International Journal of Intelligent Computing and Information Sciences

PREDICTION OF LUNG CANCER USING ARTIFICIAL NEURAL NETWORK

N. M. Numan S. M. Abuelenin M. Z..Rashad
Faculty of Computer and Information,Mansoura University, Egypt
nawras.m.numan@gmail.com dr.sherihan@yahoo.com magdi_z2011@yahoo.com

Abstract:Feed-forward neural networks trained via supervised learning have proven to be successful in the field of pattern recognition for the purpose of survival rate prediction in terminal illnesses. Thetwo problems are addressed in this research through the use of a novel model for the prediction that involves large number of multiple simple two-layered neural networks connected in a complete parallel structure called a macro-neural model. Macro-neural networks are a type of feed-forward neural networks in which the network design is composed of many interconnected simple networks with no hidden layers. The type of the devised architecture can affect both generalization and convergence as well as provide a solution to several known problems in conventional designs such as over-fit problem & the unknown of the hidden layer.The proposed model is examined with the conventional methods. Detail error analysis is presented and several data sets are analyzed and compared. In most of the comparisons carried out the proposed model proved to be superior to conventional survival rate prediction methods. However, the generalization of the obtained results cannot be claimed for any problem as the effectiveness of the proposed model has to be examined for more diverse problems before making the conclusion that the proposed model is indeed superior for any problem.

Keywords: Lung Cancer,Artificial Neural Network, Body Mass Index, PercentageError.

1. Introduction

”Artificial Intelligence” aims to simulate human intelligence and to develop important concepts from that simulation. Artificial intelligence has developed as a set of learning algorithms used to find patterns within some collectable data that may range from measurable time series to descriptive statements. Among these learning algorithms are neural networks. Neural networks have found a vast amount of applications in artificial intelligence. From expert systems to decision making including many medical applications, neural networks proved to be quite useful. For that reason neural networks were studied extensively in the past where several limitations for their performance were identified. A study of such limitations and possible ways to overcome them cannot be generalized to any application as each application has its own specific features. Therefore a specialized study of the neural network limitations can only be made with respect to a certain application.Since their early introduction neural networks were applied in many medical studies.Several decision support systems have been developed in various medical fields including medical diagnosis and medical survival rate estimation where previous experience or expert knowledge is often used to design decision support systems. Such a task becomes interesting when no prior knowledge is available. Yet many of the proposed designs lack the required amount of reliability due to the small size of input data provided.

Many data have been gathered with the advancement of technology, and computers are used to handle such large amount of data [1]. The technology of detecting relation and knowledge from data forms a branch of applied artificial intelligence (AI), and is the process of choosing, finding and modeling huge amounts of data so as to discover unknown patterns or relationships which provide a clear and practical result. There are two phases in data classification technique: classification model construction and model classification efficiency evaluation. In classification model construction, the algorithm is chosen through a dataset to build the predictive classification model. In classification efficiency evaluation, a testing dataset is applied. Every data in a control dataset or a testing dataset includes distinct number of attributes and a target class.

The artificial neural networks (ANNs) algorithm has been applied in many medical issues, for example, assessing and building predictive models for diabetes, predicting the survival rates in terminal illnesses.As the pathogenesis of a terminal disease is variable, it is difficult to accurately diagnose beforehand not to mention evaluating the survival rate once diagnosed. Despite all efforts at management, prognosis of advanced lung cancer is extremely weak, with a median survival time of ~1 year [1]. Lung cancer is the tumor with the highest death in males in today's society. Its incidence rate is increasing in women [2]. There are several types of small cell lung cancer (SCLC) or oat cell cancer, including a mix of small cell & other cell types. These cancers grow rapidly—doubling in cell number about every 30 days—and spread quickly.As the pathogenesis of a terminal disease is variable, it is difficult to accurately diagnose beforehand not to mention evaluating the survival rate once diagnosed. Of course the type of the disease and its stage at diagnosis highly affects the survival chances. A terminal illness is a disease that will result in the death of the patient regardless of any treatment intervention with a probability higher than 50%. The mortality rates of terminal diseases are above 70% of all mortalities in African and Asian countries, but the incidences are more in western countries though the mortality is less. Accordingly any fatal illness that goes untreated is considered a terminal illness in its final stages. Statistical techniques and artificial intelligence techniques are important in unearthing previously unknown knowledge.

A huge number of prognostic factors, including 150 [3,4] are mentioned in case survey and can be found in the records of the national cancer institute officially called(Surveillance, Epidemiology &End Results) SEER Program. The nature of which consist of:

(1) Information yielded by the tumor: anatomical tumor extension, histological type, and genetic-molecular pathology.
(2) Information yielded by the patient/host: clinical factors, basic biological factors.
(3)Information from the treatment: standard complete surgery and incomplete surgery, among others. However, the weight or the ascertainable predictive magnitude on the survival of each of the factors is very heterogeneous and widely controversial.

There are significant discrepancies according to the articles reviewed on the predictive capacity of basic variablesfor instance, tumor size [3].Unfortunately these wide sense differences rendered statistical analyses highly useless and this is the reason why neural network and statistical survival rate predictions were marginally successful despite all the research carried out in this field [5].

Data extracted from the medical records of the SEER program includes age, gender, smoking history, surgical procedure, surgical margin status, histological diagnosis, T stage, N stage, TNM stage, number of lymph node stations examined, number of lymph nodes removed, and treatments (forexample, surgical resection, adjuvant chemotherapy, radiotherapy or immunotherapy). Initially patients were restaged according to the 7th edition of the American Joint Committee on Cancer (AJCC) TNM classification of lung cancer [3].

Demographics and clinical characteristic of the subjects are summarized as mean ± standard deviations for continuous variables and percentage for categorical variables. Univariate and multivariate Cox proportional hazards regression analyses were carried out to identify factors having a significant impact on overall survival. Significant variables ($p < 0.05$) in univariate Cox proportional hazards regression analysis were entered into multivariate analysis. The associated results are reported as hazard ratios with 95% confidence intervals. Kaplan-Meier curves of overall survival OS are presented for the cumulative survival rates versus follow-up time for patients in whom the 4 lung cancer resection guidelines were or were not followed [3].

2. Related Work

A Biglarian, et al. [6] used artificial neural networks (ANNs) in the past to predict the survival rate among patients of several cancer types. One of these is gastric cancer where several methods have been proposed and tried using the Cox proportional hazard and artificial neural network models as well as comparing the ability of these approaches in predicting the survival of these patients.

Farid E Ahmed,[7]succeeded in using ANN fora prediction and also observation of colon cancer where a detailed accuracy analysis was performed. It was recognized, however, that accuracy must be exercised when designing, using and publishing biomedical results employing machine-learning devices such as ANNs in worldwide literature in order to enhance confidence in the quality and reliability of reported data. Such accuracy was, however, highly affected by the prognosis factors that were initially selected to determine the survival rate.

L. Zhu,et. al.[8]determined that the prognosis factors in any type of cancer can be done either statistically or using ANN. For example, the prognostic factors and their significance in gastric cancer patients were identified using the artificial neural network (ANN) and Cox regression hazard (CPH) models. A retrospective analysis was undertaken, including 289 patients with gastric cancer who had undergone gastrectomy. According to the CPH analysis, disease stage, peritoneal dissemination, radical surgery and body mass index (BMI) were selected as the significant variables while according to the ANN model, disease stage, radical surgery, serum CA19-9 levels, peritoneal dissemination and BMI were selected as the significant variables. In this research it was shown that the true prediction of the ANN was 85.3% and of the CPH model 81.9%.

