Artificial Neural Networks Techniques for Competing Risks Modelling in Double-Blinded Randomized Clinical Trial

K. A. Dauda[1]; **W. B. Yahya**[2]

[1]Department of Statistics and Mathematical Sciences,
Kwara State University,
Malete, Nigeria.
e-mail: kazeem.dauda@kwasu.edu.ng; qdauda70@gmail.com

[2]Department of Statistics,
University of Ilorin,
Ilorin, Nigeria.
e-mail: wbyahya@unilorin.edu.ng

***Abstract* — In many situations, the functional relationship between covariates and response variable is of great interest. However, due to the ability of Artificial Neural Network (ANN) to handle both linear and nonlinear functional relationship and its ability to perform intelligent tasks similar to those perform by human brain, it is recently receiving greater attention in biostatistics. In this study, the resilient backpropagation ANN for competing risk modelling using Cox-Snell residual as a common response (ANN-CS) is proposed. Data set from a double-blinded randomized clinical on prostate cancer was used to demonstrate the performance of the proposed model. It was observed from the results that, as the hidden layers increases, the sum of squared errors of the model also reduces in both risks. In addition, when the Garson's Algorithm was applied to the data set, it was found that the Electrocardiogram Code predicts the survival of the patients with the risk of dying from prostate cancer better than other covariates considered. Moreover, age of the patients also predicts the survival of patients with prostate cancer when the risk is due to other diseases. Finally, Tumour Stage, Cardiovascular Disease History and Performance Rating were found to influence the survival of the patients with prostate cancer irrespective of the risk.**

Keywords-*Artificial Neural Network (ANN), Garson's Algorithm, Competing Risks, Cox-Snell residual, Resilient backpropagation; Hidden layers.*

I. INTRODUCTION

In many clinical trial studies, survival analysis with more than one (causes of) failure often occur and this is being regarded as competing risks, i.e. subjects may fail (die) from any of the k^{th} different events with respect to a particular time point [1], $k > 1$.

Medical researchers use survival analysis to evaluate the performance of prognostic factors in outcome such as cancer recurrence, death, and provide patients with their treatment options [2-5].

There are three common classical ways of analyzing and summarizing competing risks data, these include: the complement of the Cox proportional hazard function, cumulative incidence function and cause-specific hazard function. All these classical methods rely on the assumptions such as: proportionality of the hazard function, distribution of the survival time and independent of the competing risks. However, if some of these assumptions failed the classical methods would fail.

Additionally, the above stated classical methods assume that patient's clinical response is a linear combination of some covariates or prognostic factors, whereas, such relationship is not always linear in many situations.

In other to circumvent the problems and assumptions that are associated with the use of classical methods, the present work seeks to propose a new method for modelling the survival times of prostate cancer patients with competing risks using the techniques of resilient backpropagation Artificial Neural Networks (ANN) with Cox-Snell residuals as model's response (ANN-CS).

II. MATERIALS AND METHODS

A. *ANN in Brief*

A mathematical ideal that tries to simulate from the operational and functionalities of biological neural network is known as Artificial Neural Network (ANN) [6]. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach for computation. It is a powerful tool for modelling linear and non-linear functional relationship between the covariates and response variable, especially when the underlying assumptions that govern the classical model failed [7].

Basic building block of every artificial neural network is artificial neuron, that is, a simple mathematical model (function). Such a model has three simple sets of rules: multiplication, summation and activation [8,9]. At the entrance of artificial neuron the inputs are weighted such that every input value is multiplied with individual weight. In the middle section of artificial neuron is sum function that sums all weighted inputs and bias [6].

Simulation through Electrical input can occur in neural cell and may not fire its action potentially. When the neural network is formed, it can encode information to mimic the human neuron (see Fig. 1), this process is called neural coding. The information can be encoded and can also be decoded. The decoding process is sometime refers to as recalling process and the encoding process is called training or learning. This makes the artificial neural network efficient and reliable.

