Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.4, No.7, 2013

Decision Support System Using Decision Tree and Neural Networks

L. G. Kabari[1*] and E. O. Nwachukwu[2]
1. Computer Science Department, Rivers State Polytechnic, Bori, PMB 20, Rivers State, Nigeria
2. Computer Science Department, University of Port Harcourt, Port Harcourt, PMB 5323, Nigeria
* E-mail of the corresponding author: ledisigiokkabari@yahoo.com

ABSTRACT
Decision making in a complex and dynamically changing environment of the present day demands a new techniques of computational intelligence for building equally an adaptive, hybrid intelligent decision support system. In this paper, a Decision Tree-Neuro Based model was developed to handle loan granting decision support system and clinical decision support system(Eye Disease Diagnosis) which are two important decision problems that requires delicate care. The system uses an integration of Decision Tree and Artificial Neural Networks with a hybrid of Decision Tree algorithm and Multilayer Feed-forward Neural Network with backpropagation learning algorithm to build up the proposed model. Different representative cases of loan applications and eye disease diagnosis were considered based on the guidelines of different banks in Nigeria and according to patient complaint, symptoms and physical eye examinations to validate the model. Object-Oriented Analysis and Design (OO-AD) methodology was used in the development of the system, and an object-oriented programming language was used with a MATLAB engine to implement the models and classes designed in the system. The system developed, gives 88% success rate and eliminate the opacity of an ordinary neural networks system.
Keywords: Decision Tree-Neuro Based Model, Backpropagation Learning Algorithm, Object-Oriented Analysis and Design, MATLAB Embedded Engine, Loan Granting, Eye Diseases Diagnosis.

1. Introduction

A decision support system (DSS) is a computer-based information system that supports business or organizational decision-making activities. DSSs serve the management, operations, and planning levels of an organization and help to make decisions, which may be rapidly changing and not easily specified in advance.
Decision makers are faced with increasingly stressful environments – highly competitive, fast-paced, near real-time, overloaded with information, data distributed throughout the enterprise, and multinational in scope. The combination of the Internet enabling speed and access, and the maturation of artificial intelligence techniques, has led to sophisticated aids to support decision making under these risky and uncertain conditions. These aids have the potential to improve decision making by suggesting solutions that are better than those made by the human alone.
They are increasingly available in diverse fields from medical diagnosis to traffic control to engineering applications. The granting of loans by a financial institution (bank or home loan business) is one of the important decision problems that require delicate care. Wrong decision may lead to bank distress and even near collapse of national economy which may require huge bail-outs from government to keep the economy going.

In this paper, decision tree-neuro based model will be developed for the loan granting decision support system and eye disease diagnosis, which is a hybrid of decision tree and neural network. This hybrid approach enables us to build rules for different groups of borrowers and eye diseases separately. In the first stage, bank customers or eye diseases are segmented into clusters, that are characterized by similar features and then, in the second step, for each group, decision trees are built to obtain rules that are feed into the neural net for indicating clients expected not to repay the loan or presence or absence of an eye disease. The main advantage of applying the integration of two techniques consists of building models that, may better predict risk connected with granting credits for each client with good explanations, than while using each method separately. The developed model was analyzed and designed using appropriate tools and the implementation was carried out using C programming language with an embedded MATLAB engine i.e. C programming was used with its classes to call MATLAB tools at different stages.

2.0 Literature Review

An artificial neural network (NN) is a computational structure modelled loosely on biological process. NNs explore many competing hypothesis simultaneously using a massively parallel network composed of non-linear relatively computational elements interconnected by links with variable weights. It is this interconnected set of weight that contains the knowledge generated by the Neural Networks(NN). NNs have been successfully used for low- level cognitive tasks such as speech recognition and character recognition. They are been explored for decision support and knowledge induction [1][2][3] . In general, NN models are specified by network topology, node characteristics, and training or learning rules. NNs are composed of large number of simple processing units, each interacting with others via excitatory or inhibitory connections. Distributed representation over a large number of units, together with interconnectedness among processing units, provides a fault tolerance. Learning is achieved through a rule that adapt connection weights in response to input patterns. Alterations in the weights associated with the connections permits adaptability to new situations [4]

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal. Another use of decision trees is as a descriptive means for calculating conditional probabilities.