A. Biglarian, et.Al. [9] used accuracy level obtained for every type of cancer. This motivated researchers try different ANN models for nonlinear regression of the survival rate prediction data .In this research it has been shown that in many cases the underlying assumptions of the model are not true, such as non-proportionality for the Cox model. It was also shown that choosing an appropriate model depends on the complexity and the characteristics of the data which in turn affects the appropriateness of the model. The predictions of the ANN and Cox models were compared by simulated data sets, which the average censoring rate were considered 20% to 80% in both simple and complex models.

S-H Lin,et. al.[10] investigated ANN for several other conditions that prognosis factors can be quantified easily so as to determine the ability of ANN to predict survival rate independently from the prognosis variables. In this research a comparison was made between the Logistic Regression Model and the Neural Network Model to predict the survival among intensive care unit patients. The study result showed that the ANN model has a better predicative ability than theLogistic Regression Model both with regard to the survivors and with regard to the entire population of patients studied

3. Proposed Model

Extending the concept of hierarchical parallel neurons and macro-neural networks, a new model is proposed in this research to predict and model the data as described in [11-12]. Instead of choosing a single big neural network with a variable (usually unknown) number of hidden layers that predicts the survival rate without taking the time as a variable in the prediction, a new macro-model based design is proposed in this research to predict the values of the input variables after only one year from the date of the diagnosis. The number of input variables to the macro neural network includes 10 variables chosen initially as well as couple of additional inputs. No hidden layers were assumed in such a model as only simple back-propagation is applied.Research into neural networks with no hidden layers has seen increasing interest in the past couple of decades [13-14]. In an attempt to decide the proper size for the hidden layer, a network structure optimization is conducted. The usual approach in this respect is to start with a network that is oversized. The number of redundant units in the hidden layer is then eliminated by introducing an additional cost function on a set of auxiliary linear response units. The extra cost function enables the auxiliary units to fuse together the redundant units on the original network [13]. In many cases such an approach results in no-hidden layers in the resulting network and that is what motivated this research to start with a simple model that has no hidden layers from the beginning.

In addition to the 10 prognosis variables, bias and time expressed in years from first diagnosis are entered as input variables. The entire input set is labeled S_i where $S_i = [P_i, b_i, t_i]$, P_i are the prognosis variables, b_i is the bias and t_i is the time from first diagnosis. The model only predicts the 10 prognosis variables after 1 year. Error in such a prediction is presumably small as the macro neural network does not predict the survival rate but rather the change in the prognosis variables over a time span of one year. The output set is labeled $S_o=[P_o]$ where P_o are the new predicted prognosis variables after one year. The new set of 10 variables can be used on a second similar macro NN to predict the second year prognosis variables. This process can be repeated 5 times for the consequent years. The key point here is that the errors in the predictions of each stage can be estimated and evaluated with some reasonable confidence bounds from values found by the follow-up studies. The overall survival rate prediction can be considered as a single output layer that is a function of the prognosis variables at the end of each of the 5 years as shown in Figure.1 thereby connecting the output of each macro neural network within one big function.

The Fundamental Benefits Of This Design Include The Following:

1. This design eliminates unknown hidden layersthat existed in pervious designsthereby reducing the elements of uncertaintyin designingthepredictionnetwork
2. The design uses multiple neural networks to predict the overall survival of patients. In this way it provides another degree of freedom in choosing what can be described as a heterogeneous rather than the normal homogeneous set of macro-models used in previous studies.
3. Each macro-model is essentially a simple neural network which does not requiremuch time to learn. Moreover the survival function is a complete independent neural network function that is based on top of the other macro-models.
4. This design solves fundamentally the problem of missing records and mortality among test subjects. Critically when a patient dies during the course of the study, his/her records for the coming years will naturally disappear thereby reducing the confidence interval and learning rate for the subsequent layers. This used to be a major source of error and inaccuracy in the previous

work predictions [7]. However with the current model only patients surviving throughout the year are used to train the network so the problem of lost records is basically waived.

5. The training of each macro-model is done independently using real input data from real records. That is to say that the macro-models shown in Figure.1 are disconnected from each other during the training phase. The input for each model is taken from the actual input data for surviving patients. Once the macro-models are fully trained the network is interconnected again and thus can be used directly to predict the survival of a patient during the coming 5 years.

The Proposed ModelMay Also Have Several Potential Disadvantages. ExamplesOf Those Include:

1. Propagation of error: As the different neural networks are connected progressively the error in prediction may propagate from one network to the other thereby it gets magnified and increases the overall erroneous predictions. Indeed such a problem has been observed in the simulations and may result in an unstable performance of the network. This problem can be controlled by the final survival prediction layer. An additional method will be presented also to control the error propagation problem.
2. Inconsistent learning: The network learning might become inconsistent and sensitive to the selection of the training set. Indeed if the training set has many dead patients during the five years then this means that the final stages will have only few patients for training. Although the learning rate of each macro network is generally high and the network needs only a small number of patients to converge, the resulting overall performance is still highly sensitive to the original set selected for the training.
3. Failed prediction: the back-propagation algorithm is a gradient descent algorithm meaning that it will follow a direction which will minimize the gradient of the overall error function. This error reduction is made with the hope of obtaining the global minimum rather than getting stuck in a local minimum. However, by dividing the network into sub-networks that work independently it is common to have multiple local minimum points. This is so because the training set of each macro network will be a subset of the training set used in the previous macro network.

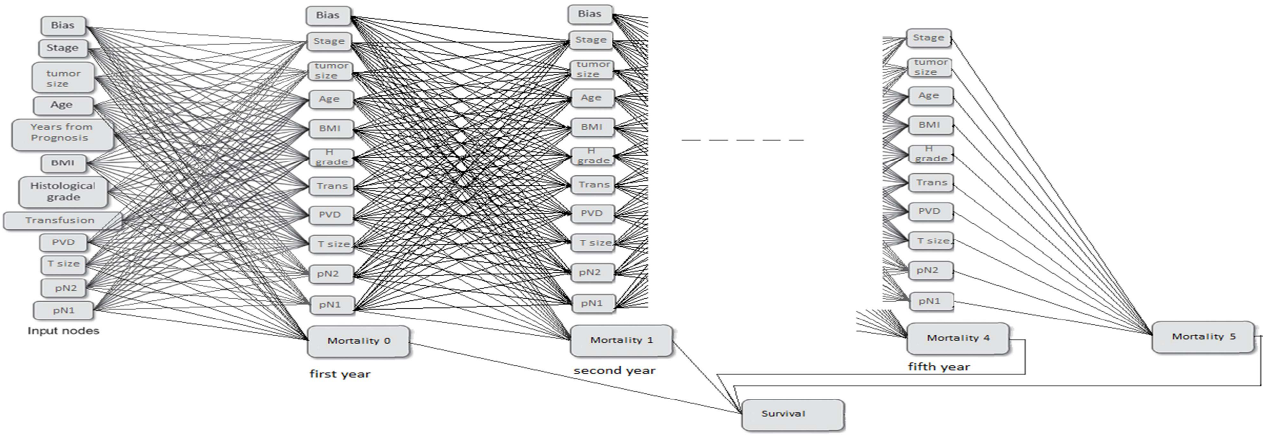


Figure..1 Proposed NN model

Set used in the previous macro network. In fact reaching a global minimum will become difficult under such conditions. However, it must be noted that the error function of each macro-network is not the same error function of the global network as each macro-network is trying to predict the change in 10 variables over a small period while the global network is trying to predict the overall survival rate. Indeed the simulation results confirmed that the rate of failed predictions is low and such a problem of getting a local minimum can be solved as will be explained later.The implementation details are presented next in order to further explain the work of the network and discuss the results.

