RNN-SURV: a Deep Recurrent Model for Survival Analysis

Eleonora Giunchiglia[1](✉), Anton Nemchenko[2], and Mihaela van der Schaar[3,2,4]

[1] DIBRIS, Università di Genova, Italy
[2] Department of Electrical and Computer Engineering, UCLA, USA
[3] Department of Engineering Science, University of Oxford, UK
[4] Alan Turing Institute, London, UK
eleonora.giunchiglia@icloud.com

Abstract. Current medical practice is driven by clinical guidelines which are designed for the "average" patient. Deep learning is enabling medicine to become personalized to the patient at hand. In this paper we present a new recurrent neural network model for personalized survival analysis called RNN-SURV. Our model is able to exploit censored data to compute both the risk score and the survival function of each patient. At each time step, the network takes as input the features characterizing the patient and the identifier of the time step, creates an embedding, and outputs the value of the survival function in that time step. Finally, the values of the survival function are linearly combined to compute the unique risk score. Thanks to the model structure and the training designed to exploit two loss functions, our model gets better concordance index (C-index) than the state of the art approaches.

1 Introduction

Healthcare is moving from a population-based model, in which the decision making process is targeted to the "average" patient, to an individual-based model, in which each diagnosis is based on the features characterizing the given patient. This process has been boosted by the recent developments in the Deep Learning field, which has been proven to not only get impressive results in its traditional areas, but also to perform very well in medical tasks.

In particular, in the medical field, the study of the *time-to-event*, i.e., the expected duration of time until one or more events happen, such as death or recurrence of a disease, is of vital importance. Nevertheless, it is often made more complicated by the presence of *censored data*, i.e., data in which the information about the time-to-event is incomplete, as it happens, e.g., when a patient drops a clinical trial. Traditionally, these issues are tackled in a field called Survival Analysis, a branch of statistics in which special models have been proposed to predict the time-to-event exploiting censored data, while only a few deep learning approaches have such an ability (e.g., [13, 28]). About the latter, it is interesting to note that most of the encountered deep learning approaches are based on feedforward neural networks and, at least so far, there does not seem to exist published results deploying recurrent neural networks despite the sequential nature of the problem.

In this paper we present a new recurrent neural network model handling censored data and computing, for each patient, both a survival function and a unique risk score. The survival function is computed by considering a series of binary classifications problems each leading to the estimation of the survival probability in a given interval of time, while the risk score is obtained through the linear combination of the estimates. RNN-SURV three main features are:

1. its ability to model the possible time-variant effects of the covariates,
2. its ability to model the fact that the survival probability estimate at time t is function of each survival probability estimate at $t' : t' < t$, and
3. its ability to compute a highly interpretable risk score.

The first two are given by the recurrent structure, while the last is given by the linear combination of the estimates.

RNN-SURV is tested on three small publicly available datasets and on two large heart transplantation datasets. On these datasets RNN-SURV performs significantly better than the state of the art models, always resulting in a higher C-index than the state of the art models (up to 28.4%). We further show that if we simplify the model we always get worse performances, hence showing the significance of RNN-SURV different features.

This paper is structured as follows. We start with the analysis of the related work (Section 2), followed by the background about Survival Analysis (Section 3). Then, we present of our model (Section 4), followed by the experimental analysis (Section 5), and finally the conclusions (Section 6).

2 Related Work

The problem of survival analysis has attracted the attention of many machine learning scientists, giving birth to models such as random survival forest [11], dependent logistic regressors [26], multi-task learning model for survival analysis [17], semi-proportional hazard model [27] and support vector regressor for censored data [21], all of which not based on neural networks.

Considering the works that have been done in the field of Survival Analysis using Deep Learning techniques, these can be divided in three main subcategories, that stemmed from just as many seminal papers:
(1) Faraggi and Simon [7] generalized Cox Proportional Hazards model (CPH) [5] allowing non-linear functions instead of the traditional linear combinations of covariates by modeling the relationship between the input covariates and the corresponding risk with a single hidden layer feedforward neural network. This work has been later resumed in [13] and [28]. Contrarily to RNN-SURV, CPH and the models [13] and [28] assume time-invariant effects of the covariates.
(2) Liestbl, Andersen and Andersen [18] subdivided time into K intervals, assumed the hazard to be constant in each interval and proposed a feedforward