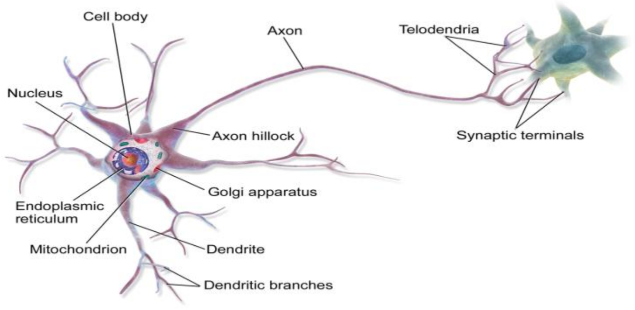

Fig. 1: Structure of Biological Neuron
Source: (https://en.wikipedia.org/wiki/File:Blausen_0657_MultipolarNeuron.png)

The original function of neural network (NN) was to use computer-based model to mimic human brain. The most widely used NN is called the Multi-layer perceptron (MLP) system, which consist of a set of input features, a number of interactive layers, in between the inputs and output layers and specifically one output layers. Therefore, this study adopts the use of MLP system via the Cox-Snell residual to model survival data with competing risks endpoints. Detail of the procedures and the methodologies are given in the subsequent sub-sections.

B. The set-up of the proposed ANN-CS method

Suppose there are observations (T_i, δ_i, x_i) in a competing risk setting for individual subject i, $i = 1, 2, \dots, n$, where T_i is the individual failure time, δ_i is the censoring index (type of event) and $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})'$ is a vector of covariates. A normal way of summarizing event of interest adjusting for other events is by the use of *cumulative incidence function* defined in the following equation;

$$F_1(t) = P(T \leq t, \delta_i = 1) \qquad (1)$$

So, equation (1) represents the expected proportion of patients suffering from event 1 over the course of time *t*. For investigating the effect of covariates on the quantity in equation (1), Fine and Gray [10] developed and built a model called sub-distribution hazard given by the following equation;

$$\lambda_1(t) = \frac{dF_1(t)}{dt}\left(\frac{1}{1-F_1(t)}\right) \qquad (2)$$

They specifically proposed to fit a Cox proportional hazards model given by;

$$\lambda_1(t|x_i) = \lambda_{1,0}(t)\exp\,(x_i'\beta) \qquad (3)$$

where $\lambda_{1,0}(t)$ is an unspecified baseline hazard, and $\beta = (\beta_1, \beta_2, \dots, \beta_p)'$ is a vector of parameters to be estimated by maximizing the partial likelihood with modified risk set and inverse probability of censoring weights. Thus, the partial likelihood and the score function of model (3) are given as:

$$L_P(\beta) = \prod_{t_i=1}^{n}\left[\frac{e^{\beta x_i'}}{\sum_{j\epsilon R_i} e^{\beta x_j'}}\right]^{\delta_i} \qquad (4)$$

where R_i is the risk set at time t_i. The log-partial likelihood is:

$$l_P(\beta) = \sum_{t_i=1}^{n} \delta_i\left[\beta x_i' - \log\left\{\sum_{j\epsilon R_i} e^{\beta x_j'}\right\}\right] \qquad (5)$$

Then, the partial likelihood score function is:

$$U(\beta) = \frac{d\,l_P(\beta)}{d\beta} = \sum_{t_i=1}^{n} \delta_i\left[x_i' - \log\left\{\frac{\sum_{j\epsilon R_i} x_j' e^{\beta x_j'}}{\sum_{j\epsilon R_i} e^{\beta x_j'}}\right\}\right] (6)$$

Thus, the maximum partial likelihood estimator can be found by solving $U(\beta) = 0$.

In this study, we are interested in the Cox-Snell residual of model (3) but without the influence of the covariates with respect to each event, and, thus, the standard method of Cox-Snell residual is adopted and describe here.

C. *The Cox-Snell Residuals*

Cox-Snell residuals [11] are very useful brand of residuals based on the cumulative hazard function. Additionally, Cox-Snell residuals are used to test the overall fit of the Cox proportional hazard model. In the Cox-Snell residual, the i^{th} individual, $i = 1,2, \dots., n$ for null model is given by

$$r_{c_i} = \widehat{H}(t_i) \qquad (8)$$

where $\widehat{H}(t_i)$ is an estimate of cumulative hazard function at time (t_i), and t_i is the observed survival time. The quantity in equation (8) can be reduced to equation (9) with respect to j^{th} events as follows;

$$r_j(t) = -\log\left[\hat{S}_j(t)\right] \qquad (9)$$

where $S(t) = P_r(T \geq t, \delta_i = j) = e^{-\Lambda_j(t)}$, thus $\hat{S}_j(t) = e^{-\hat{\Lambda}_j(t)}$ and the estimation will be done by using Nelson-Aalen estimator [12] in equation (10) below:

$$\widehat{\Lambda}_j(t) = \sum_{i:t_i \leq t} \frac{\delta_{ij}}{n_i} \qquad (10)$$

where n_i is the number of individuals alive and uncensored just prior to time t_i. Therefore, the actual Cox-Snell residual for this study is the quantity $r_j(t)$ in equation (9) and this will be used as a response variable in the ANN model.