4. Partitioning the dataset

The input vectors and the corresponding target vectors are used to train each macro network in theproposed modeluntil it can reasonably approximate a function that associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined initially. As networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities [15-16], it is easy to see the reason why such an arrangement was chosen for each macro-model.As was explained previously, standard back-propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule [17], in which the network weights are moved along the negative of the gradient of the performance error function. The term back-propagation refers to the manner in which the gradient is computed for nonlinear multilayer networks. Properly trained back-propagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs.

There are generally four steps in the training process:

1. Assemble the training data
2. Create the network object
3. Train the network
4. Simulate the network response to new inputs

There are two features of the Neural Network that are designed to improve network generalization - regularization and early stopping. These features and their use are discussed later.A sample of the dataset used is shown in Table.1.The complete set of input data for 2060 cases is obtained from the Surveillance, Epidemiology, and End Results (SEER) Program of the National Cancer Institute (NCI) [18].

Table.1 Sample input data for the first year showing the selected10 diagnosis variables

Case #	Tumoral size (cm)	PN	Age	BMI	PVD	Transfusion	Surgery	Dead after first year
1	1.9	3	72	20.16	1	1	1	0
2	3.5	2	58	35.27	0	0	1	0
3	7.2	2	56	26.89	1	0	1	0
4	3.1	3	82	19.76	0	0	1	1
5	3	2	66	23.62	1	1	1	0
6	3.7	3	71	19.55	1	0	2	0
7	3	2	69	24.47	1	1	1	0
8	4.8	3	82	26.95	1	1	1	1
9	4.3	3	81	32.56	0	1	2	1
10	3.7	1	75	18.72	0	1	1	0

As mentioned before the data set collectively represents our training and our verification sets combined. In order to get any meaningful results the data set should be randomly divided into three sets. The training set, the verification set and the control set. Each set represents the data for about 686 patients. The first set is used in training the network. The second set is used to test the performance of the network and obtain typical performance curves for data points that are not part of the training set. The final set is used in both the proposed model and an alternative survival prediction method in order to compare the two methods in a totally randomized input set [19].

Since the training will be supervised training [17], the results will be sensitive to the partitioning method. Thus each time the code is run the random function will randomly select a new set from the raw data to be our training set. Therefore only the very first random partitioning needs to be stored for later reproduction of the typical performance characteristics and for repeatability purposes. This will read the data in raw format as cells containing text, numeric and raw data and will represent them in Matlab as a matrix. This data will collectively need to be re-divided into the ten diagnosis variables which will be stored as 10 vectors. The data needs next to be pre-processed and conditioned to suit the neural network. Both of these functions will be explained in the next section. Finally the data variables are randomly divided into three sets by randomizing the indices of the vectors that will be put in each set. This is done by the function *randperm* which gives a random permutation of the set 1 to 686. Once this permutation is obtained the program will test to see if an older permutation is already stored. If this is so then the older permutation will be used while if not then the new permutation will be stored for later use. Figure.2 shows the flowchart of the procedure used to pre-process and pre-condition the data.

5. Assembling and Pre-processing of the data

As it is evident from inputs data, many of the input variables are not suitable as inputs to the proposed model neural network as they come in binary format or discrete format and not as continuous variables. The reason why those variables are not suitable can be traced back to the very definition of the perceptron function in equation(5.1) as follows [20].

$$f_N(x) = \sigma(\langle w|x\rangle) = \sigma\left(\sum_{i=0}^{k} w_i\, x_i\right) \qquad (5.1)$$

Where:

σ Activation Function

w_iWeights

x_i Inputs vector

This definition represents the output of the perceptron as a function of the input variables multiplied by the weights. Thus if the input variable is binary (i.e. 0 and 1) then this means the input weights will be totally canceled for a 0 input. Such a discontinuity in the input will alter training and stop convergence. To get around this problem a pre-processing of the data needs to be done so as to avoid zero inputs. For example to convert the binary input into a bi-sign variable (±1) rather than a 0/1 input. This will ensure that all the weights are present for any input combination and helps classifying the input categories. The best way to convert a binary variable into a bi-sign variable is to multiply the binary variable by 2 and subtract 1 from the result. The same reasoning applies to other discrete or categorical input variables. In all cases the zero input should be avoided in as much as possible to make sure that the corresponding weight will remain present during the training process. The output variable on the other hand, which represents mortality, can continue to be binary. Indeed the output of a neural network can represent binary outputs quite well with most decision functions. However since the mortality variable will be used as an input variable to the survival prediction neural network as shown in Fig.1 Then it too needs to be preprocessed to avoid the zero values as will be explained later.

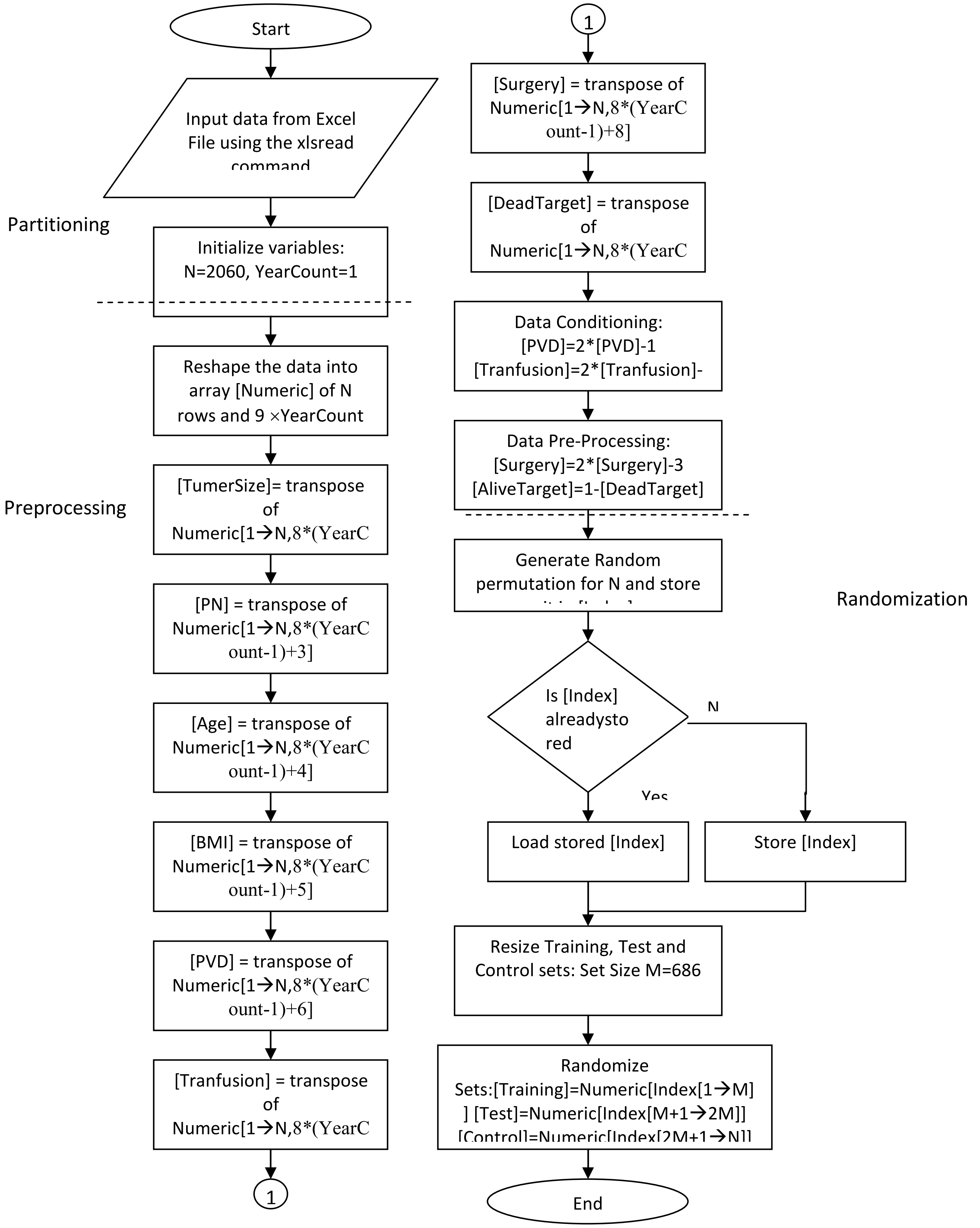
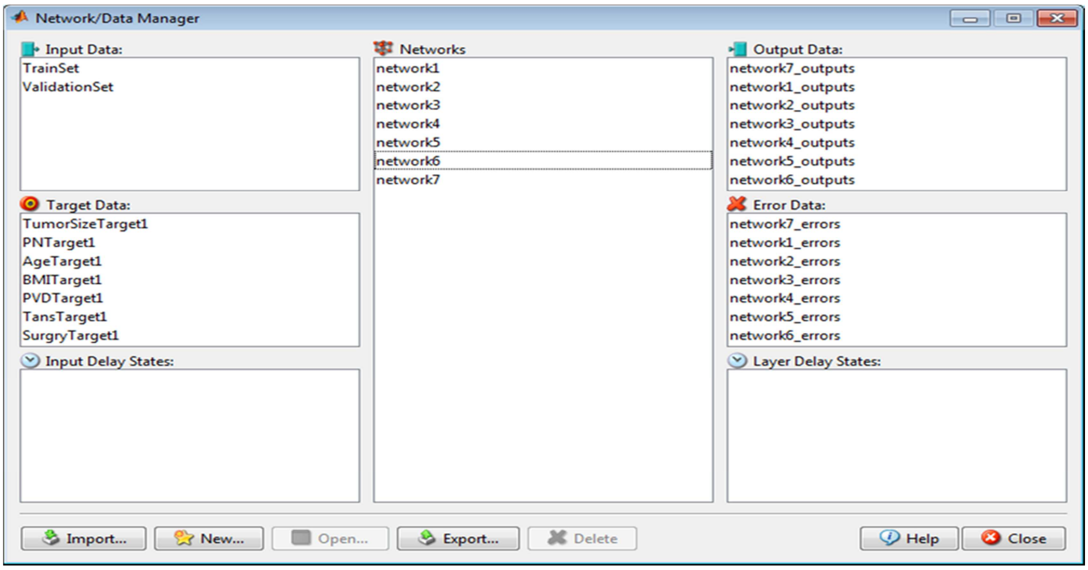

