compute the importance values for all evaluation times, and report the average. Furthermore we average importance values over I permutations.
3. We finally choose the top m_k features according to their importance results, treating the number of features to be selected ($m_k, k = 1, \ldots, K$) as additional hyperparameters that are optimised with random search during training. The deep neural network architecture is as for FilterDeepHit$^+$ in Figure 1. The shared sub-network's input features are the common features that are selected for all events.

C Implementation and hyperparameter optimization

Fine-Gray We implemented Fine-Gray models using the `crr` function in the R package `cmprsk`[3], using default values for all function parameters. The model fails when there are singularities in the input data. We therefore remove linearly dependent columns from the dataset prior to training the model.

Random Survival Forest Random Survival Forests were implemented using the `rfsrc` function in the R package `randomForestSRC`[4]. The number of trees was chosen to be 1000, and all other parameters were set to defaults.

DeepHit We implemented DeepHit (as well as DeepHit$^+$ and all its extensions) using Python's `tensorflow`[5] package. The fixed model settings and layer sizes used were as described in [10], apart from early stopping, which was conducted using validation C-index performance, as opposed to validation loss.

The remaining hyperparameters β and σ were chosen by random search. This was conducted with 20 search iterations on the first cross-validation train/test split (almost amounting to exhaustive search given the small hyperparameter space). We determined the best hyperparameters based on average C-index performance across events and evaluation times on the validation dataset. During the random search procedure, values for the hyperparameters were chosen from the sets in Table 2. The second to fifth cross-validations used the hyperparameters determined from the first split in order to improve computational efficiency. The size of the validation dataset for random search and early stopping was 20% of the original training data.

Table 2: Random search hyperparameter choices for DeepHit

Hyperparameter	Set of choices
β: Weight of ranking loss	0.1, 0.3, 1, 3, 10
σ: Parameter for risk comparator η	0.1, 0.3, 1, 3, 10

DeepHit$^+$ Here we conduct additional hyperparameter random search for the number of layers and hidden nodes in each sub-network. In order to focus on these variables, we fix the hyperparameters β and σ to their optimal values determined by DeepHit's initial run on the same dataset. Given the larger hyperparameter space, we conduct 50 random search iterations on the first cross-validation, and apply the optimal parameters to all other train/test splits. The sets from which the hyperparameters were chosen are given in Table 3.

Table 3: Random search hyperparameter choices for DeepHit$^+$

Hyperparameter	Set of choices
n_s, number of shared layers:	1, 2, 3
h_s, nodes per hidden shared layer:	50, 100, 200
n_k, number of cause-specific layers:	1, 2, 3
h_k, nodes per hidden cause-specific layer:	50, 100, 200

Filter-, Sparse- and HybridDeepHit$^+$ These models are trained in the same way as DeepHit$^+$. However, as each extension adds additional parameters, we restrict the choices for network sizing compared to DeepHit$^+$ in an effort to keep the search space manageable within the computational budget. The sets from which these hyperparameters were chosen are given in Table 4. In order to reduce the search space for SparseDeepHit$^+$, we set the weighting for the shared sub-network's regularisation term to be $\gamma_s = \frac{1}{K}\sum_{k=1}^{K}\gamma_k$.

[3]https://cran.r-project.org/web/packages/cmprsk/

[4]https://cran.r-project.org/web/packages/randomForestSRC/

[5]https://www.tensorflow.org/

Table 4: Random search hyperparameter choices for Filter-, Sparse- and HybridDeepHit$^+$

Hyperparameter	Set of choices
n_s, number of shared layers:	1, 2
h_s, nodes per hidden shared layer:	50, 100
n_k, number of cause-specific layers:	1, 2
h_k, nodes per hidden cause-specific layer:	50, 100
m_k, number of features selected (Filter-, HybridDeepHit$^+$):	20, 40, 60
γ_k, regularisation loss weights (SparseDeepHit$^+$):	0.00001, 0.0001, 0.001

D Full results

We present full results for all algorithms and evaluation horizons in Tables 5 and 6.

Table 5: C-index performance of all models on PLCO. SD denotes standard deviation of the 5 train-test splits, Δ denotes C-index difference to DeepHit$^+$.