The simplest multi-layers perceptron is given by;

$$g(x) = f(w_0 + \textstyle\sum_{i=1}^{n} w_i x_i) \qquad (11)$$

where w_0 denotes the intercept, $w_i = (w_1, w_2, \dots, w_n)$ the vectors consisting of all synaptic weights without the intercept, and $x = (x_1, x_2, \dots, x_n)$ the vector of all covariates [7]. We proceed to formulate a new model by letting $g_j(x) = r_j(t)$ where $j = 1\,\&\,2$ for the two competing risks.

$$g_j(x) = f_j(w_0 + \textstyle\sum_{i=1}^{n} w_i x_i) \qquad (12)$$

Model (12) describes how information comes into the body of an artificial neuron via inputs that are weighted (each input can be individually multiplied with a weight). The body of an artificial neuron then sums the weighted inputs, bias and "processes" the sum with a transfer function. At the end, an artificial neuron passes the processed information via output(s) as shown in Fig. 2.

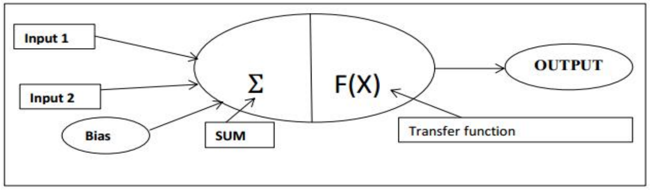

Fig. 2: Artificial Neuron Design

D. *Learning Perceptron Algorithm*

The perceptron learning rule was originally developed by Frank Rosenblatt in the late 1950s [13], which involves presenting the training set to the network input and the output is computed. This study proposed to use online training, which uses stochastic gradient descent optimization technique. The algorithm is:

i. Initialize the weights and threshold to small random numbers.

ii. Present a vector **x** to the neuron inputs and calculate the output.

iii. Update the weights according to:

$$w_j(t+1) = w_j(t) + \eta(y(x) - g(x))x_i \qquad (6)$$

where:
η = Learning rate that takes on value between 0 to 1.
$y(x)$ = actual response values and in our study is $r_j(t)$, $j = 1,2$.
$g(x)$ = Output from learning rule.

iv. Repeat steps ii. and iii. until either:

   a. the iteration error is less than a user-specified error threshold; or

   b. a predetermined number of iterations have been completed.

During this training, it is often useful to measure the performance of the network as it attempts to find the optimal weight set in the algorithm.

A common error measure or cost function used is sum of squared error. It is computed over all of the input vector/output vector pairs in the training set and is given by equation (13).

Let the training set be defined by

$$D = \{(y^{(1)}, g^{(1)}), (y^{(2)}, g^{(2)}), \dots, (y^{(m)}, g^{(m)})\}$$

where $y(x) = y$ and $g(x) = g$ for simplicity. Then, the cost function with respect to weight *w* is

$$E(w) = E = \frac{1}{2}\textstyle\sum_{d\epsilon D}\left(y^{(d)} - g^{(d)}\right)^2 \qquad (13)$$

where ½ is a scaling constant which makes it easier to take the derivative of the error measure, as commonly used in optimization problems [14]. To implement the algorithm, we need to calculate the gradient of E as:

$$\nabla E(w) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_m}\right]$$

and to update the weight by the following equation;

$$\nabla w = -\eta \nabla E(w)$$

Now, to be able to differentiate the quantity in equation (13) with respect to w_i, the network required a continuous function such as sigmoid function of the form:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Thus,

$$f(x) = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^{n} w_i x_i)}}$$

where the new$f(x)$ is melted from equation (12),

E. Derivative of the weight via stochastic gradient descent

We start by recalling equation (12) and re-write it with respect to online training as:

$$g_j^d(x) = w_0 + \textstyle\sum_{i=1}^{n} w_i x_j^d \qquad (14)$$

Let's learn w_i's that minimizes the squared error in equation (13).

