ELSEVIER

Computational Statistics & Data Analysis 34 (2000) 243–257

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Comparison of the performance of neural network methods and Cox regression for censored survival data ☆

Anny Xiang[a,*], Pablo Lapuerta[b], Alex Ryutov[a], Jonathan Buckley[a], Stanley Azen[a]

[a] *Statistical Consultation and Research Center, Department of Preventive Medicine, Keck School of Medicine, University of Southern California, 1540 Alcazar Street, CHP 218, Los Angeles, CA 90089-9010, USA*

[b] *Department of Outcomes Research, Pharmaceutical Research Institute, Bristol-Meyers Squibb, Princeton, NJ, USA*

Received 1 January 1999; received in revised form 1 November 1999

Abstract

Strategies that have been developed to extend NN prediction methods to accommodate right-censored data include methods due to Faraggi–Simon, Liestol–Andersen–Andersen, and a modification of the Buckley–James method. In a Monte Carlo simulation study, we evaluated the performance of all three NN methods with that of Cox regression models which included main effects and interactions, when interactions exist. Using the EPILOG PLUS® PROC NEURAL utility, feed-forward back-propagation networks were examined under nine designs representing a variety of experimental conditions which varied (a) the number of inputs and interactions, (b) the degree of censoring, (c) proportional vs. non-proportional hazards, and (d) sample size.

Minimization methods were implemented that efficiently determined optimal parameters. The C-index was used as a measure of performance. For the testing phase of the study, none of the NN methods outperformed Cox regression. Compared to Cox regression, the Faraggi–Simon, Buckley–James, and Liestol–Andersen–Andersen methods performed as well as Cox regression for 7, 5 and 1 of the nine designs, respectively. The effect on performance of modeling interactions in Cox regression, varying the number of intervals in the Liestol–Andersen–Andersen method, and varying the NN architecture are also presented. The results of our study suggest that NNs can serve as effective methods for modeling

☆ National Institutes of Health Grants: R03 HL55150 and NS RR33899.

* Corresponding author. Tel.: (323)-442-1810; fax: (323)-442-2993.
E-mail address: xiang@hsc.usc.edu (A. Xiang).

PII: S 0167-9473(99)00098-5

right-censored data. However, the performance of the NN is somewhat variable, depending on the underlying data structure.

Keywords: Neural network; Cox regression; Monte Carlo simulation

1. Introduction

Neural networks (NNs) are sophisticated computer programs that model the capabilities of the human brain by mimicking the structure and function of neurons in the brain (Bishop, 1995; Hornik, 1989). Utilizing principles of artificial intelligence, NNs permit the modeling of complex functional relationships (Warner and Misra, 1996). A NN consists of layers of nodes (analogous to neurons) linked by inter-connections (axons/dendrites), together with rules that specify how the output of each node is determined by input values from all nodes at the level below. A layered architecture of neurons in the brain can be used to provide progressively more abstract representation of input stimuli as the information is filtered through successive layers. Neural networks attempt to reproduce this effect, although most networks are limited in practice to three or four layers in total. Theoretical work suggests that NNs can consistently match or exceed the performance of statistical methods (Bishop, 1995; Hornik, 1989).

NNs are being used in the areas of prediction and classification of outcomes in medicine — areas where regression models have traditionally been used. In most applications, outcomes (termed *outputs* by users of NNs) are characterized by the presence or absence of an event. In this situation, a NN is regarded as an alternative to traditional logistic regression methods. However, in the situation where outcomes are characterized by the time to an event, the application of NNs to predict clinical events requires development of a strategy to address the time course of a disease process. In cases where the event of interest does not occur, outcomes are regarded as (right)-censored. Simple exclusion of censored observations from the available training set would limit the amount of data available for network development and could lead to significant biases in event predictions.

Several strategies have been developed to extend NN prediction methods to accommodate right-censored data. These are methods due to Faraggi and Simon (1995), Liestol et al. (1994), and a modification of the Buckley–James method (1979). Because Cox regression analysis is an accepted solution to the problem of analyzing censored data, the performance of Cox regression models, when compared to those of NNs, can provide a useful perspective on the utility of a NN approach.