Start
Input data from Excel File using the xlsread command
Partitioning
Initialize variables: N=2060, YearCount=1
Reshape the data into array [Numeric] of N rows and 9 ×YearCount
Preprocessing
[TumerSize]= transpose of Numeric[1→N,8*(YearC
[PN] = transpose of Numeric[1→N,8*(YearCount-1)+3]
[Age] = transpose of Numeric[1→N,8*(YearCount-1)+4]
[BMI] = transpose of Numeric[1→N,8*(YearCount-1)+5]
[PVD] = transpose of Numeric[1→N,8*(YearCount-1)+6]
[Tranfusion] = transpose of Numeric[1→N,8*(YearC
1
1
[Surgery] = transpose of Numeric[1→N,8*(YearCount-1)+8]
[DeadTarget] = transpose of Numeric[1→N,8*(YearC
Data Conditioning: [PVD]=2*[PVD]-1 [Tranfusion]=2*[Tranfusion]-
Data Pre-Processing: [Surgery]=2*[Surgery]-3 [AliveTarget]=1-[DeadTarget]
Generate Random permutation for N and store
Randomization
Is [Index] alreadystored
N
Yes
Load stored [Index]
Store [Index]
Resize Training, Test and Control sets: Set Size M=686
Randomize Sets:[Training]=Numeric[Index[1→M]] [Test]=Numeric[Index[M+1→2M]] [Control]=Numeric[Index[2M+1→N]]
End

Figure.2 Flowchart of data partitioning, pre-processing and randomization

6.Training the Macro-Models

Once the model has been created, training the network is performed by separating each macro neural network and training each one individually. As will be shown in the next section, error propagation will be different for different stages. This will require special treatment for each stage with a different learning algorithm and a learning rate. Specifically, batch learning mode with momentum will be used in the first two stages while batch gradient descent will be used in subsequent stages. The code generates threesetsof data. The final set will be used for comparison with conventional methods of survival prediction. It must be noted here that since prediction of the 10 variables is being done in each stage we have effectively 10 neural networks in each stage. Luckily the training is simplified by the use of the training wizard supplied by the nntool as shown in Figure.3


Network/Data Manager

Input Data:
TrainSet
ValidationSet

Networks
network1
network2
network3
network4
network5
network6
network7

Output Data:
network7_outputs
network1_outputs
network2_outputs
network3_outputs
network4_outputs
network5_outputs
network6_outputs

Target Data:
TumorSizeTarget1
PNTarget1
AgeTarget1
BMITarget1
PVDTarget1
TansTarget1
SurgryTarget1

Error Data:
network7_errors
network1_errors
network2_errors
network3_errors
network4_errors
network5_errors
network6_errors

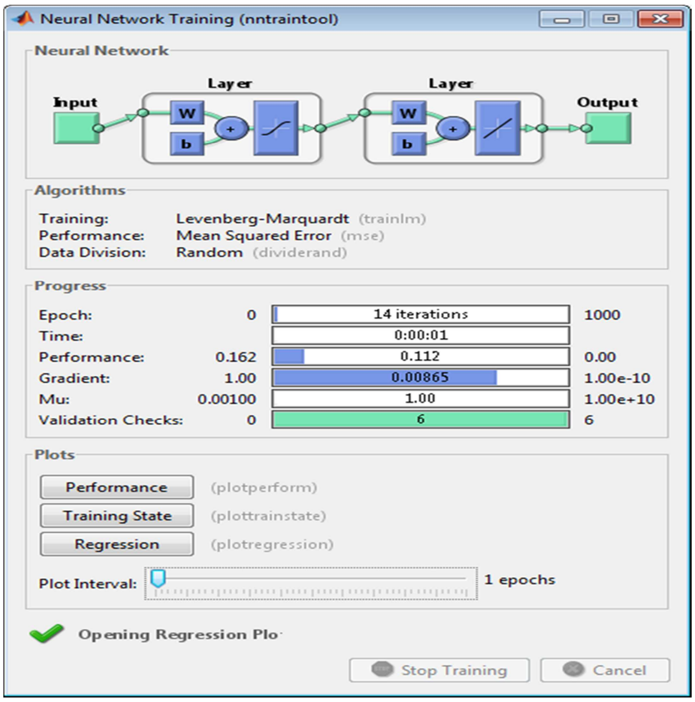
Input Delay States:

Layer Delay States:

Figure.3 Training of the first stage macro-model neural network

This tool makes things quite easy in addition to providing good graphical tool. The standard network that was used for the modeling of each macro-model show asin Figure.4.

A final note in this section is regarding the consideration of the patient age as a prognosis variable. In this research this variable was used following [1,21]. It might seem quite unfruitful to have the age (in number of years) as a prediction variable in each macro-model given that this variable should only increase by one from one stage to the next. However, practical simulation showed that the predicted value of this variable will not be an integer even though the target values are integers. In fact it was verified that the overall performance of the classifier improved by an appreciable amount with the presence of this variable. The main reason is attributed to the increase in confidence interval provided by each statistical function of the input variables.