	Event	Event 1			Event 2		
		Mean	SD	Δ	Mean	SD	Δ
Horizon	**Algorithm**						
$\Delta t = 012$	**HybridDeepHit$^+$**	**0.936**	0.008	0.008	0.748	0.032	-0.012
	FilterDeepHit$^+$ (Anova)	0.935	0.009	0.007	0.753	0.028	-0.007
	RSF	0.934	0.006	0.007	0.716	0.040	-0.045
	FilterDeepHit$^+$ (SVM)	0.933	0.007	0.005	0.741	0.050	-0.019
	FilterDeepHit$^+$ (ReliefF)	0.931	0.007	0.004	0.747	0.034	-0.013
	SparseDeepHit$^+$	0.930	0.008	0.003	**0.760**	0.027	-0.000
	DeepHit$^+$	0.927	0.009	0.000	**0.760**	0.030	0.000
	DeepHit	0.906	0.013	-0.021	0.746	0.038	-0.014
	Fine-Gray	0.710	0.020	-0.218	**0.760**	0.028	-0.001
$\Delta t = 036$	**HybridDeepHit$^+$**	**0.890**	0.004	0.020	0.752	0.014	-0.006
	FilterDeepHit$^+$ (Anova)	0.888	0.006	0.017	0.755	0.015	-0.003
	FilterDeepHit$^+$ (SVM)	0.887	0.006	0.016	0.745	0.015	-0.013
	SparseDeepHit$^+$	0.886	0.005	0.016	**0.758**	0.011	0.000
	FilterDeepHit$^+$ (ReliefF)	0.885	0.005	0.014	0.749	0.017	-0.009
	RSF	0.884	0.006	0.013	0.742	0.016	-0.016
	DeepHit$^+$	0.871	0.007	0.000	**0.758**	0.011	0.000
	DeepHit	0.835	0.010	-0.036	0.746	0.012	-0.012
	Fine-Gray	0.666	0.021	-0.205	0.757	0.011	-0.001
$\Delta t = 060$	**HybridDeepHit$^+$**	**0.872**	0.006	0.027	0.751	0.015	-0.005
	FilterDeepHit$^+$ (Anova)	0.871	0.006	0.025	0.751	0.016	-0.005
	FilterDeepHit$^+$ (SVM)	0.870	0.004	0.025	0.744	0.017	-0.012
	SparseDeepHit$^+$	0.869	0.004	0.024	**0.757**	0.012	0.001
	FilterDeepHit$^+$ (ReliefF)	0.866	0.003	0.021	0.746	0.016	-0.010
	RSF	0.866	0.006	0.020	0.748	0.011	-0.009
	DeepHit$^+$	0.845	0.008	0.000	0.756	0.012	0.000
	DeepHit	0.807	0.012	-0.038	0.744	0.013	-0.013
	Fine-Gray	0.647	0.013	-0.198	0.755	0.013	-0.002
$\Delta t = 120$	**HybridDeepHit$^+$**	**0.821**	0.005	0.032	0.738	0.002	-0.006
	SparseDeepHit$^+$	0.820	0.006	0.031	0.745	0.005	0.001
	FilterDeepHit$^+$ (SVM)	0.820	0.006	0.031	0.733	0.006	-0.011
	FilterDeepHit$^+$ (Anova)	0.819	0.007	0.030	0.739	0.003	-0.005
	FilterDeepHit$^+$ (ReliefF)	0.814	0.006	0.025	0.732	0.004	-0.012
	RSF	0.813	0.005	0.025	0.737	0.007	-0.007
	DeepHit$^+$	0.789	0.008	0.000	0.744	0.004	0.000
	DeepHit	0.749	0.011	-0.039	0.726	0.007	-0.017
	Fine-Gray	0.622	0.008	-0.167	**0.746**	0.005	0.003

Table 6: C-index performance of all models on SEERsynth (100 additional synthetic features). SD denotes standard deviation of the 5 train-test splits, Δ denotes C-index difference to DeepHit$^+$.