In this paper, we report the results of a Monte Carlo simulation study that compares and evaluates the performance of all three NN methods with that for Cox regression. Using the EPILOG PLUS® PROC NEURAL utility developed by our group, feed-forward back-propagation networks are examined under a variety of experimental conditions. Minimization methods are implemented that efficiently determine optimal parameters. To evaluate the performance of the NN methods relative to Cox regression analysis, a generalized version of the receiver operating characteristic (ROC) curve, the C index, is used as a measure of performance.

2. Methods

2.1. Neural network methods for censored data

Three strategies for applying neural networks to censored data were compared. These methods are summarized as follows:

2.1.1. Faraggi–Simon method
The method due to Faraggi and Simon (1995) generalizes Cox regression to allow non-linear functions in place of the usual linear combination of covariates (i.e., inputs). Because a NN can be mathematically represented as a non-linear function of covariates, they proposed a Cox-like model in which the NN output is used in place of the linear function. This method retains the proportional hazard nature of the Cox model, but provides the ability to model complexities and interactions in the input data that simple Cox models would miss. The Faraggi–Simon method permits standard statistical methods for evaluating covariates; consequently, selecting the "best" model is straightforward.

2.1.2. Liestol–Andersen–Andersen method
For the method due to Liestol et al. (1994), survival times are grouped into time intervals during which the hazard is assumed to be constant and output nodes are established for each interval. Under the conditions of no hidden layers, and identical weights from the same input node to all output nodes, the NN is trained to predict the conditional event probabilities for each person, with results being identical to a "grouped" version of the Cox regression analysis of the same data. Non-constant hazards over time can be modeled when the weights to each output node are allowed to differ. Adding a hidden layer of nodes produces a form of Cox-NN hybrid with non-linear covariate effects; the degree of non-linearity depends on the number of hidden nodes and the choice of activation function.

2.1.3. Modified Buckley–James method
For the original Buckley–James method, as applied to linear regression, censored survival times are replaced by their expected values, based on the covariates and the residual Kaplan–Meier distribution about the fitted regression line (Buckley and James, 1979). Because the residual distribution is a function of the parameters, the estimation method is iterative and the expected values at each iteration are based on the current parameter estimates.

For the modified Buckley–James method, the NN outputs are used instead of the fitted regression line, and the residuals for estimating the Kaplan–Meier distribution are simply the difference between the observed outcomes and the neural network outputs. Thus, the original Buckley–James approach is generalized to the NN setting to determine the expected survival for all censored individuals (based on the current weight matrix) and to substitute the expected value for the censored value when determining and back-propagating the error. Specifically, at each iteration NN predictions are compared to the actual values and the differences (residuals) are used to calculate a Kaplan–Meier-type curve. Based on the residual distribution (as reflected in the Kaplan–Meier curve), the expected survival time for any person who was censored is estimated using the same approach as in the original Buckley–James method (Buckley and James, 1979).

2.2. Minimization Algorithms

For a description of the following minimization algorithms, see the text by Bishop (1995). In their paper, Faraggi and Simon recommended and utilized the Newton–Raphson algorithm to maximize the partial likelihood function and approximate both the first and second derivatives numerically (Faraggi and Simon, 1995). In contrast, Liestol and coworkers recommended and utilized the gradient algorithm to minimize the error function via back propagation (Liestol et al., 1994). For the Newton–Raphson algorithm, the numerical approximation of the derivatives can lead to large round-up errors and can become computationally prohibitive for large networks. Although the gradient algorithm takes less time for a single iteration step, it requires many iterations and consequently more computing time.

For these reasons, we evaluated a limited memory quasi-Newton minimization algorithm for all three methods under study (see Bishop, 1995, pp. 289–290). This quasi-Newton minimization algorithm builds up an approximation to the inverse Hessian matrix over a number of steps, and uses a “line-search” to detect the learning rate. As a result, the quasi-Newton algorithm requires less computer memory and converges much faster than the other algorithms.

In preliminary evaluations, we found that the quasi-Newton algorithm performed very well for the Liestol–Andersen–Andersen and Buckley–James methods. However, we found that it was not a good approximation for non-quadratic error functions associated with the Faraggi–Simon method. As a consequence, the simple gradient algorithm was used in this study for the Faraggi–Simon method.