Algorithms

Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Data Division: Random (dividerand)

Progress

Epoch:	0	14 iterations	1000
Time:		0:00:01	
Performance:	0.162	0.112	0.00
Gradient:	1.00	0.00865	1.00e-10
Mu:	0.00100	1.00	1.00e+10
Validation Checks:	0	6	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Opening Regression Plo·

Figure.4 Macro-model structure for each network in each stage

7.Battling through Error Propagation

As described previously one of the main problems with the proposed model is the error propagation. Specifically, as small errors are generated in the predicted values of the first stage, they tend to increase and get amplified by subsequent stages. This is to be expected as each stage is trained independently from the subsequent stages and no global error control has been enforced on the entire model during the training process. To address this problem an error propagation model was first found based on the results of the simulation. Figure.5 shows the absolute relative error generated by the first stage as it predicts the tumor size after one year. The relative error that reaches 1 (or 100%) is the result of wrong prediction of survival while any error less than 1 is the result of erroneous predication of the tumor size. As it is evident such an error (although occurring in less than 6% of the cases) may exceed 50%. Of course when such an error is passed to the subsequent stages, it will immediately reach 100%.

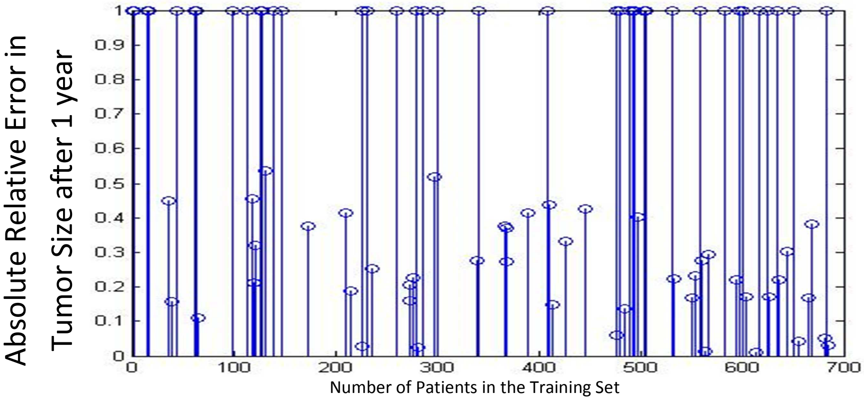


Figure.5 Relative Absolute Error generated from the First Stage

The problem becomes more severe as the error self-generated by the second stage is also added to the result. The key point here is that due to reduction of records because of death among the patients the error contribution in the second stage will exceed 6% and tends to occur in about 10% of the cases. Thus the self-contribution of the error for each stage is increasing because of reduction in the sample size. Added to the self-contribution error is the induced-error due to previous stages. Thus the total relative error in survival rate based solely on predictions will increase up to 60% at the final stage rendering the entire model useless as shown in Figure.6.

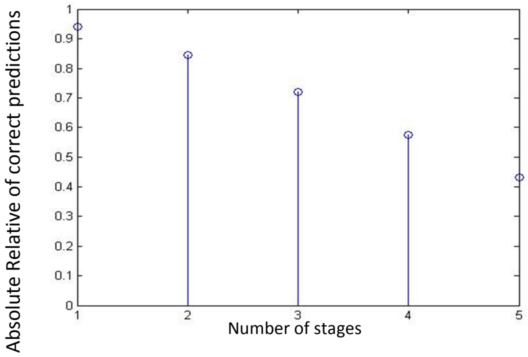


Figure. 6. Relative Absolute of correct survival predictions in each stage

For the purpose of solving this problem it is important to note that the first stage has the highest contribution to the final error. Thus by enforcing more strict learning rules and much slower learning rate on first macro-neural network, the first-stage enforced error will be reduced. This has proven to apply to other stages as well. However, this will only decrease the total amount of error by a small amount as another unpredicted error component also exists. The important point to remember here is that only the survival rate is needed from this model and not the actual tumor size, body mass index, or any of the predicted input variables. Thus the mortality rate MR from the first stage will be dealt with in the most conservative way as this rate is used to predict the survival after five years in the final stage. To this end a formula has to be applied to shape the mortality rate before being used in the final stage. Such formula already exists in multistage multiple hypotheses testing and has been used successfully in the proposed model in equation (7.1) as follows [22].

$$CorrectedMR_i = \frac{C \times MR_i}{\prod_{j=1}^{i-1} MR_j} \qquad (7.1)$$

Where:

C is constant that is between 0 and 1.

The best values for the constant C in each stage found empirically are shown in Table.2.

Table.2 Best values for the constant C as found by simulation

Stage	1	2	3	4	5
C value	0.2	0.4	0.7	0.9	0.95

Accordingly the percentage of correct survival predictions has increased substantially as shown in Figure.7

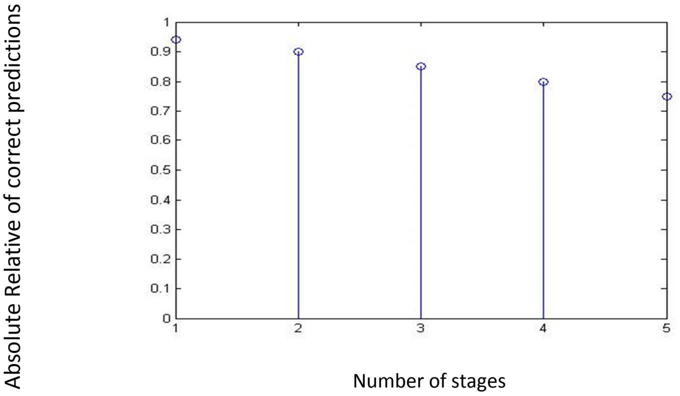


Figure.7 Relative Absolute of correct survival predictions in each stage after correction

8. Performance Evaluation of the Proposed Model

To compare the performance of the proposed method to the conventional methods we need to define some probability measures. However, the comparison has to take into account that some of the chosen variables are discrete and some are continuous. Specifically the tumor size, the body mass index and the patient age are considered continuous variables while all the other variables are discrete. The continuous variables need to be classified first and plotted into histograms. As an example, Figure.8 shows the histogram of the tumor size, the body mass index and the patient age

It is evident from Fig. 8 that the selected data from SEER site does not really represent the distribution of patients in the third world and are mainly representative of the patients in the United States as most of the data in the SEER site are mainly collected in the US. In this respect the age parameter for example clearly indicates that although non-small cell lung cancer NSCLC might develop earlier while the patient is much younger, it is only detected at later stages in life when the disease has reached an advanced stage which makes the treatment ineffective thereby contributing to the high death rate associated with this disease.Once the parameters are classified, we need to define the accuracy, the specificity and the sensitivity of the proposed model based on the probability of death having one input variable fixed. To this aim we have to find the conditional probability of death while fixing each variable. For example the In the same sense the probability of death having the patient age in the range 45-50 year will be Pr(death/45<age<50). Next we need to define a threshold for comparison and in this case it will be considered the median of the variables and we find the median probability of death where the median probability is the probability of having 50% of the test subjects dying given the that the test variable is fixed. Thus the accuracy, specificity and sensitivity of the proposed model will be defined for the output of the proposed model as that modified the equations bellow as follows [23].