In financial institutions Loan applications can be categorized into good applications and bad applications. Good applications are the applications that are worthy of giving the loan. Bad applications are those ones that should be rejected due to the low probability of the applicants ever returning the loan. The institution usually employs loan officers to make credit decisions or recommendations for that institution. These officers are given some hard rules to guide them in evaluating the worthiness of loan applications. After some period of time, the officers also gain their own experiential knowledge or intuition (other than those guidelines given from their institution) in deciding whether an application is loan worthy or not. Generally, there is widespread recognition that the capability of humans to judge the worthiness of a loan is rather poor [5]. Some of the reasons are:

(i) There is a large gray area where the decision is up to the officers, and there are cases which are not immediately obvious for decision making;

(ii) Humans are prone to bias, for instance the presence of a physical or emotional condition can affect the decision making process. Also personal acquaintances with the applicants might distort the judgmental capability;

(iii) Business data warehouses store historical data from the previous applications. It is likely that there are knowledge hidden in this data, which may be useful for assisting the decision making. Unfortunately, the task of discovering useful relationships or patterns from data is difficult for humans [6]. The reasons for such difficulties are the large volume of the data to be examined, and the nature of the relationships themselves that are not obvious.

Given the fact that humans are not good at evaluating loan applications, a knowledge discovery tool thus is needed to assist the decision maker to make decisions regarding loan applications. Knowledge discovery provides a variety of useful tools for discovering the non-obvious relationships in historical data, while ensuring those relationships discovered will generalize to the new/future data [7][8]. This knowledge in the end can be used by the loan officers to assist them in rejecting or accepting applications. Past studies show that even the application of a simplistic linear discriminant technique in place of human judgment yields a significant, although still unsatisfactory increase in performance [5]. Treating the nature of the loan application evaluation as a classification [9] and forecasting problem [10], have also been offered.

Techniques in literature ranges from traditional statistical methods like logistic regression [11], k-nearest neighbor [12], classification trees [13] or simple neural network models [14][15][16], as well as cluster analysis [17][18][19] to combination of methods to obtain stronger general rules[20].The combination approach allowed for connecting two kinds of representation knowledge and for formulating rules for a set of typical examples.

The use of expert system as a mean of conducting medical diagnosis and recommending successful treatments has been a highly active research field in the past few years. Development of medical expert system that uses artificial neural networks (ANN) as knowledge base appears to be a promising method for diagnosis and possible treatment routines. One of the major applications of medical informatics has been the implementation and use of expert systems to predict medical diagnoses based upon a set of symptoms [21]. Furthermore, such expert systems serve as an aid to medical professionals in recommending effective laboratory tests and treatments of diseases. An intelligent computer program assisting medical diagnosis could provide easy access to a wealth of information from past patient data. Such a resource may help hospitals reduce excessive costs from unnecessary laboratory test and ineffective patient treatment, while maintaining high quality of medical care. One major drawback of conventional medical expert systems is the use of static knowledge base developed from a limited number of cases and a limited population size, demographics, and geographic location. The knowledge base is inherently not dynamic and is not routinely updated to keep up with emerging trends such as the appearance or increased prevalence of unforeseen diagnoses. The result is that, after a given period of time this inflexibility limits the use of the knowledge base as it no longer reflects the current characteristics of the population at risk.

In [22], Clinical Decision Support System (CDSS) using a hybrid of neural networks and decision trees for the diagnosis of eye diseases was presented. Neural networks are first trained and then combined with decision trees in order to extract knowledge learnt in the training process. Artificial neural networks are used for the diagnosis of selected eye diseases according to patient complaint, symptoms and physical eye examinations. The trained network diagnosis the eye diseases according to the knowledge the network acquired by learning from previous eye diseases. After successful training, knowledge is extracted from these trained networks using decision trees in the form of 'if-then' rules. The system can be used by ophthalmologists to minimize unnecessary laboratory tests and reduce operational costs in treatment of eye diseases. The general paradigm can be applied to other categories of diseases.