2.3. NN architecture and parameters

For all three NN methods evaluated in this study, the “architecture” consisted of an input layer, one hidden layer and an output layer. The input layer consisted of two or four inputs (covariates) plus a special bias node with value equal to 1. The hidden layer consisted of two hidden nodes and a special hidden node that played a role similar to the constant term in linear regression. For the Faraggi–Simon method, this hidden node was not needed because its effect is incorporated into the baseline hazard. The output layer consisted of one output node for the Faraggi–Simon and the Buckley–James methods. The output layer consisted of multiple nodes for the Liestol–Andersen–Andersen method with the number of nodes equal to the number of intervals specified. For the simulation study, the general form of the Liestol–

Andersen–Andersen method was used, so that the criterion of proportional hazards was not assumed. In addition, we considered three intervals with the cut points determined by the times when the survival probabilities are approximately 35% and 65%.

For all NN methods, the logistic activation function was applied to the output of the hidden layer. For all but the Faraggi–Simon method, the logistic function was applied to the output of the output layer. For the Faraggi–Simon method, no activation function was used for the output layer. The error functions to be minimized were: the negative partial likelihood for the Faraggi–Simon method; the negative log-likelihood for binary data for the Liestol–Andersen–Andersen method; and the quadratic error function with one output node for the Buckley–James method. The initial learning rate was set to 0.05 and updated using the "line search" algorithm. The criterion for convergence was that the absolute change in the error function was less than 10^{-6}.

2.4. Cox regression

Standard Cox regression methods were used with inputs equal to covariates and two-way interactions of covariates when the true model included any interaction between inputs.

2.5. Simulation study

We conducted a Monte Carlo simulation study to evaluate the predictive accuracy of the three NN methods and Cox regression for handling censored data. We considered simulated data with two or four inputs (covariates), various censoring patterns, interaction between covariates, as well as proportional vs. non-proportional hazards. An exponential survival distribution was assumed for the proportional hazard models.

$$\text{Let } \lambda(t,X) = \exp\left\{ \sum_{i=1}^{p} \beta_i(t)x_i + \sum_{i\neq j} \gamma_{ij}(t)x_ix_j + \sum_{i\neq j\neq k} \gamma_{ijk}(t)x_ix_jx_k + \sum_{i\neq j\neq k\neq l} \gamma_{ijkl}(t)x_ix_jx_kx_l \right\}$$

be the hazard at any time t given p covariates $\boldsymbol{X}$, the survival times were then generated using inverse probability transformations (Newman and Odell, 1971). The following nine designs were considered:

Design 1: $p=2$ inputs with $\beta_1=1$, $\beta_2=0.25$ and no interaction (i.e., all $\gamma'\text{s}=0$). The distributions of the inputs are: x_1 has a Bernoulli distribution with probability 0.5, x_2 has a normal distribution with mean 0 and standard deviation 1, and x_1 and x_2 are independent. All subjects were followed to extinction (i.e., no censoring).

Design 2: Same as Design 1, except that a 20% censoring rate was applied, under the assumption that the censoring time was exponentially distributed and independent of the survival time.

Design 3: Same as Design 2, except that an interaction between x_1 and x_2 was assumed, i.e., $\gamma_{12} = 0.2$.

Design 4: $p = 4$ inputs with moderate two-way interactions and small three and four-way interactions (relative to the main effects): $\beta_1 = \beta_2 = 2$, $\beta_3 = 0.5$ and $\beta_4 = 1.0$, $\gamma_{12} = \gamma_{13} = 1.0$, all other γ's $= 0.5$. The distributions of the inputs are: x_1 and x_2 each have a Bernoulli distribution with probabilities 0.25 and 0.5, respectively, and x_3 and x_4 each have a normal distributions with mean 0 and standard deviation 1, and all variables are independent. A 30% censoring rate was applied.

Design 5: Same as Design 4, except the censoring rate was increased to 70%.

Design 6: $p = 4$ inputs with large three-way interactions (relative to the main effects): $\beta_1 = \beta_2 = 0.5$, $\beta_3 = \beta_4 = 0.25$, $\gamma_{123} = \gamma_{124} = \gamma_{234} = 3.0$, all other γ's $= 0.0$. The distributions of the inputs are: x_1 and x_2 each have a Bernoulli distribution with probabilities 0.25 and 0.5, respectively, and x_3 and x_4 each have a normal distributions with mean 0 and standard deviation 1, and all variables are independent. A 50% censoring rate was applied.

Design 7: Modification of Design 2, so that the hazard is not proportional: $\beta_1 = 1.0$ and $\beta_2 = 0.25$ for x_1 and x_2 before the time point with 70% survival probability. Thereafter, $\beta_1 = \beta_2 = 0$ for x_1 and x_2 (i.e., the survival probability was not associated with x_1 and x_2).