Suppose: $y = f(u)\ and\ u = h(x), then\ \frac{\partial y}{\partial x} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial x}$. We make use of this as follows:

$$\frac{\partial E^{(d)}}{\partial w_i} = \frac{\partial E}{\partial g^{(d)}} \frac{\partial g^{(d)}}{\partial net^{(d)}} \frac{\partial net^{(d)}}{\partial w_i} \qquad (15)$$

where $net^{(d)} = w_0 + \sum_{i=1}^{n} w_i x_j^{(d)}$, $g^{(d)} = g_j(x)$ and

$$g^{(d)} = \frac{1}{1+e^{-net^{(d)}}} \qquad (16)$$

By substituting equation (14) into equation (15) we have

$$\frac{\partial E^{(d)}}{\partial w_i} = \frac{\partial}{\partial g^{(d)}} \frac{1}{2} \sum_{d \epsilon D} \left(y^{(d)} - g^{(d)}\right)^2 \frac{\partial g^{(d)}}{\partial net^{(d)}} \frac{\partial net^{(d)}}{\partial w_i}$$

$$\rightarrow \frac{\partial E^{(d)}}{\partial w_i} = -\left(y^{(d)} - g^{(d)}\right) \frac{\partial g^{(d)}}{\partial net^{(d)}} \frac{\partial net^{(d)}}{\partial w_i}$$

The sigmoid function in equation (16) has the useful property that its derivative is easily expressed in terms of its output as $\frac{\partial g^{(d)}}{\partial net^{(d)}} = g^{(d)}(1 - g^{(d)})$ [15]. Hence,

$$\frac{\partial E^{(d)}}{\partial w_i} = -\left(y^{(d)} - g^{(d)}\right) g^{(d)} \left(1 - g^{(d)}\right) \frac{\partial net^{(d)}}{\partial w_i}$$

$$\rightarrow \frac{\partial E^{(d)}}{\partial w_i} = -\left(y^{(d)} - g^{(d)}\right) g^{(d)} \left(1 - g^{(d)}\right) x_j^{(d)} \qquad (17)$$

Now, we have covered how to do gradient descent for single-layer networks with sigmoid output units in equation (17).

The next step is to adopt multilayer ANN and backpropagation errors from the output units to the hidden units. In backpropagation, each weight is changed by

$$= -\eta \frac{\partial E}{\partial w_{ji}}$$

$$= -\eta \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}$$

$$= \eta \delta_j g_j \qquad (18)$$

where $\delta_j = -\frac{\partial E}{\partial net_j}$

These gradients are now inserted into the algorithm and thereby govern the correction of the weights for each iteration step in the training procedure.

F. Relative Importance by Garson Algorithm

After experimenting with the proposed algorithm on the real life (or simulated) dataset, the contribution of each independent (input) variable is measured using the Garson Algorithm [16]. The algorithm partitions the hidden layer weights into components associated with each input node. Subsequently, the percentage of all hidden node weight associated with the input node was used to measure the relative importance of that attribute. In general, each input node j, where $j = 1,2, \dots$, the relative importance (RI_j) can be calculated using the equation (19) below.

$$RI_j = \frac{\sum_{m=1}^{N_h}\left[\left[\frac{\left|w_{jm}^{ih}\right|}{\sum_{k=1}^{N_i}\left|w_{km}^{ih}\right|}\right] \times \left|w_{jm}^{ho}\right|\right]}{\sum_{n=1}^{N_i}\left[\sum_{m=1}^{N_h}\left[\left[\frac{\left|w_{nm}^{ih}\right|}{\sum_{k=1}^{N_i}\left|w_{kw}^{ih}\right|}\right] \times \left|w_{jm}^{ho}\right|\right]\right]} \qquad (19)$$

In equation (19), N_i and N_h are the number of input and hidden neuron respectively, w is the connection weight, the superscripts $"i", "h"\ and\ "o"$ refer to input, hidden and output layers, respectively and subscripts $"k", "m"\ and\ "n"$ refer to input, hidden and input neurons used. In our case here, there is only one output neuron. In our algorithm, for each input node j, the relative contribution of j to the outgoing signal of each hidden neuron is calculated and presented in percentage, which then serves as measure of importance for each node of the given variable. Additionally, the entire inputs variable with smallest contribution to the final output of the network is eliminated.