3.0 Methodology

In this paper, an integration of Decision Tress and Neural Networks is used to form a hybrid called Decision Tree-Neuro Based Decision Support System (DTNBDSS). The implementation of the design was carried out on two areas:- Loan granting decision support system(LGDSS) and Clinical Decision Support System(Eye Diseases Diagnosis)

3.1 Analysis of the Method used

When decision trees and neural networks are compared, one can see that their advantages and disadvantages are almost complementary. For instance knowledge representation of decision tree is easily understood by humans, which is not the case for neural networks; decision trees have trouble dealing with noise in training data, which is again not the case for neural networks; decision trees learn fast and neural networks learn relatively slow, etc. Therefore, the idea is to combine decision trees and a neural network in order to combine their advantages seems to be a welcome research area.

Artificial neural networks (ANN) are very efficient in solving various kinds of problems. But Lack of explanation capability (Black box nature of Neural Networks) is one of the most important reasons why artificial neural networks do not get necessary interest in some parts of industry [23]. Even though neural networks have huge potential we will only get the best of them when they are integrate with other computing techniques, fuzzy logic and so on[24].

Decision Tress on the other hand simple to understand and interpret. People are able to understand decision tree models after a brief explanation, but decision-tree learners can create over-complex trees that do not generalize the data well. This is call overfitting, mechanisms such as pruning are necessary to avoid this problem.

3.2 Decision Tree Neuro-Based Model Design

The development of decision tree neuro-based model is to fine tune the intelligence decision making progress in a complex computational and organizational decision making process so that it can adapt to the present day complex and dynamically changing environment. Financial decision making have increasingly become far more challenging, on the other side the ever changing nature of our environment is bring new eye diseases every day hence, making the use of decision trees less efficient in handling such increasing complexity of decision making process based on increasing volume of data required in decision making

In the design of the system model, there are two major parts to the system as illustrated in the decision tree neuro-based architecture in figure3.3. The first part is the decision tree part which handles decision making based on the fundamental decision rules. At the end the decision tree output becomes the input to the neural net which uses the result as a pad to start-up further refinement that we believe resulted in a much high level of accuracy in decision-making.

Ordinarily humans are the ones that will select an action from a decision tree based on the suggested lines or actions in complex organizational conditions. These conditions are built-in into the algorithm that reduces the complexity and the computational effort required by system in arriving at a given result based on the sample input and the resultant output from the tree. The processing effort of the neural net carries out further refinement based on the level of processing done by the decision tree. The decision tree then acts as the first processing engine for the system. In the model presented, the neural net takes over and completes the final decision making process from the point that complex decisions stop. This greatly reduces possibilities of error in decision making process involving complex organizational situations or in areas where wrong a decision leads to irredeemable catastrophic consequences such as loan granting.

The procedure of designing a neural network model is a logical process. The process was not a single-pass one, but it required going back to previous steps several times (see figure1). The neural net hidden layer processes the input variables based on the algorithm and the weight of the threshold offered to the system. The computation of the hidden layer is based on various trials until the error is minimized. The overall design is as shown in figure2.