neural network with a single hidden layer that for each patient outputs the conditional event probabilities $p_k = P(T \geq t_k | T \geq t_{k-1})$ for $k = 1, ..., K$, T being the time-to-event of the given patient. This work was then expanded in [2], but even in this later work the value of the estimate p_{k-1} for a given patient is not exploited for the computation of the estimate p_k for the same patient. On the contrary, RNN-SURV, thanks to the presence of recurrent layers, is able to capture the intrinsic sequential nature of the problem.
(3) Buckley and James [4] developed a linear regression model that deals with each censored data by computing its most likely value on the basis of the available data. This approach was then generalized using neural networks in various ways (e.g., [6]). Unlike RNN-SURV, in [4] and in the following ones, estimated and known data are treated in the same way during the regression phase.

3 Background on Survival Analysis

Consider a patient i, we are interested in estimating the duration T_i of the interval in between the event of interest for i and the time t_0 at which we start to measure time for i. We allow for *right censored* data, namely, data for which we do not know when the event occurred, but only that it did not occur before a censoring time C_i. The *observed time* Y_i is defined as $Y_i = \min(T_i, C_i)$, and each datapoint corresponds to the pair (Y_i, δ_i) where $\delta_i = 0$ if the event is censored (in which case $Y_i = C_i$) and $\delta_i = 1$ otherwise.

In Survival Analysis, the standard functions used to describe T_i are the survival function and the hazard function [15].

1. The *survival function* $S_i(t)$ is defined as:
$$S_i(t) = Pr(T_i > t) \tag{1}$$
with $S_i(t_0) = 1$.
2. The *hazard function* $h_i(t)$ is defined as:
$$h_i(t) = \lim_{dt \to 0} \frac{Pr(t \leq T_i < t + dt \mid T_i \geq t)}{dt}. \tag{2}$$

Further, in order to offer a fast understanding of the conditions of the patient, a common practice of the field is to create a risk score r_i for each patient i: the higher the score the higher the risk of the occurrence of the event of interest.

4 RNN-SURV

In order to transform the survival analysis problem in a series of binary decision problems, we assume that the maximal observed time is divided into K intervals $(t_0, t_1], \dots, (t_{K-1}, t_K]$ and that the characteristic function modeling T_i is constant within each interval $(t_{k-1}, t_k]$ with $k = 1, \dots, K$. Given a patient i, the purpose of our model is to output both an estimate $\hat{y}_i^{(k)}$ of the survival probability S_i for the kth time interval and a risk score r_i.

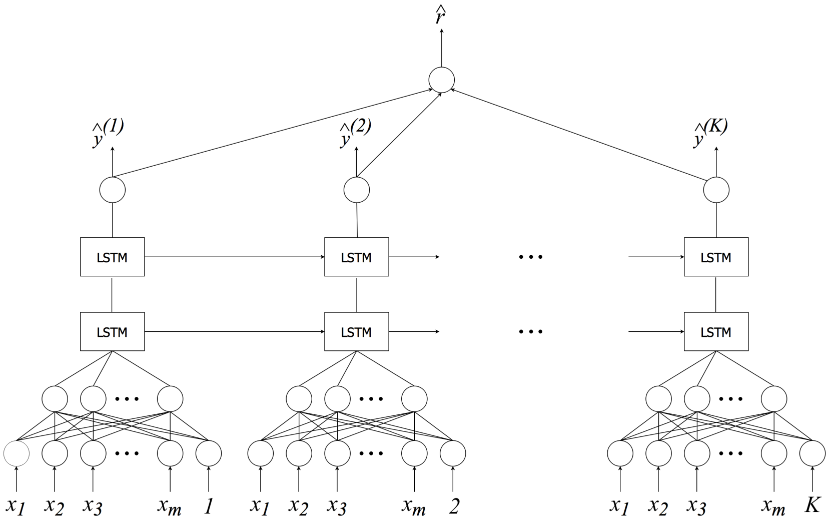


Fig. 1. RNN-SURV with $N_1 = 2$ feedforward layers, followed by $N_2 = 2$ recurrent layers.