Design 8: Equivalent to Design 3 with $n = 200$.

Design 9: Equivalent to Design 5 with $n = 200$.

For Designs 1–7, a database of 200 realizations (cases) was generated. The database was randomly split into 100 training cases and 100 testing cases. For Designs 8 and 9, a database of 400 cases was generated and randomly split into 200 training cases and 200 testing cases. The performance of each of the four NN methods and Cox regression was determined by the C index for both the training and the testing sets (see below). The simulation process was then repeated 50 times.

2.6. The C index of discrimination

Motivated by rank tests based on Kendall's tau developed by Brown et al., Harrell et al. derived an index of discrimination, the C index, which can be considered as a generalization of the area under the ROC Curve for censored data (Harrell et al., 1982; 1984). Although other measures of discrimination exist, the C-index is the most often used with survival data, and is comparable to other methods (Harrell et al., 1984). The C index is calculated by taking all possible pairings of patients. For a given pair, the predictions are said to be *concordant* with the outcome if the patient having a higher predicted probability of survival lived longer. If the survival times for both patients are censored, or if only one died and the follow-up duration of the other was less than the survival time of the first, the pair is not counted. The C index is the proportion of predictions that are concordant out of all pairs of patients for which ordering or the survival times can be determined. Values of the C index near 0.5 indicate that the model is not predictive. Values of the C index near 1 indicate the input data virtually always determine which patient has better prognosis.

For the Liestol–Andersen–Andersen method with three intervals, the predicted survival probabilities were calculated as follows. If both patients in the pair fell in the same interval, then the predicted survival probabilities were calculated up to that interval and the concordancy status was determined as described above. In contrast, if the two patients in the pair fell in different intervals, then the predicted survival probabilities were calculated up to the first interval, and the predictions were regarded as concordant if the patient who had the event in the first interval also had the lower predicted survival probability.

For comparison purposes, the mean C index (± SD) was then tabulated. Repeated measures ANOVA were used to compare the average C-indices among all methods. Tukey's studentized range test was used for pairwise comparisons between methods. Significance was set at 0.05.

2.7. Software

Cox regression analyses utilized SAS (Cary, NC) and NN analyses utilized EPILOG PLUS® (Pasadena CA), an integrated PC-based statistical package for epidemiological and clinical trial applications.

3. Results

The results of the simulation study are presented in Tables 1–3 for Designs 1–9. For the training phase, the Farragi–Simon method performed the same as Cox regression for 7 of the 9 designs (Designs 1–3, 5, 7–9), and performed worse for two designs (Designs 4 and 6). The Liestol–Andersen–Andersen method outperformed Cox regression for three designs (Designs 2, 3 and 7), was equivalent to Cox regression for three designs (Designs 5, 8 and 9), and performed worse for three designs (Designs 1, 4 and 6). The Buckley–James method performed as well as Cox regression for five designs (Designs 1–3, 7, 8), but worse than Cox regression for the four other designs.

For the testing phase, a different pattern of performance for the NN methods was observed. In the situation where there was no censoring and no interaction (Design 1), the Faraggi–Simon and Buckley–James methods performed as well as Cox regression. In contrast, the Liestol–Andersen–Andersen method performed significantly worse than Cox regression ($p < 0.05$). A similar pattern among the NN methods was observed in the situation where there was moderate censoring (approximately 20%) with or without interaction between the two inputs (Designs 2 and 3).

For the situation (Design 4) with four inputs, small or moderate interactions, and moderate censoring (approximately 30%), Cox regression outperformed the Faraggi–Simon method, which in turn significantly outperformed the Liestol–Andersen–Andersen and Buckley–James methods ($p < 0.05$). For Design 5, which is similar to Design 4, except that the rate of censoring was increased from 30% to 70%, the Faraggi–Simon performed as well as Cox regression, and significantly outperformed

Table 1

Results of simulation study for Designs 1–3: C-index (mean±SD) ($n = 100$ cases, 50 replications)