G. *Comparing the Proposed Model with Classical Competing Risk Model*

The comparison between the proposed and classical competing risk model was assessed using the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Relative Squared Error (RSE) criteria.

All the analysis implemented in this research work was done through the use of package "survival", "KMsurv", "NeuralNetTools", "RSNNS", "nnet", "devtools", "neuralnet" and "clusterGeneration" all in the language R.

III. Analysis

In this study, dataset on double-blinded randomized clinical trial on 236 patients with prostate cancer was considered. This dataset was collected from Royal Berkshire Hospitals, Reading, United Kingdom, and is freely available on the institution website. There was only one treatment in the study and this is 1.0mg diethylstilbestrol (DES). Of the 236 patients, 119 of them were given placebo, the remaining 117 patients were given DES treatments. Apart from the treatment, information on other prognostic factors was also reported in the dataset.

The main aim of the clinical trial was to determine how survival time of patients was affected by the treatment and any other prognostic factor(s). Also, possible toxic effect of DES particularly in terms of mortality from heart disease was to be examined. However, endpoint considered was death from prostatic cancer and death from any other causes. The following covariates were used for this research while others were discard due to missing observations: Age in years, Weight index (WTINDEX), Performance rating (PERF), Cardiovascular Disease History (HISTCD), Electrocardiogram Code (ECGCODE), Tumour Size (SIZEPT), Tumour Stage & Grade Index (SHINDEX) and Serum Haemoglobin gm/100ml (SERUMHG). The Follow-up Time is in Months and the Survival Status was recorded by treating death from prostatic cancer as the event of interest (cause 1), death from Heart or Vascular Disease and others causes as competing risks (cause 2) and discharge alive as true **censoring (event 0).**

The results in Table 1 were the estimated performance measures of the proposed and classical models based on the double-blinded randomized clinical trial data. The two competing risks are death due to prostate cancer and death due to other diseases like Heart or Vascular Disease, Cerebrovascular accident, Thrombosis and Respiratory Disorder. The results of proposed and existing models under the death due to prostate cancer (event 1) are presented in Table 1.

Table1: Comparison between the proposed model and classical model on Double-blinded randomized clinical trial data.

Death due to Prostatic Cancer (Event 1)			
Model	**RMSE**	**MAE**	**RSE**
Proposed model (CMLP)	**0.1237**	**0.1397**	**0.9933**
Classical model	0.3020	0.1709	4.6391
Death due to Other Diseases(Event 2)			
Proposed model (CMLP)	**0.2021**	**0.2519**	**1.0000**
Classical model	0.2261	0.2763	1.2036

It was observed from the various performance measures in Table 1 that irrespective of the criteria, the proposed ANN-CS model performed better than the existing technique for modelling survival data with competing risks under the two causes of deaths (events 1 & 2). In all cases, the proposed model predicted the probability of survival of patients under the two risks better than the existing model.

Table 2: Results of the back propagation SNN at varying hidden layers

Hidden layers	**Sum of Squared Error**	**Reached Threshold**	**Iteration step**
2	2.3196	0.00001	22
5	2.3128	0.0070	24
10	1.8250	0.0093	4277
15	1.2254	0.0097	40391

Table 2 shows the results of proposed ANN-CS model with four different hidden layers 2, 5, 10 and 15. It can be deduced from the results that as the hidden layers increases the sum of square error reduces while the reached threshold increases and the number of iterations or steps taken to attain the threshold equally increased.

For clarity, we present the virtualization of the results of training process of proposed model with 2 hidden layers as shown in Fig. 3. The plot consists of trained synaptic weights, different intercept and basic information about the training process like the overall error and number of iteration steps as explained earlier in the methodology.