4.1 The Structure of the Model

The overall structure of RNN-SURV is represented in Figure 1 and is described and motivated below:

1. the input of each layer is given by the features $\mathbf{x}_i$ of each patient i together with the time interval identifier k. Thanks to this input, RNN-SURV is able to capture the time-variant effect of each feature over time,
2. taking the idea from the natural language processing field, the input is then elaborated by N_1 embedding layers. Thanks to the embeddings we are able to create a more meaningful representation of our data, and
3. the output of the embedding layers is then passed through N_2 recurrent layers and a sigmoid non-linearity. This generates the estimates $\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(K)}$ from which we can compute the risk score with the following equation:

$$\hat{r}_i = \sum_{k=1}^{K} w_k \hat{y}_i^{(k)} \tag{3}$$

   where w_k for $k = 1, \ldots, K$ are the parameters of the last layer of RNN-SURV. Thanks to the linear combination, the risk score, whose quality is evaluated with the C-index [9], is highly interpretable.

Further, in order to handle the vanishing gradient problem, the feedforward layers use the ReLU non-linearity [19], while the recurrent layers are constituted of LSTM cells [10], which are defined as:

$$\begin{aligned}
\begin{pmatrix}\mathbf{i}_t\\ \mathbf{f}_t\\ \mathbf{o}_t\\ \mathbf{g}_t\end{pmatrix} &= \begin{pmatrix}\sigma(\mathbf{W}_i[\mathbf{w}_t,\mathbf{h}_{t-1}]+\mathbf{b}_i)\\ \sigma(\mathbf{W}_f[\mathbf{w}_t,\mathbf{h}_{t-1}]+\mathbf{b}_f)\\ \sigma(\mathbf{W}_o[\mathbf{w}_t,\mathbf{h}_{t-1}]+\mathbf{b}_o)\\ f(\mathbf{W}_g[\mathbf{w}_t,\mathbf{h}_{t-1}]+\mathbf{b}_g)\end{pmatrix}\\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \mathbf{g}_t\\
\mathbf{h}_t &= \mathbf{o}_t * f(\mathbf{c}_t).
\end{aligned} \tag{4}$$

4.2 Training

Since the neural network predicts both the discrete survival function and the risk score for each datapoint, it is trained to jointly minimize two different loss functions:

1. The first one is a modified cross-entropy function able to take into account the censored data, defined as:

$$\mathcal{L}_1 = -\sum_{k=1}^{K}\sum_{i\in U_k}\left[\mathbb{I}(Y_i > t_k)\log\hat{y}_i^{(k)} + (1-\mathbb{I}(Y_i > t_k))\log(1-\hat{y}_i^{(k)})\right] \tag{5}$$

   where $U_k = \{i \mid \delta_i = 1 \text{ or } C_i > t_k\}$ represents the set of individuals that are uncensored throughout the entire observation time or for which censoring has not yet happened at the end of the kth time interval.
2. The second one is an upper bound of the negative C-index [23] defined as:

$$\mathcal{L}_2 = -\frac{1}{|\mathcal{C}|}\sum_{(i,j)\in\mathcal{C}}\left[1+\left(\frac{\log\sigma(\hat{r}_j - \hat{r}_i)}{\log 2}\right)\right] \tag{6}$$

   where $\mathcal{C}$ is the set of pairs $\{(i,j) \mid \delta_i = 1 \text{ and } (Y_i \le Y_j)\}$. The advantage of minimizing (6) instead of the negative C-index is that the former still leads to good results [23], and the latter is far more expensive to compute and would have made the experimental evaluation impractical.

The two losses $\mathcal{L}_1$ and $\mathcal{L}_2$ are then linearly combined, with the hyperparameters of the sum optimized during the validation phase.

In order to avoid overfitting, we apply dropout to both the feedforward layers [22] and to the recurrent layers [8], together with a holdout-based early stopping as described in [20]. Further, we add $L2$-regularization to the linear combination of the losses. The entire neural network is trained using mini-batching and Adam optimizer [14].