	Design 1		Design 2		Design 3	
Method[1]	Training	Testing	Training	Testing	Training	Testing
COX	$0.650 \pm 0.028a$	$0.647 \pm 0.029a$	$0.645 \pm 0.030b$	$0.652 \pm 0.035a$	$0.672 \pm 0.029b$	$0.650 \pm 0.037a$
FS	$0.655 \pm 0.027a$	$0.642 \pm 0.035a$	$0.650 \pm 0.033a, b$	$0.648 \pm 0.034a$	$0.671 \pm 0.032b$	$0.648 \pm 0.041a$
Liestol	$0.634 \pm 0.031b$	$0.600 \pm 0.041b$	$0.655 \pm 0.038a$	$0.620 \pm 0.051b$	$0.679 \pm 0.030a$	$0.635 \pm 0.037b$
Buckley	$0.650 \pm 0.029a$	$0.648 \pm 0.039a$	$0.644 \pm 0.032b$	$0.646 \pm 0.043a$	$0.667 \pm 0.037b$	$0.647 \pm 0.042a$

[1] FS = Faraggi–Simon method; Liestol = Liestol–Andersen–Andersen method with 3 time intervals; Buckley = Buckley–James method. COX = Cox regression with main effects and two-way interactions for designs with any interactions (Design 3).

Design 1: $\gamma = \exp(x_1 + 0.25x_2)$, $x_1 \sim \text{Bernoulli}(0.5)$, $x_2 \sim N(0,1)$, no censoring.

Design 2: $\gamma = \exp(x_1 + 0.25x_2)$, $x_1 \sim \text{Bernoulli}(0.5)$, $x_2 \sim N(0,1)$, average censoring for training and testing sets=19% each.

Design 3: $\gamma = \exp(x_1 + 0.25x_2 + 0.2x_1x_2)$, $x_1 \sim \text{Bernoulli}(0.5)$, $x_2 \sim N(0,1)$, average censoring for training and testing sets = 20% each.

Values in a column that share the same letter (a, b) are not statistically different. Different letters indicate significant differences by Tukey's studentized range test.

Table 2
Results of simulation study for Designs 4–6: C-index (mean±SD) ($n = 100$ cases, 50 replications)

	Design 4		Design 5		Design 6	
Method[1]	Training	Testing	Training	Testing	Training	Testing
COX	$0.842 \pm 0.030a$	$0.824 \pm 0.028a$	$0.916 \pm 0.030a$	$0.870 \pm 0.041a$	$0.749 \pm 0.041a$	$0.697 \pm 0.050a$
FS	$0.825 \pm 0.032b$	$0.801 \pm 0.032b$	$0.904 \pm 0.032a$	$0.863 \pm 0.043a$	$0.683 \pm 0.050b$	$0.613 \pm 0.075b$
Liestol	$0.823 \pm 0.031b$	$0.775 \pm 0.044c$	$0.906 \pm 0.034a$	$0.827 \pm 0.051b$	$0.701 \pm 0.055b$	$0.601 \pm 0.050b$
Buckley	$0.803 \pm 0.033c$	$0.786 \pm 0.034c$	$0.861 \pm 0.066b$	$0.842 \pm 0.071b$	$0.551 \pm 0.100c$	$0.519 \pm 0.086c$

[1] FS = Faraggi–Simon method; Liestol = Liestol–Andersen–Andersen method with 3 time intervals; Buckley = Buckley–James method. COX = Cox regression with main effects and two-way interactions for designs with any interactions (Designs 4–6).

Design 4: $\gamma = \exp[2(x_1 + x_2) + 0.5x_3 + 1.0x_4 + 1.0(x_1x_2 + x_1x_3) + 0.5(x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4) + 0.5(x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 + x_1x_2x_3x_4)]$, $x_1 \sim$ Bernoulli(0.25), $x_2 \sim$ Bernoulli(0.5), $x_3 \sim \mathrm{N}(0,1)$, $x_4 \sim \mathrm{N}(0,1)$, average censoring for training and testing sets=32%.

Design 5: same as Design 4, average censoring for training and testing sets=70%.

Design 6: $\gamma = \exp[0.5(x_1 + x_2) + 0.25(x_3 + x_4) + 3.0(x_1x_2x_3 + x_1x_2x_4 + x_2x_3x_4)]$, $x_1 \sim$ Bernoulli(0.25), $x_2 \sim$ Bernoulli(0.5), $x_3 \sim \mathrm{N}(0,1)$, $x_4 \sim \mathrm{N}(0,1)$, average censoring for training and testing sets = 47%.

Values in a column that share the same letter (a, b, c) are not statistically different. Different letters indicate significant differences by Tukey's studentized range test.