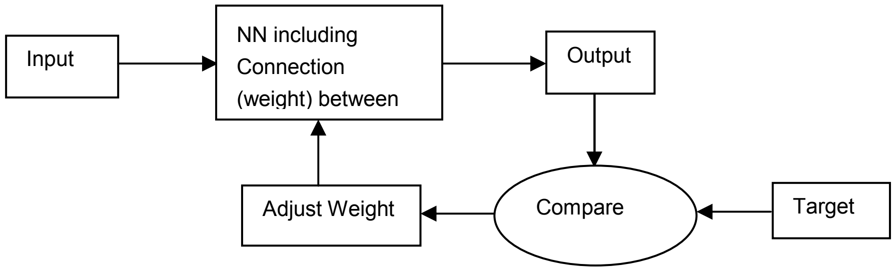

Figure1: Adjusting network based on a comparison of the output and the target

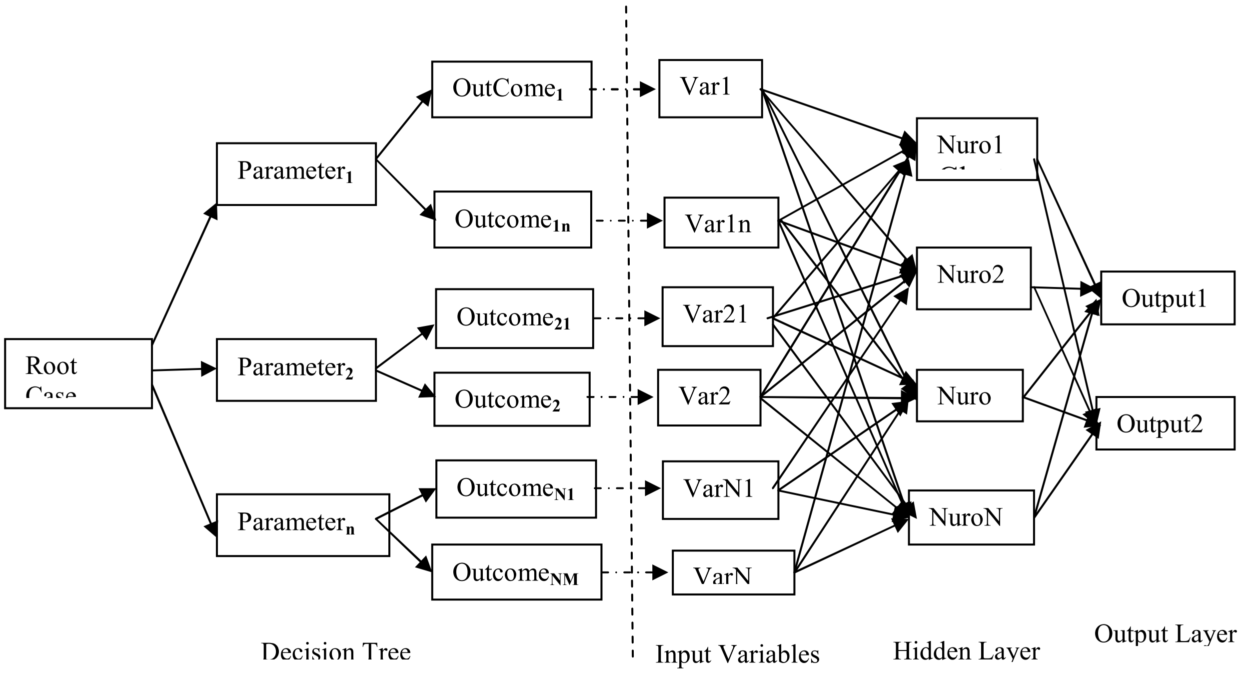

Figure2: Decision Tree Neuro-Based Architecture

In figure2 the root case leads to parameters as children of the decision tree. The number of parameters are based on the features identified in the case to be resolved by the decision tree Neuro-Based model. Once the parameters are identified and determined as the child nodes, their possible outcomes become the next level of the tree children. The number of outcomes varies based on the parameters; hence the tree may not necessarily be a binary tree. Once the outcomes are determined they form the bases for input variable into the neural net on the right hand side of the system. The variables are then transformed to the first and second layers where possible and depending on the conditions from the decision tree outcomes. The output can then be generated based on this possible handling of variables from the neural net. The decision tree neuro-based model is clearly divided into two and has an interlinking interface that joined them to make it a single model.

3.3 Data for the System

To carry out this study a random selection was made in a universe of clients of a bank in Nigeria, 1000 credit contracts, 500 considered as good and 500 considered as bad, dated from March 2009 to April 2011. All these contracts had already matured, that is to say the sample was collected after the due date of the last installment of all contracts. This is an historical data-base with monthly information on the utilization of the product. Based upon this structure, the progress of the contract could be accompanied and particularized when the client did not pay one or more installments.

Of this data set, 500 cases were used in the training and 500 were used in the testing. Both training and testing data sets contained half-good applications and half-bad applications. There are 12 influential variables over the loan decision. The definition and recoding of the variables are given in Table 3.1. On the other hand; the output for the neural network was 1 for good applications or 0 for bad applications.