5 Experimental Analysis

All our experiments are conducted on two large datasets, UNOS Transplant and UNOS Waitlist, from the United Network for Organ Sharing (UNOS)[1] and on

[1] https://www.unos.org/data/

Table 1. Datasets description

Dataset	Num. Features	Num. patients	(%) Censored	Missing data
UNOS Transplant	53	60400	51.3	Yes
UNOS Waitlist	27	36329	48.9	Yes
NWTCO	9	4028	85.8	No
FLCHAIN	26	7874	72.5	Yes
AIDS2	12	2843	38.1	No

three publicly available, small datasets, AIDS2, FLCHAIN, NWTCO. [2] In each experiment we deploy 60/20/20 division into training, validation and test sets and the early stopping is configured as a no validation gain for 25 consecutive epochs. The main characteristics of these datasets are shown in Table 1, while the structure of RNN-SURV for each dataset is shown in Table 2. The performances of our model are measured using the C-index [9].[3]

Table 2. Structure of the model for each experiment.

	UNOS Transplant	UNOS Waitlist	NWTCO	FLCHAIN	AIDS2
Number FF layers	2	2	3	3	2
Number Recurrent layers	2	2	2	2	2
Number neurons I FF layer	53	33	18	45	22
Number neurons II FF layer	51	35	18	40	25
Number neurons III FF layer	-	-	18	35	-
LSTM state size	55	26	17	32	15

5.1 Preprocessing

Our datasets present missing data and thus they require a preprocessing phase. UNOS Transplant and UNOS Waitlist contain data about patients that registered in order to undergo heart transplantation during the years from 1985 to 2015. In particular UNOS Transplant contains data about patients who have already undergone the surgery, while UNOS Waitlist contains data about patients who are still waitlisted. From the complete datasets, we discard 12 features that can be obtained only after transplantation and all the features for which more than 10% of the patients have missing information. In order to deal with the missing data on the remaining 53 and 27 features, we conduct 10 multiple imputations using Multiple Imputation by Chained Equations (MICE) [24].

The three small datasets contain data about:

1. NWTCO: contains data from the National Wilm's Tumor Study [3],

[2] https://vincentarelbundock.github.io/Rdatasets/datasets.html.

[3] Implementation by LIFELINES package.

2. FLCHAIN: contains half of the data data collected during a study [16] about the possible relationship between serum FLC and mortality, and
3. AIDS2: contains data on patients diagnosed with AIDS in Australia [25].

For these datasets, we complete the missing data using the mean value for the continuous features and using the most recurrent value for the categorical ones. Once complete the missing data, we then use one-hot encoding for the categorical features and we standardize each feature so that each has mean $\mu = 0$ and variance $\sigma = 1$.

5.2 Comparison with Other Models

We have compared rnn-surv with the two traditional Survival Analysis models, CPH and Aalen Additive Hazards model (AAH) [1], and with three recent models that try to conjugate Machine Learning with Survival Analysis: rfs [11], deep-surv [13] and mtlsa [17]. Both CPH and AAH have been implemented using the lifelines package [4], while we deployed the randomForestSRC package[5] for rfs, the deepsurv package[6] for deep-surv and the mtlsa package[7] for mtlsa. The results shown in Table 3 are obtained using k-fold cross validation (with $k = 5$). As it can be seen from the table, rnn-surv outperforms the other models in all the datasets. In particular, the biggest improvements are obtained with respect to mtlsa, with a peak of 28.4% on the FLCHAIN dataset.

Table 3. Performances, in terms of C-index, of rnn-surv, CPH, AAH, deep-surv, rfs and mtlsa together with the 95% confidence interval for the mean C-index. The * indicates a p-value < 0.05 while ** < 0.01.

	UNOS Transplant	UNOS Waitlist	NWTCO	FLCHAIN	AIDS2
CPH	0.566** (0.565 - 0.567)	0.642** (0.637 - 0.647)	0.706 (0.687 - 0.725)	0.883* (0.879 - 0.887)	0.558 (0.546 - 0.570)
AAH	0.561** (0.557 - 0.565)	0.636** (0.632 - 0.640)	0.710 (0.601 - 0.719)	0.885 (0.879 - 0.891)	0.557 (0.542 - 0.572)
deep-surv	0.566** (0.560 - 0.572)	0.645* (0.638 - 0.652)	0.706 (0.686 - 0.726)	0.835 (0.774 - 0.896)	0.558 (0.532 - 0.584)
rfs	0.563** (0.561 - 0.565)	0.646* (0.642 - 0.650)	0.663* (0.648 - 0.678)	0.828 (0.765 - 0.891)	0.501** (0.489 - 0.513)
mtlsa	0.484** (0.480 - 0.488)	0.529** (0.525 - 0.533)	0.595* (0.567 - 0.623)	0.696** (0.688 - 0.704)	0.520* (0.500 - 0.540)
rnn-surv	**0.587** **(0.583 - 0.591)**	**0.656** **(0.652 - 0.660)**	**0.724** **(0.697 - 0.751)**	**0.894** **(0.886 - 0.902)**	**0.573** **(0.553 - 0.593)**

[4] https://github.com/CamDavidsonPilon/lifelines.