Table 3
Results of simulation study for Designs 7–9: C-index (mean±SD) ($n = 100$ cases for design 7; $n = 200$ cases for Designs 8 and 9; 50 replications)

	Design 7		Design 8		Design 9	
Method[1]	Training	Testing	Training	Testing	Training	Testing
COX	$0.604 \pm 0.028b$	$0.592 \pm 0.038a$	$0.662 \pm 0.023a, b$	$0.661 \pm 0.025a$	$0.902 \pm 0.022a$	$0.886 \pm 0.026a$
FS	$0.609 \pm 0.029b$	$0.587 \pm 0.041a$	$0.662 \pm 0.026a, b$	$0.662 \pm 0.025a$	$0.889 \pm 0.025a$	$0.875 \pm 0.027a, b$
Liestol	$0.623 \pm 0.031a$	$0.583 \pm 0.038a$	$0.664 \pm 0.022a$	$0.651 \pm 0.029b$	$0.890 \pm 0.030a$	$0.857 \pm 0.042b, c$
Buckley	$0.597 \pm 0.043b$	$0.580 \pm 0.059a$	$0.661 \pm 0.024b$	$0.661 \pm 0.026a$	$0.858 \pm 0.084b$	$0.847 \pm 0.076c$

[1] FS = Faraggi–Simon method; Liestol = Liestol–Andersen–Andersen method with 3 time intervals;
Buckley = Buckley–James method. COX = Cox regression with main effects and two-way interactions for designs with any interactions (Designs 8 and 9).
Design 7: Non-proportional hazard modification of Design 2. $\beta_1 = 1.0$ and $\beta_2 = 0.25$ x_1 and x_2 before the time point with 70% survival probability. Thereafter, $\beta_1 = \beta_2 = 0$.
Design 8: Same as Design 3 except $n = 200$ cases.
Design 9: Same as Design 5 except $n = 200$ cases.
Values in a column that share the same letter (a, b, c) are not statistically different. Different letters indicate significant differences by Tukey's studentized range test.

the other two NN methods ($p < 0.05$). For the situation with large three-way interactions (Design 6), Cox regression outperformed both the Faraggi–Simon and Liestol–Anderson–Anderson methods, which in turn significantly outperformed the Buckley–James method. In the situation of a non-proportional hazard model (Design 7), all three NN methods performed as well as Cox regression.

We then evaluated the effect of increasing the number of cases from $n = 100$ to 200 (Designs 8 and 9). Increasing the sample size generally increased the performance of Cox regression and all the NN methods. Cox regression and two of the NN methods (Faraggi–Simon and Buckley–James) performed similarly for Design 8 (Table 3); Cox regression and one of the NN methods (Faraggi–Simon) performed similarly for Design 9 (Table 3).

4. Discussion

4.1. Performance of the methods

Several methods have arisen in recent years to permit survival analysis using neural networks. These methods were implemented by our group in the EPILOG PLUS® NERUAL utility, and they were evaluated in a Monte Carlo simulation. The simulations presented in this paper illustrate a representative sample of possible models and methods. Based on the mean C index, the overall findings for the testing phase of Designs 1–9 is as follows.

None of the NN methods outperformed Cox regression when Cox regression is used optimally (i.e., interactions were included in the model when they existed). Compared to Cox regression, the Faraggi–Simon method performed as well for 7 of the 9 designs. Performance was similar between Cox regression and the Faraggi–Simon method because the latter retained the proportional hazard nature of the Cox model while providing the ability to model complexities and interactions in the input data.

The Liestol–Andersen–Andersen method performed as well as Cox regression in the situation of non-proportional hazards (Design 7). These results were probably due to use of the general form of the Liestol–Andersen–Andersen method, which permitted modeling non-proportional hazards. Overall, the Liestol–Andersen–Andersen method showed poorer performance, which may have been due to loss of information when survival times are grouped into a small number of intervals. Furthermore, the general form of the Liestol–Andersen–Andersen method has more parameters to estimate than the other two NN methods. Overlearning was more prominent for the Liestol–Andersen–Andersen method than for the other two NN methods.

Finally, the Buckley–James method performed as well as Cox regression for 5 of the 9 designs, including the design with non-proportional hazards (Design 7). However, the Buckley–James method performed worse than Cox regression in the case of higher level models with interactions (Designs 4–6 and 9). These results were probably due to the method itself and/or the choice of error function.