Table1. Loan Granting Decision Factor

Variable code	Variable description	Variable Explanation
X_1	Age	1 if the applicant age is accepted, 0 otherwise
X_2	Income	0 if income < ₦ 40,000, 1 for (₦40,000≤income≤₦ 100,000) and 2 for (income≥₦100,000).
X_3	Job Experience	0 if (exp < 6 months), 1 for (6 months≤exp< 2years) and 2 (exp ≥ 2years)
X_4	Account type	1 for payroll account, 2 for self account
X_5	Nationality	1 for Nigerian, 0 otherwise
X_6	Residency	1 resident in Nigeria, 0 otherwise
X_7	Loan Size	1 if (100,000≤ LS≤350,000) if X_4=1, 0 otherwise Or 1 if (100,000≤ LS≤250,000) if X_4=0, 0 otherwise
X_8	Place of work	1 of company is accredited by the bank, 0 otherwise
X_9	Guarantor	1 if applicant has guarantor, 0 otherwise
X_{10}	Collateral	1 if collateral exist, 0 otherwise
X_{11}	Debt balance ratio	1 for good DBR and 0 otherwise
X_{12}	Social security	1 if applicant has social security, 0 therwise

Table1 Eye Test Decision Tree Input Data

Symptom	Description	Output
1	Recent negative examination	Yes or no
2	Eye pressure normal	Yes or no
3	Optic nerve shape normal	Yes or no
4	Iris angle open	Yes or no
5	Iris angle closed	Yes or no
6	Cornea thickness normal	Yes or no
7	Recent hazy/blurred vision	Yes or no
8	Seeing rainbow colored circles around bright lights	Yes or no
9	Severe eye pain	Yes or no
10	Severe head pain	Yes or no
11	Nausea/vomiting	Yes or no
12	Sudden sight loss	Yes or no

3.4 Decision Tree Modeling

The operation of DTs are based on the ID3 or C4.5 divide-and-conquer algorithms[25] and search heuristics which make the clusters at the node gradually purer by progressively reducing disorder in the original data set. The algorithms place the attribute that has the most predictive power at the top node of the tree and they have to find the optimum number of splits and determine where to partition the data to maximize the information gain. The fewer the splits, the more explainable the output is as there are less rules to understand. Selecting the best split is based on the degree of impurity of the child nodes. For example, a node which contains only cases of class good_loan or class *bad_loan* has the smallest disorder = 0. Similarly, a node that contains an equal number of cases of class *good_loan* and class *bad_loan* has the highest disorder = 1. Disorder is measured by the well established concept of entropy and information gain which we formally introduce below.

Given a collection *S*, containing the positive (*good_loan*) and negative examples (*bad_loan*) of

some target concept, the entropy of *S* relative to this Boolean classification is

$$Entropy(s) = -p_{good\_loan}\log_2(p_{good\_loan}) - p_{bad\_loan}\log_2(p_{bad\_loan}) \qquad (1)$$

where p_{good_loan} is the proportion of positive examples in *S* and p_{bad_loan} is the proportion of negative examples in *S*. If the output variable takes on *k* different values, then the entropy of *S* relative to this *k*-wise classification is defined as

$$Entropy(S) = \sum_{i=1}^{k} -p_i \log_2(p_i) \qquad (2)$$

Hence we see that both when the category is nearly - or completely - empty, or when the category nearly contains - or completely contains - all the examples, the score for the category gets close to zero, which models what we wanted it to. Note that *0*ln(0)* is taken to be zero by convention.

Thus, if disorder is measured by entropy, the problem of trying to determine the best attribute to choose for a particular node in a tree can be obtained by following measure that calculates a numerical value for a given attribute, A, with respect to a set of examples, S. Note that the values of attribute A will range over a set of possibilities which we call Values(A), and that, for a particular value from that set, v, we write S_v for the set of examples which have value v for attribute A.

In tree-growing, the heuristic plays a critical role in determining both classification performance and computational cost. Most modern decision-tree learning algorithms adopt a (im)purity-based heuristic, which essentially measures the purity of the resulting subsets after applying the splitting attribute to partition the training data. Information gain, defined as follows, is widely used as a standard heuristic.

$$IG(S,X) = Entropy(S) - \sum_x \frac{|S_x|}{|S|} Entropy(S_x) \quad (3)$$

where S is a set of training instances, X is an attribute and x is its value, Sx is a subset of S consisting of the instances with X = x, and Entropy(S) is defined as

$$Entropy(S) = -\sum_{i=1}^{|C|} P_s(C_i) \log P_s(C_i) \quad (4)$$

where $P_s(C_i)$ is estimated by the percentage of instances belonging to (C_i) in S, and $|C|$ is the number of classes. $Entropy(S_x)$ is similar.