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

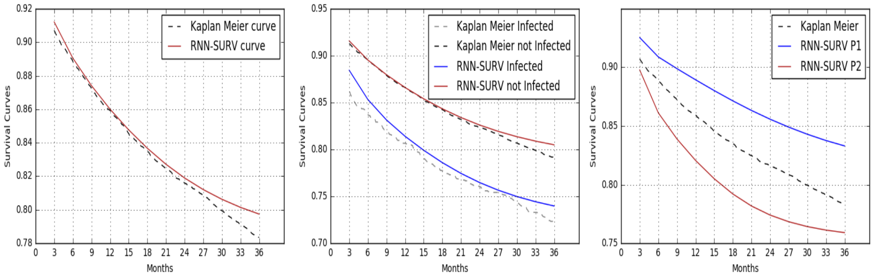
[6] https://github.com/jaredleekatzman/DeepSurv

[7] https://github.com/yanlirock/MTLSA

5.3 Estimating the Survival Curves

To further demonstrate the good results obtained by RNN-SURV, in Figure 2 we show some of the survival curves obtained in largest dataset available, the UNOS Transplant dataset.

Figure 2 shows that our model is able to capture the average trend of the survival curves, both for the whole population and for subsets of it. Further, RNN-SURV demonstrates to have a great discriminative power: it is able to plot a unique survival function for each patient and, as it is shown in Figure 2(c), the survival curves can be very different one from another and from the average survival curve.



(a) (b) (c)

Fig. 2. Performances of RNN-SURV on UNOS Transplant dataset on a 36 months horizon on the test set. Figure 2(a): average Survival Function obtained with RNN-SURV and Kaplan-Meier curve [12]. Figure 2(b): average Survival Functions obtained with RNN-SURV and Kaplan-Meier curves for two subgroups of patients: patients who experienced an infection and patients who did not. Figure 2(c): Kaplan-Meier curve together with the survival curves of two different patients (P1: Patient 1, P2: Patient 2).

5.4 Analysis of the Model

We now analyze how the different main components of RNN-SURV contribute to its good performances. In particular, we consider the model without the three main features of the model:

1. We first consider the case in which we do not have the feedforward layers, i.e., with $N_1 = 0$;
2. Then the case in which the interval identifier k as input to the feedforward layer is always set to 1;
3. Finally the case in which the model has only one likelihood, i.e., $\mathcal{L}_2$.

The C-index of the various versions and of the complete model on the different datasets are shown in Table 4. In the Table the best results are in bold, while the worst results are underlined. As it can be seen, the best performances are always obtained by the complete model, meaning that all the different components have a positive contribution. Interestingly, the worst performances are obtained when we disable the $\mathcal{L}_1$ score on the large datasets and the feedforward layers in the small ones. The explanation for the very positive contribution of using both the $\mathcal{L}_1$ and $\mathcal{L}_2$ scores on the two large datasets is that $\mathcal{L}_1$ allows to take into account the intermediate performances of the network when computing $\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(K)}$. On the other hand, for the small datasets, the positive contribution of using the two scores is superseded by the feedforward layers and this can be explained by the characteristics of the datasets presenting a majority of discrete features.

Table 4. Performances, in terms of C-index, of the complete model compared with its incomplete versions.

Dataset	without k input	without $\mathcal{L}_1$	without FF	RNN-SURV
UNOS Transplant	0.583	0.501	0.562	**0.587**
UNOS Waitlist	0.653	0.516	0.623	**0.656**
NWTCO	0.683	0.665	0.578	**0.724**
FLCHAIN	0.874	0.874	0.865	**0.894**
AIDS2	0.558	0.542	0.535	**0.573**

6 Conclusions

In this paper we have presented RNN-SURV: a new recurrent neural network model for predicting a personalized risk score and survival probability function for each patient in presence of censored data. The proposed model has three main distinguishing features, each having a positive impact on the performances on two large and three small, publicly available datasets. Our experiments show that RNN-SURV always performs much better than competing approaches when considering the C-index, improving the state of the art up to 28.4%.