	Event	Event 1			Event 2			Event 3		
Horizon	**Algorithm**	Mean	SD	Δ	Mean	SD	Δ	Mean	SD	Δ
$\Delta t = 012$	**RSF**	**0.852**	0.015	0.029	0.616	0.081	-0.071	0.747	0.046	-0.003
	HybridDeepHit$^+$	0.848	0.016	0.026	0.692	0.072	0.005	0.763	0.023	0.013
	SparseDeepHit$^+$	0.846	0.022	0.024	0.698	0.089	0.011	0.763	0.029	0.013
	FilterDeepHit$^+$ (Anova)	0.840	0.018	0.018	0.662	0.124	-0.024	0.760	0.026	0.010
	FilterDeepHit$^+$ (ReliefF)	0.836	0.012	0.014	**0.709**	0.084	0.022	**0.769**	0.018	0.019
	DeepHit$^+$	0.822	0.017	0.000	0.687	0.057	0.000	0.750	0.019	0.000
	FilterDeepHit$^+$ (SVM)	0.809	0.045	-0.013	0.696	0.094	0.009	0.748	0.030	-0.003
	DeepHit	0.798	0.021	-0.025	0.660	0.048	-0.027	0.736	0.024	-0.014
	Fine-Gray	0.793	0.020	-0.029	0.642	0.061	-0.045	0.701	0.015	-0.050
$\Delta t = 036$	**RSF**	**0.792**	0.013	0.043	0.662	0.042	-0.009	0.704	0.024	0.007
	HybridDeepHit$^+$	0.786	0.012	0.037	0.690	0.036	0.019	0.714	0.022	0.016
	SparseDeepHit$^+$	0.782	0.024	0.034	0.686	0.024	0.015	**0.715**	0.021	0.017
	FilterDeepHit$^+$ (Anova)	0.777	0.013	0.029	0.647	0.049	-0.025	**0.715**	0.021	0.017
	FilterDeepHit$^+$ (ReliefF)	0.773	0.026	0.024	**0.702**	0.024	0.031	0.714	0.019	0.017
	FilterDeepHit$^+$ (SVM)	0.761	0.020	0.013	0.695	0.022	0.024	0.699	0.029	0.002
	DeepHit$^+$	0.749	0.006	0.000	0.671	0.016	0.000	0.697	0.023	0.000
	DeepHit	0.725	0.008	-0.024	0.652	0.036	-0.019	0.697	0.029	-0.000
	Fine-Gray	0.697	0.009	-0.052	0.654	0.021	-0.017	0.674	0.013	-0.023
$\Delta t = 060$	**RSF**	**0.770**	0.006	0.053	0.680	0.041	0.015	0.693	0.016	0.011
	HybridDeepHit$^+$	0.763	0.008	0.046	0.682	0.031	0.018	0.695	0.012	0.013
	SparseDeepHit$^+$	0.758	0.020	0.041	0.678	0.032	0.014	**0.697**	0.013	0.015
	FilterDeepHit$^+$ (Anova)	0.754	0.007	0.037	0.651	0.036	-0.014	**0.697**	0.013	0.016
	FilterDeepHit$^+$ (ReliefF)	0.749	0.028	0.032	**0.698**	0.022	0.033	0.696	0.014	0.014
	FilterDeepHit$^+$ (SVM)	0.741	0.016	0.024	0.695	0.029	0.030	0.685	0.014	0.003
	DeepHit$^+$	0.717	0.007	0.000	0.665	0.008	0.000	0.682	0.014	0.000
	DeepHit	0.695	0.010	-0.022	0.648	0.033	-0.016	0.682	0.014	-0.000
	Fine-Gray	0.667	0.007	-0.050	0.662	0.035	-0.002	0.672	0.011	-0.010
$\Delta t = 120$	**RSF**	**0.746**	0.004	0.074	**0.670**	0.031	0.036	**0.697**	0.012	0.015
	FilterDeepHit$^+$ (Anova)	0.729	0.003	0.057	0.635	0.028	0.001	0.693	0.010	0.010
	HybridDeepHit$^+$	0.724	0.009	0.052	0.658	0.026	0.024	0.688	0.010	0.005
	SparseDeepHit$^+$	0.719	0.013	0.047	0.652	0.032	0.018	0.696	0.009	0.014
	FilterDeepHit$^+$ (ReliefF)	0.706	0.029	0.034	0.665	0.019	0.031	0.692	0.010	0.009
	FilterDeepHit$^+$ (SVM)	0.701	0.011	0.029	0.660	0.023	0.026	0.685	0.009	0.003
	DeepHit$^+$	0.672	0.004	0.000	0.634	0.015	0.000	0.683	0.009	0.000
	DeepHit	0.661	0.010	-0.011	0.618	0.021	-0.016	0.684	0.010	0.002
	Fine-Gray	0.637	0.005	-0.036	0.641	0.024	0.007	0.686	0.013	0.004