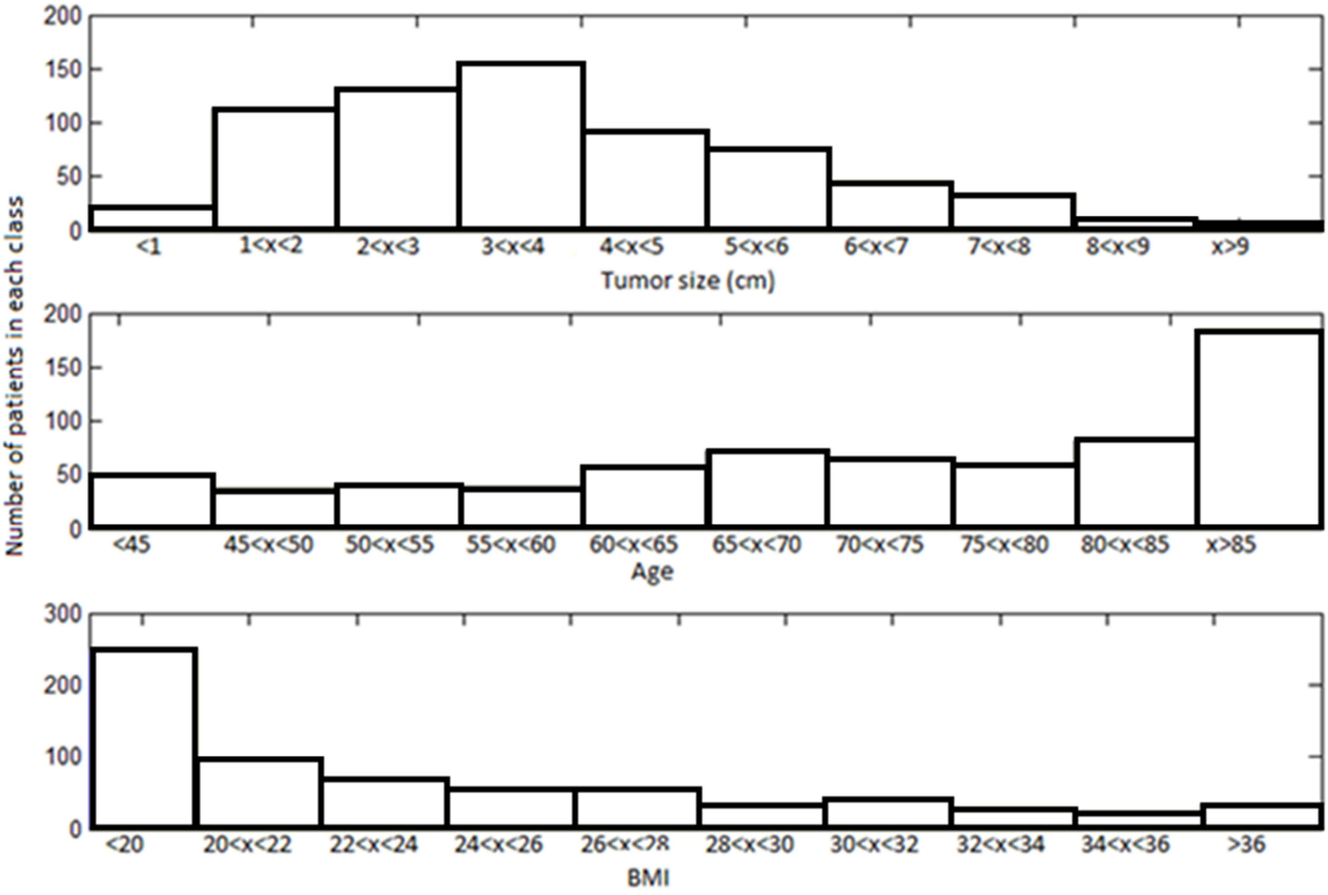


Figure.8Histogram of the Tumor size, Age and BMI input variables

$$accurary = \frac{\sum_{p_i > p_m} q_i + \sum_{p_i < p_m} (1 - q_i)}{total\ number\ of\ patients\ in\ the\ set} (8.1)$$

probability of death having tumor of size less than 2 cm will be defined as Pr(death/tumor size < 2cm). $specificity = \frac{\sum_{p_i < p_m} (1 - q_i)}{\sum_i (1 - q_i)}$ (8.2)

$$sensitivity = \frac{\sum_{p_i > p_m} q_i}{\sum_i q_i} (8.3)$$

Where:

p_m is the median probability, p_i is the predicted probability of death with any input variable kept constant and q_i is the probability of wrong predictions (prediction of survival while the patient dies or prediction of death while the patient survives). The expected number of correct predictions will therefore be given by

$$E(number\ of\ correct\ predictions) = \sum_{p_i > p_m} p_i + \sum_{p_i < p_m} (1 - p_i) (8.4)$$

9Results of the Simulation

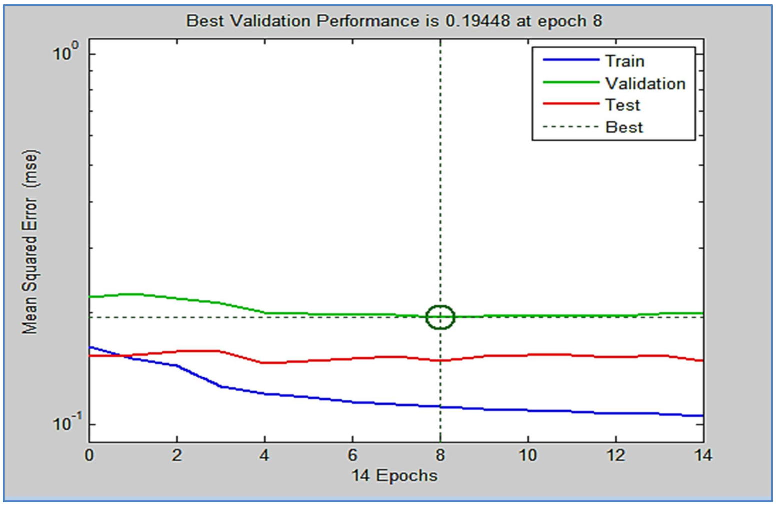
The experimental were done using a computer system having the following specification of hardware and software respectively, Intel Core i-5 second edition, 4GB RAM and Windows 7 Ultimate 64 bit. Statistical analyses were performed using Matlab statistics and data toolbox software. From theses analyses only 10 variables were considered of high significance in determining the survival rate for NSCLC.The performance of the neural network model is given in terms of theMSE(mean squared error) and %E (the percentage error) , also given in Table.3. The calculation was stopped after 14 epochs asshown in Fig.6. The best validation performance was at epoch 8. The computation time was relatively short, mainly ascribed to the simplified macro-model structure and network size.