4.2. The effect of modeling interactions in Cox regression analyses

In our study, all two-way interactions are included in the Cox regression when the design had any interactions (Design 3–6, 8 and 9). If interactions had not been included in the Cox regression analyses, then the relative performance of the NN methods would appear to be better. For example, the Faraggi–Simon method would perform significantly better than Cox regression for Design 6 (large three-way interaction), and would have been equivalent to Cox regression for all other designs which incorporated interactions (Designs 3–5, 8 and 9). The Liestol–Andersen–Andersen would have been equivalent to Cox regression for two designs which incorporated interactions (Design 6 and 9). (No improvement in the Buckley–James method was noted.) These results indicate that NNs designed for survival analysis can automatically accommodate interactions, whereas interactions in Cox regression analyses require the insight and experience of the data analyst.

4.3. The effect of varying the number of intervals for the Liestol–Andersen–Andersen method

One factor that could have direct impact on the performance of the Liestol–Andersen–Andersen method was that the survival times were grouped into only three intervals. For example, consider the situation that a given interval is rather large (e.g., 2–5 years), and that two subjects fail in the same interval, one at 2.5 years and another at 4.5 years. In the Liestol–Andersen–Andersen method, the observed outcome for both subjects at the target node is equal to 1, so that the information that the second subject has better survival is not used. In contrast, the Faraggi–Simon and the Buckley–James method do use this information in the error functions.

We explored the effect of increasing the number of intervals (and hence reducing the length of each interval) for the Liestol–Andersen–Andersen method. When we increased the number of intervals from 3 to 5 for Design 4, we realized an average increase in the C-index from 0.775 to 0.781 (a difference of 0.006) for the testing set. Increasing the number of intervals from 5 to 9 increased the C-index to 0.782 (a difference of 0.007).

4.4. Varying the NN architecture

We also explored the effect of altering the NN architecture by increasing the number of hidden nodes from 2 to 5 for Designs 4 and 6 (Table 4). For Design 4, increasing the number of hidden nodes improved the NN performance for the Farragi–Simon and Liestol–Andersen–Andersen methods for the training, but not the testing set. When the interaction effect was relatively large (Design 6), increasing the number of hidden nodes improved the performance for the Farragi–Simon and Liestol–Andersen–Andersen methods for both the training and testing datasets. Overfitting for the training sets was evident for both designs. A negligible increase in performance was found for the Buckley–James method suggesting that this method does not perform well when complex interactions exist.

Table 4
Results of simulation study with NN modifications for Designs 4 and 6: C-index (mean±SD) ($n = 100$ cases; 50 replications)

	No. hidden	Design 4		Design 6	
Method[1]	Nodes	Training	Testing	Training	Testing
FS	2	0.825 ± 0.032	0.801 ± 0.032	0.683 ± 0.050	0.613 ± 0.075
FS	5	0.839 ± 0.032	0.794 ± 0.032	0.761 ± 0.046	0.651 ± 0.049
Liestol	2	0.823 ± 0.031	0.775 ± 0.044	0.701 ± 0.055	0.601 ± 0.050
Liestol	5	0.872 ± 0.022	0.744 ± 0.042	0.849 ± 0.030	0.643 ± 0.055
Buckley	2	0.803 ± 0.033	0.786 ± 0.034	0.551 ± 0.100	0.519 ± 0.086
Buckley	5	0.805 ± 0.034	0.787 ± 0.035	0.558 ± 0.106	0.521 ± 0.078

[1] FS = Faraggi–Simon method; Liestol = Liestol–Andersen–Andersen method with 3 time intervals; Buckley = Buckley–James method.

Design 4: $\gamma = \exp[2(x_1 + x_2) + 0.5x_3 + 1.0x_4 + 1.0(x_1x_2 + x_1x_3) + 0.5(x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4) + 0.5(x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 + x_1x_2x_3x_4)]$, $x_1 \sim$ Bernoulli(0.25), $x_2 \sim$ Bernoulli(0.5), $x_3 \sim \mathrm{N}(0,1)$, $x_4 \sim \mathrm{N}(0,1)$, average censoring for training and testing sets=32%.

Design 6: $\gamma = \exp[0.5(x_1 + x_2) + 0.25(x_3 + x_4) + 3.0(x_1x_2x_3 + x_1x_2x_4 + x_2x_3x_4)]$, $x_1 \sim$ Bernoulli(0.25), $x_2 \sim$ Bernoulli(0.5), $x_3 \sim \mathrm{N}(0,1)$, $x_4 \sim \mathrm{N}(0,1)$, average censoring for training and testing sets=47%.