E Feature rankings

Table 7 gives an example of the feature ranking output of FilterDeepHit$^+$. This table, as well as all other feature ranking tables in this section, exhibits results from the first of the five cross-validation train/test splits. For SEERsynth, synthetic features are denoted as *Synth*** (** a placeholder for the feature number). It can be seen that on PLCO, the medically relevant PSA level and DRE result are ranked top, whilst on SEERsynth irrelevant synthetic features are correctly filtered out.

Table 7: Top 10 features by ANOVA p-value for Event 1 on PLCO and SEERsynth

PLCO Feature	p-value
PSALevel	7.5e-154
DRE_AbnormSuspi	9.6e-47
DRE_Negative	1.7e-13
Age	3.0e-13
NumRelativesPrCancer	2.1e-06
FamHistPrCan_No	3.5e-06
FamHistPrCan_Yes	8.0e-06
Occ_Retired	1.4e-04
Occ_Working	3.2e-04
Race_BlackNonHisp	3.7e-04

SEERsynth Feature	p-value
ACJJ Stage	1.1e-233
Tumor Marker	2.5e-92
EOD 10 Extent	2.8e-48
Positive cytology	1.3e-33
In situ/Malignant Tumors	2.2e-29
Histology ICD-O-3	2.3e-25
Bilateral	1.4e-17
Married	4.4e-17
Single	6.6e-13
Lymphod node	7.2e-12

Since the weights of SparseDeepHit$^+$'s first layers are regularized, they also represent an indication of feature relevance. As an example, we present the top 10 for each sub-network for

PLCO in Table 8, denoting input features as *Shared*** (** a placeholder for the node number) when a cause-specific sub-network utilizes a feature from the shared connection $f_s(\mathbf{x})$. It can be seen that whilst PSA level and age at trial entry are dominating, further analysis of the weights is more difficult as shared features are highly ranked, but not immediately attributable to original features from the dataset.

Table 8: Top 10 features by first layer weights in SparseDeepHit$^+$'s sub-networks on PLCO. The ranking is determined by the weight's absolute value.

Feature	Shared
PSALevel	0.551
EnlargedProsBPH	0.141
Marital_Married	0.137
AgeInflamedPros	-0.127
UrinateNight_Never	0.122
Prostatectomy	-0.119
Marital_NeverMarr	-0.117
PersCanHist_Unkn	-0.116
Hypertension	0.114
GallbladderStones	-0.111

Feature	Event 1
PSALevel	-0.577
Age	0.243
Shared20	0.201
Shared49	0.198
Shared42	-0.192
Shared38	0.191
Shared25	0.185
Cig_Current	-0.182
Hypertension	-0.173
HeartAttack	0.173

Feature	Event 2
PSALevel	0.765
Age	0.342
DRE_AbnormSuspi	0.272
PersCanHist_No	-0.239
Shared29	-0.238
Shared42	0.237
DurationSmokedCig	0.234
Shared35	0.213
Occ_Disabled	0.204
SmokingPackYears	0.204

HybridDeepHit$^+$'s feature importance values are by construction a direct representation of their relevance for the model's prediction. As can be seen in Table 9, the top features overlap with those chosen by the ANOVA filter feature selection from Table 7.

Table 9: Top 10 features by permutation importance for Event 1 on PLCO and SEERsynth

PLCO Feature	Importance
PSALevel	0.2692
DRE_AbnormSuspi	0.0085
Cig_Former	0.0071
DRE_NotDoneExp	0.0037
DurationSmokedCig	0.0037
Race_Asian	0.0029
HadProstateBiopsy	0.0028
DRE_Negative	0.0026
Race_BlackNonHisp	0.0022
ProsSurgeries_No	0.0020

SEERsynth Feature	Importance
ACJJ Stage	0.0794
Lymphod node	0.0723
Tumor Marker	0.0312
Single	0.0043
Histology ICD-O-3	0.0029
In situ/Malignant Tumors	0.0019
Synth22	0.0018
Positive cytology	0.0015
Synth96	0.0015
Positive histology	0.0015