Table.3 Data Sizes and Performance of each stage

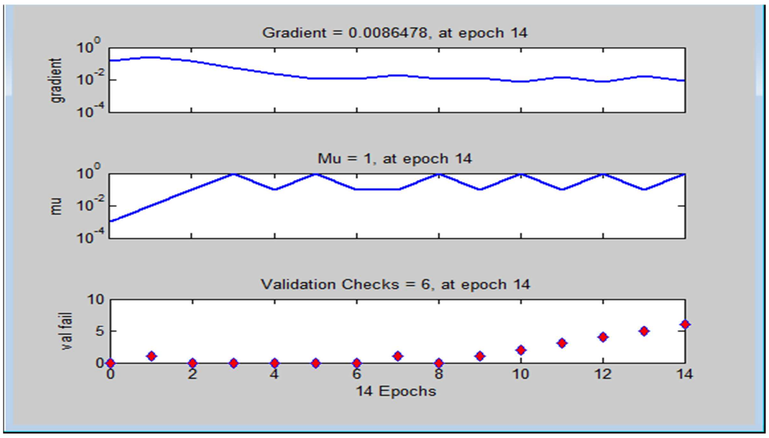
Stages	Sample size	MSE	%E	Accuracy %	Specificity %	Sensitivity %
Stage1	**Sample size**	**MSE**	**%E**	**Accuracy %**	**Specificity %**	**Sensitivity %**
Training	**686**	**0.04**	**2.57**	**98.54**	**99.3**	**98.9**
Validation	**686**	**0.45**	**23.3**	**96.21**	**97.8**	**96.9**
Stage2	**Sample size**	**MSE**	**%E**	**Accuracy %**	**Specificity %**	**Sensitivity %**
Training	**404**	**0.07**	**5.23**	**92.33**	**93.1**	**94.2**
Validation	**413**	**0.58**	**28.1**	**87.4**	**89.7**	**88.1**
Stage3	**Sample size**	**MSE**	**%E**	**Accuracy %**	**Specificity %**	**Sensitivity %**
Training	**226**	**0.13**	**9.44**	**87.5**	**89.3**	**90.68**
Validation	**232**	**0.66**	**33.3**	**82.3**	**84.1**	**86.3**
Stage4	**Sample size**	**MSE**	**%E**	**Accuracy %**	**Specificity %**	**Sensitivity %**
Training	**104**	**0.16**	**12.3**	**84.3**	**85.1**	**86.2**
Validation	**132**	**0.72**	**38.4**	**79.7**	**81.4**	**82.9**
Stage5	**Sample size**	**MSE**	**%E**	**Accuracy %**	**Specificity %**	**Sensitivity %**
Training	**63**	**0.19**	**15.8**	**79.8**	**80.1**	**81.2**
Validation	**88**	**0.78**	**44.1**	**75.3**	**76.4**	**77.9**

As shown in table.3 the error is increasing quickly due to the error propagation problem. It can also be noted that smaller learning rate and different learning algorithm did improve the results slightly from those that were predicted in the previous section.From table 3 it can also be seen how the accuracy is degrading fast with even slight decrease of specificity and sensitivity. The highest accuracy is achieved with the same training data as it is expected. However, with the validation test set the accuracy of the model did not degrade by a large amount and remain relatively high

The convergence of the learning algorithm for batch gradient descent for a sample input variable is shown in Figure.8. As can be seen the convergence is quite easily obtained as the gradient descends to zero quite quickly. The simplicity of the model and the fact that many inputs come in discrete format makes things much simpler for each network. However, this has its toll on the number of required simplified neural network models. Due to the oversimplification, the proposed model needs a lot of neural networks to perform well. The ability of the proposed model to discriminate between data points and perform classification is best represented in Figure.9. As can be see the data points are distributed in a very difficult arrangement that makes finding a classification line quite difficult. However, since multi-stage macro-models are used, the classification process in each stage can be simplified and the burden to differentiate closely located points is waived to be handled by the next stage. This makes the model ideal in solving rather difficult problems.



(a)



(b)

Figure.9Performance of the Training for the Transfusion input variable (a) Mean square error (b) Gradient of error function and performance of learning algorithm

The quality of the classifier was assessedby ROC (Receiver-Operator Characteristic) curves as shown in figure.10. The ROC curve is a plot of the true positive rate in apposition to the false positive rate as the threshold varies. The area under the curve of All ROC is 0.829 (lower right of Figure.10), which is calculated by taking all the ANN inputs into account, i.e., all the input variables. As a reference, the ROC curve obtained based on statistical method for only one prognosis variable (tumor size) is provided in Figure.12 and the corresponding area under the curve is 0.723. As a matter of fact plotting the ROC curves using the conventional methods for any input variable did not perform as well as the proposed model.

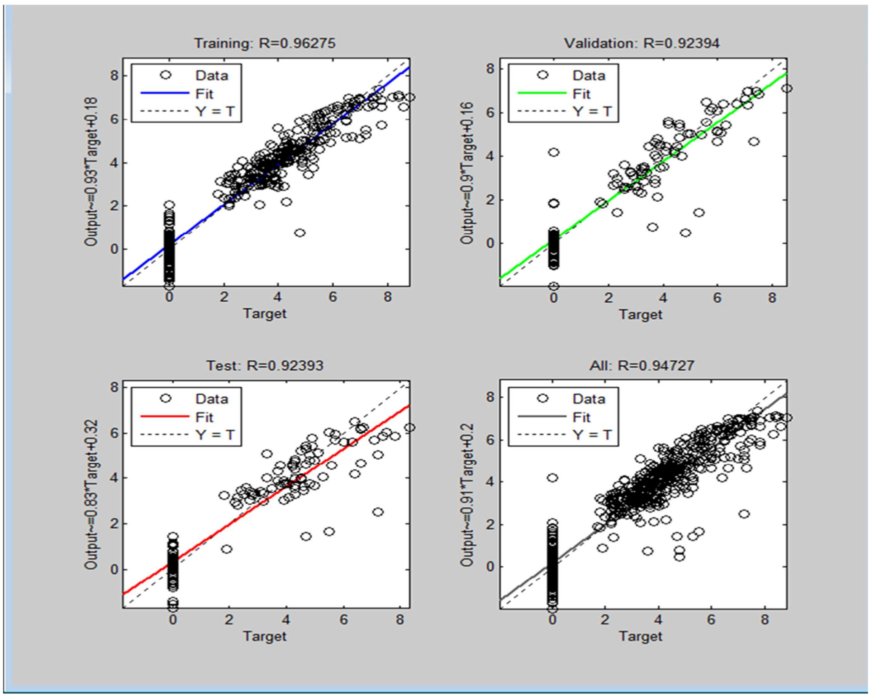


Figure.10 Classification performance of the model

In this paper there is the absence of follow-up time of the cases. Generally this introduces some form of error called information bias. In other words, the way neural network generates outputs is difficult to understand for non-specialized user. Because of that, heuristic statistical identification of prognosis variables as inputs for the proposed model was selected as a suitable approach. Adoption of the findings obtained previously from statistical analysis which suggest that age, body mass index BMI, tumour stage, surgical approach and transfusion are the risk factors was done. These parameters are then employed as the inputs of the proposed neural network model.

10. Conclusion

In this paper an attempt was made to address several limitations of neural network models in the predication of survival rate for lung cancer by proposing a new neural network model. The proposed model is designed without any hidden layers to avoid data over-fitting problem. However, due to the over-simplification in the proposed model, error propagation and convergence to a local minimum affected the overall performance significantly. Further analysis of such error showed that solutions to these problems were possible by making use of the temporal variable variations to correct for the prediction as well as using more strict learning rules for the first stages as they produce the highest contribution to the error. Such corrections improved the model significantly and the simulation results showed a performance enhancement making the overall performance comparable to that produced by conventional neural network models. Thus with the proposed model the data over-fit problem was solved while the high prediction performance remained unaffected and thereby the entire system reliability was improved.