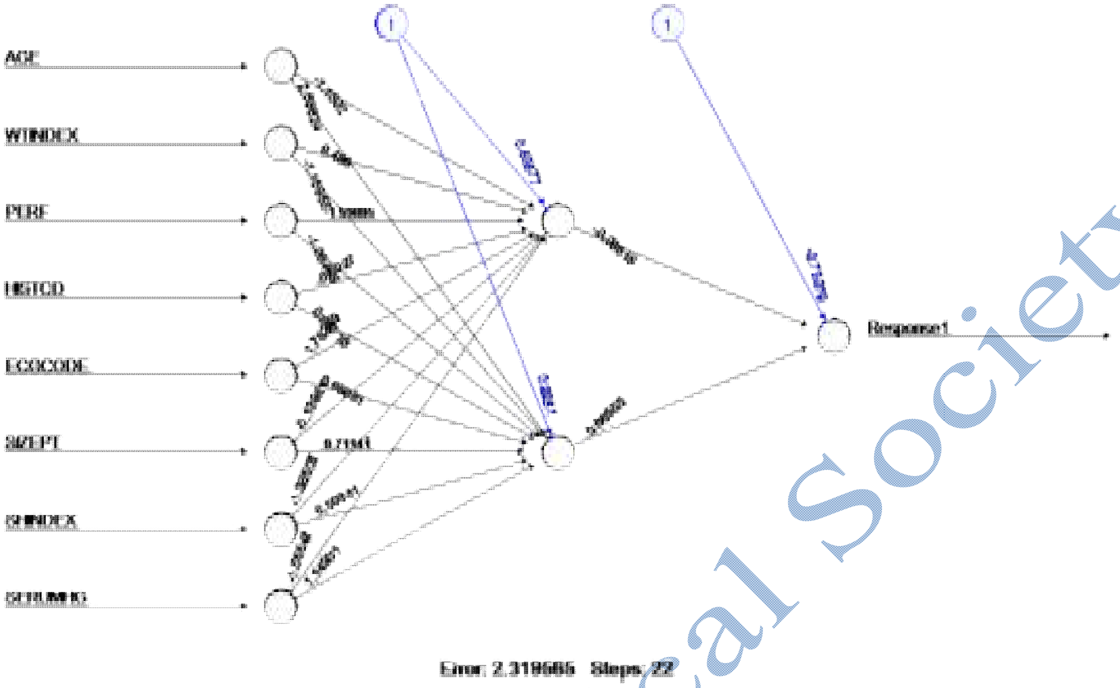

Fig. 3: Plot of trained neural network including trained synaptic weights and basic information about the training process.

Table 3: Variable Importance

Covariate	Relative Important	Remarks on the survival of the patient
Age in years (AGE)	0.2268	Positive relationship
Weight Index (WTINDEX)	0.0001	No importance
Performance Rating (PERF)	-0.4577	Fair negative relationship
Cardiovascular Disease History (HISTCD)	-0.4635	Fair negative relationship
Electrocardiogram Code (ECGCODE)	-1.0000	Perfect negative relationship
Tumour Size(SIZEPT)	-0.0397	Weak negative relationship
Tumour Stage, Grade Index (SHINDEX)	-0.7489	Strong negative relationship
Serum Haemoglobin gm/100ml (SERUMHG)	0.13292	Weak positive relationship

The variable important values in Table 3 were determined using equation (18) and were further presented in Fig. 4. The results showed that the covariates Age, Tumour Stage, Grade Index (SHINDEX) and Electrocardiogram Code (ECGCODE) have the strongest positive and negative relationship individually with the response variable (probability of survival). On the other hand, the following covariates - Weight Index (WTINDEX) and Tumor size - have no effects on the survival of the patients with prostate cancer. Nevertheless, these covariates with zero or close to zero relative important most likely have some marginal effect on the response variable, but its effect is irrelevant in the context of the other covariates. Other covariates were found to have either very weak negative or week positive relationship with the response variable.

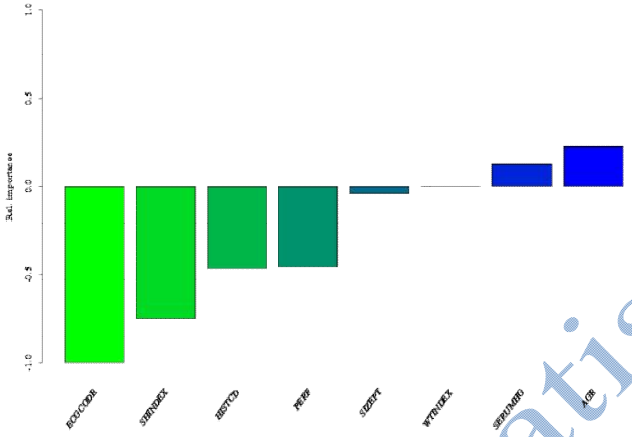

Fig 4: Relative important of the eight covariates on response variable (probability of survival)

IV. DISUSSION

This work extended the techniques of ANN to competing risk modelling in survival analysis. The new proposed method was found to yield very impressive and encouraging results. Comparison of the proposed method with the classical method for modelling survival data with competing risk revealed the superiority of the new method over the classical one based on real life dataset on prostate cancer patients. A number of prognostic variables were found to be highly related to the survival time of the patients. The relative importance of these variables was determined using Garson Algorithm for both competing risks events.