4.5. Calibration accuracy

In this study, we quantified the predictive discrimination of the NN methods. An alternative criterion of performance would have been to study the absolute prediction error. However, in order to develop calibration curves, predicted values from the output nodes were needed – data which are not provided by the EPILOG package. Therefore, additional research is needed to evaluate calibration measures of how well model prediction correspond to the actual data, within risk groups (Harrell et al., 1996).

4.6. Limitation of the study

A limitation of the present study was the use of simulated data. Although this approach allowed a systematic evaluation of model performance, the data followed patterns that may have been most conducive to Cox regression modeling. Variable interactions were only products, and variables followed either a normal distribution or a Bernoulli distribution. Therefore, a Cox regression with interactions would be expected to provide high performance. Some clinical datasets may have interactions and other properties that are not easily modeled by a Cox regression approach.

4.7. Advantages and disadvantages of NNs

NN software has generated a great deal of interest partly because it effectively places advanced modeling tools in the hands of users of personal computers. One of the advantages of NNs is that they can detect complex patterns among the inputs.

In contrast, one disadvantage of NN models is that they can model idiosyncratic features of the training dataset. When this happens the NN has *overlearned*, and will appear to perform extremely well on the training dataset but further testing on new data will generally be far less successful. In fact, overlearning was apparent in our testing datasets. For example, comparing training to testing results for Design 5, the Faraggi–Simon, Liestol–Andersen–Anderson and Buckley–James methods declined by 0.041, 0.079 and 0.019, respectively (Table 2). Overfitting can be reduced by using the m-item out validation approach, which is time intensive (Lachenbruch and Mickey, 1968).

Another disadvantage is that NNs are "computationally intensive" and may take a long time to train and converge to a solution. Using the quasi-Newton minimization algorithm, the Liestol–Andersen–Andersen and Buckley–James methods converged to a solution very fast. In contrast, the Faraggi–Simon method, which used the simple gradient minimization algorithm, took a long time to train. Additionally, the NN may converge not to the optimal solution, but rather to a local minimum, so that the resulting NN will perform sub-optimally. Some methods such as genetic algorithms can be used to avoid local minima. However, the genetic algorithm requires a lot of computer memory and is slow in convergence.

One advantage of Cox regression, is that the regression coefficients can be interpreted as the likelihood of an outcome given some value(s) of the risk factor (e.g., *odds ratios* or *relative risks*). NN weights usually do not lend themselves to such interpretation. The NN methods presented here represent only a few of the many options in survival modeling. It may be more valuable to explore issues of methods, applications, and potential for improvement than to draw conclusions about whether one method is inherently "superior" to another.

4.8. Summary

In summary, the results of our study shed light on the relative merits of the NN methods. In general, the results of our study also suggest that NNs can serve as effective methods for modeling right-censored data. However, the performance of the NN is somewhat variable, depending on the method that is used.

References

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Buckley, J., James, I., 1979. Linear regression with censored data. Biometrika 66, 429–436.

Faraggi, D., Simon, R., 1995. A neural network model for survival data. Statist. Med. 14, 73–82.

Harrell, F.E., Califf, R.M., Pryor, D.B., Lee, K.L., Rosati, R.A., 1982. Evaluating the yield of medical tests. J. Amer. Med. Assoc. 247, 2543–2546.

Harrell, F.E., Lee, K.L., Califf, R.M., Pryor, D.B., Rosati, R.A., 1984. Regression modeling strategies for improved prognostic prediction. Statist. Med. 3, 143–152.

Harrell, F.E., Lee, K.L., Mark, D.B., 1996. Tutorial in biostatistics, multivariate prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Statist. Med. 15, 361–387.

Hornik, K., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366.

Lachenbruch, P.A., Mickey, M.R., 1968. Estimation of error rates in discriminant analysis. Technometrics 10, 1–11.

Liestol, K., Andersen, P.K., Andersen, U., 1994. Survival analysis and neural nets. Statist. Med. 13, 1189–1200.

Newman, T.G., Odell, P.L., 1971. The Generation of Random Variates. Charles Griffin and Co., London, p. 28.

Warner, B., Misra, M., 1996. Understanding neural networks as statistical tools. Amer. Statist. 50, 184–293.