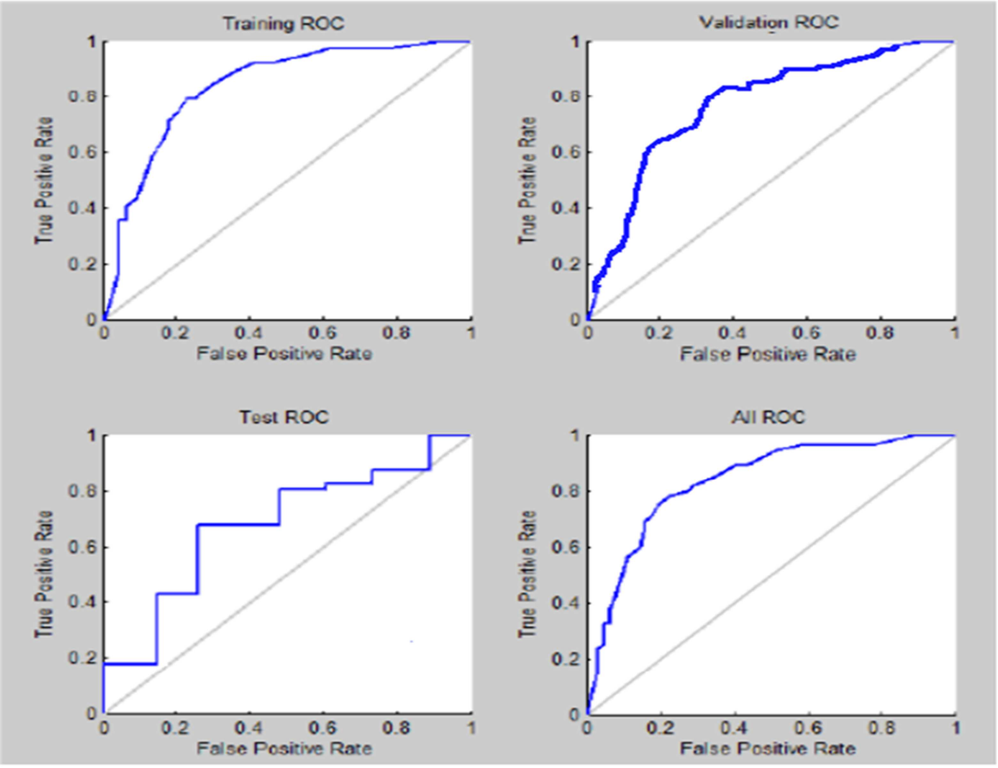


Figure.11 Overall ROC analysis of the neural network.

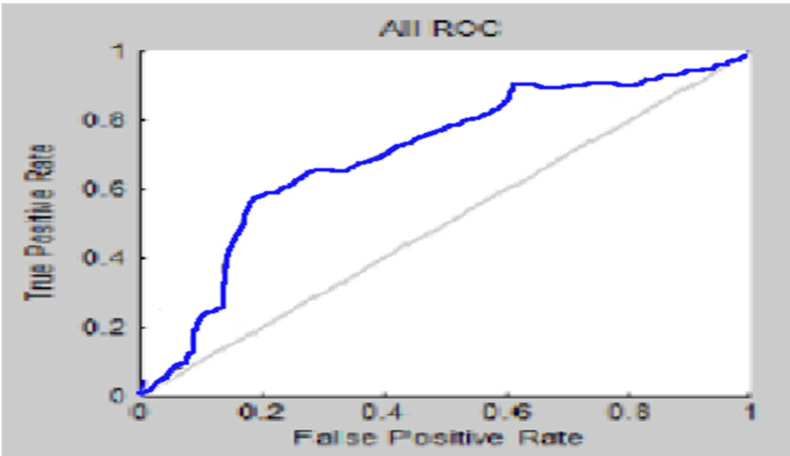


Figure.12 ROC Analysis of Tumor Size Using Classical KM Analysis

References

1. F. Tas, R. Ciftci, L. Kilic and S. Karabulut, “Age is a prognostic factor affecting survival in lung cancer patients”, Oncology Letters, Volume 6, pp 1507-1513, 2013.
2. C. M. Wilson, S. Tobin and R. C. Young, “The exploding worldwide cancer burden: the impact of cancer on women”, International Journal of Gynecol Cancer, Vol. 14, pp 1-11, 2004.
3. D. Yue, et al, “Survival analysis of patients with non-small cell lung cancer who underwent surgical resection following 4 lung cancer resection guidelines”, Journal of BMC Cancer Vol. 14, p. 422, 2014.
4. S. Abbasi and A. Badheeb, “Prognostic Factors in Advanced Non-Small-Cell Lung Cancer Patients: Patient Characteristics and Type of Chemotherapy”, Lung Cancer International Journal, Volume 2011, Article ID 152125, 2011.
5. A. Agrawal, S. Misra, R. Narayanan, L. Polepeddi, and A. Choudhary, “A lung cancer outcome calculator using ensemble data mining on SEER data ”, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. August 2011.
6. *A Biglarian, E Hajizadeh, A Kazemnejad, and MR Zali,“*Application of Artificial Neural Network in Predicting the Survival Rate of Gastric Cancer Patients”**,** Iranian Journal of Public Health, Vol 40, issue 2, pp 80-86, 2011
7. F. E. Ahmed, “Artificial neural networks for diagnosis and survival prediction in colon cancer”, BioMed Central Journal: Molecular Cancer, Vol. 4, Issue 29, August 2005.
8. L. Zhu et. al, “ Comparison between artificial neural network and Cox regression model in predicting the survival rate of gastric cancer patients”, Journal of Biomedical Reports, pp 757-760, 2013.
9. A. Biglarian, et. al, “Nonlinear Survival Regression Using Artificial Neural Network”, Journal of Probability and Statistics, Vol. (2013), Article ID 753930, 2013.
10. S-H Lin et. al, “ A comparison of MICU survival prediction using the logistic regression model and artificial neural network model”, The journal of nursing research, vol. 14 issue 4, pp. 306-314, 2007.
11. KM. Lam, XJ. He, and KS. Choi, “Using Artificial Neural Network to Predict Mortality of Radical Cystectomy for Bladder Cancer”, SmartComp proceedings, IEEE Computer Society Digital Library CSDL, pp 201-207, 2014.
12. PF. Bassi, E. Sacco, V. De Marco, M. Aragona and A. Volpe, “Prognostic accuracy of an artificial neural network in patients undergoing radical cystectomy for bladder cancer: a comparison with logistic regression analysis”, BJU International Journal Compilation, Volume 99, pp 1007-1012, 2007.
13. A. Burkitt, “Optimization of the Architecture of Feed-forward Neural Networks with Hidden Layers by Unit Elimination”, Journal of Complex Systems, Issue 5, pp 371-380, 1991.
14. S. Karsoliya, “Approximating Number of Hidden layer neurons inMultiple Hidden Layer BPNN Architecture”, International Journal of Engineering Trends and Technology, Vol. 3, Issue 6, 2012.
15. H. Demuth and M. Beale, *Neural Network Toolbox For Use with MATLAB*, Mathworks user’s guide version 4, MathWorks Inc. 2002.
16. M. Hagan, H. Demuth and M. Beale, *Neural Network Design*, PWS Publishing Co. Boston, MA, USA,1996
17. C. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics series, Springer Science+Business Media, 2006.

18. Program of the National Cancer Institute (NCI) website, World Wide Web resources http://seer.cancer.gov/seerstat/analysis.html
19. D. Delen, G. Walker, A. Kadam, “Predicting breast cancer survivability: a comparison of three data mining methods”, J. of Artificial Intelligence in Medicine, DTD 5, July 2004.
20. I .Guyon, S.Gunn, M. Nikravesh, LA.Zadeh, *Feature Extraction:Foundations and Applications* , publisher:springer, Aug.29,2006.
21. Aaron C. Denson, MD, and Amit Mahipal,”Participation of the Elderly Population in Clinical Trials: Barriers and Solutions”, Cancer Control Journal, Vol. 21, No. 3, July 2014.
22. J. Bartroff, T. L. Lai, “Multistage tests of multiple hypotheses”, Communications in Statistics—Theory and Methods, Vol 39, pp. 1597–1607, 2010.
23. R. M. Ripley, “Neural Network Models for Breast cancer Prognosis”, PhD Thesis, St Cross College, Trinity Term, pp.81,1998**.**