References

1. Aalen, O.: A Model for Nonparametric Regression Analysis of Counting Processes, pp. 1–25. Springer New York (1980)
2. Biganzoli, E., Boracchi, P., Mariani, L., Marubini, E.: Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Statistics in Medicine **17**, 1169–1186 (1998)
3. Breslow, N.E., Chatterjee, N.: Design and analysis of two-phase studies with binary outcome applied to wilm's tumour prognosis. Applied Statistics **48** (1999)
4. Buckley, J., James, I.: Linear regression with censored data. Biometrika **66**(3), 429–436 (1979)
5. Cox, D.R.: Partial likelihood. Biometrika **62**(2), 269 (1975)
6. Dezfouli, H.N.: Improving gastric cancer outcome prediction using time-point artificial neural networks models. Cancer Informatics **16** (February 2017)
7. Faraggi, D., Simon, R.: A neural network model for survival data. Statistics in medicine **14**(1), 73–82 (1995)

8. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: 29th NIPS. pp. 1019–1027 (2016)
9. Harrell, F.J., Califf, R., Pryor, D., Lee, K., Rosati, R.: Evaluating the yield of medical tests. JAMA **247**(18), 2543–2546 (1982)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735
11. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S.: Random survival forests. The annals of applied statistics pp. 841–860 (2008)
12. Kaplan, E.L., Meier, P.: Non parametric estimation from incomplete observations. Journal of the American Statistical Association (1958)
13. Katzman, J., Shaham, U., Cloninger, A., Bates, J., Jiang, T., Kluger, Y.: Deep survival: A deep Cox proportional hazards network. CoRR (2016)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014), http://arxiv.org/abs/1412.6980
15. Klein, J.P., Moeschberger, M.L.: Survival Analysis Techniques for Censored and Truncated Data. Springer-Verlag New York, Inc., second edn. (2003)
16. Kyle, R., Therneau, T., Rajkumar, S.V., Larson, D., Plevak, M., Offord, J., Dispenzieri, A., Katzmann, J., Melton, L.: Use of monclonal serum immunoglobulin free light chains to predict overall survival in the general population. New England J Medicine **354**, 1362–1369 (2006)
17. Li, Y., Wang, J., Ye, J., Reddy, C.K.: A multi-task learning formulation for survival analysis. In: 22nd ACM SIGKDD. pp. 1715–1724. KDD '16, ACM, NY, USA (2016)
18. Liestbl, K., Andersen, P.K., Andersen, U.: Survival analysis and neural nets. Statistics in medicine **13**(12), 1189–1200 (1994)
19. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: 27th ICML. pp. 807–814 (2010)
20. Prechelt, L.: Early Stopping — But When?, pp. 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_5
21. Shivaswamy, P.K., Chu, W., Jansche, M.: A support vector approach to censored targets. In: Proceedings of 7th IEEE ICDM. pp. 655–660. ICDM 2007, IEEE Computer Society (2007). https://doi.org/10.1109/ICDM.2007.93
22. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
23. Steck, H., Krishnapuram, B., Dehing-oberije, C., Lambin, P., Raykar, V.C.: On ranking in survival analysis: Bounds on the concordance index. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) 20th NIPS, pp. 1209–1216. Curran Associates, Inc. (2008)
24. Van Buuren, S., Oudshoorn, K.: Flexible mutlivariate imputation by MICE. Leiden: TNO (1999)
25. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. Springer (2002)
26. Yu, C.N., Greiner, R., Lin, H.C., Baracos, V.: Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) 24th NIPS, pp. 1845–1853. Curran Associates, Inc. (2011)
27. Zhang, J., Chen, L., Vanasse, A., Courteau, J., Wang, S.: Survival prediction by an integrated learning criterion on intermittently varying healthcare data. In: 30th AAAI. pp. 72–78. AAAI 2016, AAAI Press (2016)
28. Zhu, X., Yao, J., Huang, J.: Deep convolutional neural network for survival analysis with pathological images. In: IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016, Shenzhen, China. pp. 544–547 (2016)