V. CONCLUSION

This study has demonstrated the use of artificial neural network techniques in competing risks analysis. Then new proposed ANN-CS model has yielded relatively higher performance compared to the classical method due to its flexibility by being robust to some of the assumptions and inherent problems of the classical method.

Also, the new method has the capability to accurately identify the important prognostic factors that are predictive of the patients' survival times. It was further demonstrated that the new method is quite effective at handling both linear and nonlinear association between the covariate and probability of survival.

Finally, this work has shown that proper selection of training data enables the ANN to cover a wide range of flexibility. On the other hand, it appears that increasing the number of layers provides a slight improvement on the performance of the trained ANN (as equally reported elsewhere[19]) and reduction in mean square error.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their observations and suggestions on the original manuscript which have immensely helped to improve the paper.

REFERENCES

[1] Klein JP, Moeschberger, ML.(2003). *Survival analysis: Techniques for censored and truncated data*. New York: Springer.

[2] Yeh R. W, Secemsky E. A and Kereiakes D. J. (2016). Development and validation of a prediction rule for benefit and harm of dual antiplatelet therapy beyond 1 year after percutaneous coronary intervention. JAMA, 315(16):1735–1749.

[3] Patrick R. and Douglas G. A. (2013). External validation of a cox prognostic model: principles and methods. BMC medical research methodology, 13(1):1.

[4] Bair E. and Tibshirani R. (2004). Semi-supervised methods to predict patient survival from gene expression data. PLoS Biol, 2(4):e108.

[5] Cheng W-Y, Yang T-H. O., and Anastassiou A. (2013). Development of a prognostic model for breast cancer survival in an open challenge environment. Science translational medicine, 5(181):181ra50–181ra50.

[6] Krenker A, Bešter J. and Kos A. (2011). Introduction to the Artificial Neural Networks, Artificial Neural Networks - Methodological Advances and Biomedical Applications, Prof. Kenji Suzuki (Ed.), ISBN:978-953-307-243-2.

[7] Günther F. and Fritsch S. (2010). Neuralnet: Training of Neural Networks. R Journal Vol. 2/1, June.

[8] Sima J. (1998). *Introduction to Neural Networks,* Technical Report No. V 755, Institute of Computer Science, Academy of Sciences of the Czech Republic.

[9] Gurney K. (1997). *An Introduction to Neural Networks*.(1st ed.) UCL Press, London EC4A 3DE, UK.

[10] Fine J. P. and Gray R. J. (1999) A proportional hazards model for the sub-distribution of a competing risk. *J Am Stat Assoc*. 94:496–509.

[11] Cox, D. R. and Snell, E. J. (1968). A general definition of residuals. *Journal of the Royal Statistical Society*, series B, 30(2):248–275.

[12] Aalen, O. O. (1978). Nonparametric inference for a family of counting processes, Annals of Statistics, 6, 701-726.

[13] Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory. *Psychological Review*, v65, No. 6, pp. 386–408. doi:10.1037/h0042519

[14] Chow M.Y., Goode P., Menozzi A., Teeter J., and Thrower J. (1994). Bernoulli error measure approach to train feed forward artificial neural networks for classification problems. *Proc. IEEE Int. Conf. Neural Networks*, pp. 44–49.

[15] Mitchell T. (1997). *Machine Learning.* McGraw-Hill.

[16] Garson GD.(1991). Interpreting Neural-Network Connection Weights. *AI Expert*; 6:46-51.

.[17] Andersen, P.K., Borgan O, Gill R.D., Keiding N. (1993), *Statistical Models Based on Counting Processes*, Springer-Verlag.

[18] Therneau T. and Grambsch P. (2000), *Modeling Survival Data: Extending the Cox Model*, Springer-Verlag, New York. ISBN: 0-387-98784-3.

[19] Yahya WB, Oladiipo MO and Jolayemi ET (2012): A fast algorithm to construct neural networks classification models with high-dimensional genomic data. *Annals. Computer Science Series,* **10** (1): 39-58.