3.4 Neural Networks Modeling

Artificial Neural networks learn by training on past experience using an algorithm which modifies the interconnection weight links as directed by a learning objective for a particular application. A *neuron* is a single processing unit which computes the weighted sum of its inputs. The output of the network relies on cooperation of the individual neurons. The learnt knowledge is distributed over the trained networks weights. Neural networks are characterized into feedforward and recurrent neural networks. Neural networks are capable of performing tasks that include pattern classification, function approximation, prediction or forecasting, clustering or categorization, time series prediction, optimization, and control. Feedforward networks contain an input layer, one or many hidden layers and an output layer. Equation (5) shows the dynamics of a feedforward network.

$$S^l_j = g_i\left(\sum_{i=1}^{m} S_i^{l-1} W_{ji}^l - \theta_j^l\right) \quad (5)$$

where S^l_j is the output of the neuron j in layer l, S_i^{l-1} is the output of neuron j in layer $l-1$ (containing m neurons) and W_{ji}^l the weight associated with that connection with j. θ_j^l is the internal threshold/bias of the neuron and g_i is the sigmoidal discriminant function

Backpropagation is the most widely applied learning algorithm for neural networks. It learns the weights for a multilayer network, given a network with a fixed set of weights and interconnections. Backpropagation employs gradient descent to minimize the squared error between the networks output values and desired values for those outputs. The goal of gradient descent learning is to minimize the sum of squared errors by propagating error signals backward through the network architecture upon the presentation of training samples from the training set. These error signals are used to calculate the *weight* updates which represent the knowledge learnt in the network. The performance of backpropagation can be improved by adding a momentum term and training multiple networks with the same data but different small random initializations prior to training. In gradient descent search for a solution, the network searches through a weight space of errors. A limitation of gradient descent is that it may get trapped in a local minimum easily. This may prove costly in terms for network training and generalization performance. In the past, research has been done to improve the training performance of neural networks which has significance on its generalization. Symbolic or expert knowledge is inserted into neural networks prior to training for better training and generalization performance as demonstrated in [26]. The generalization ability of neural networks is an important measure of its performance as it indicates the accuracy of the trained network when presented with data not present in the training set. A poor choice of the network architecture i.e. the number of neurons in the hidden layer will result in poor generalization even with optimal values of its weights after training. Until recently neural networks were viewed as black boxes because they could not explain the knowledge learnt in the training process. The extraction of rules from neural networks shows how they arrived to a particular solution after training.

The backpropagation algorithm with supervised learning was used, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea is to reduce this error, until the ANN learns the training data.

The training begins with random weights, and the goal is to adjust them so that the error will be minimal. The activation function of the artificial neurons in ANNs implementing the backpropagation algorithm is given as follows[27] in equation6,7,8,9.

$$A_j(\bar{x},\bar{w}) = \sum_{i=0}^{n} x_i . w_{ji} \quad (6)$$

$$O_j(\bar{x},\bar{w}) = \frac{1}{\left[1+e^{A_j(\bar{x},\bar{w})}\right]} \quad (7)$$

$$E_j(\bar{x},\bar{w},d) = \sum\left(O_j(\bar{x},\bar{w}) - d_j\right)^2 \quad (8)$$

$$\Delta w_{ji} = \eta \left(\frac{\partial E}{\partial w_{ji}}\right) \quad (9)$$

Where: x_i are the inputs, w_{ji} are the weights, $O_j(x,w)$ are the actual outputs, d_j are the expected outputs and η - learning rate.

4.0 Testing and Results

Several data files for both processing modes were generated and the network's functionality was tested extensively for various sized training and testing files. The results indicated that the system was indeed able to perform both loan application processing and eye test diagnosis as predicted. As an example, two 500 vector data files were generated that represent 500 loan applicants and 500 medical patients. The data for these cases is given in loan1tran.mat and eye1trn.mat. An additional 500 vectors was generated for testing both modes. This data is given in loan1tst.mat and eye1tst.mat.

Both train and test data files were processed through the tree module to generate 500 vector long input train and test files for the ANN inference engine of module netinf. This data is given in loan2trn.mat, loan2tst.mat, eye2trn.mat and eye2tst.mat. NOTE: 2 corresponds to an input file for the netinf module and 1 corresponds to an input file for the tree module. This naming convention does not have to be adopted; however, it is advisable to utilize some scheme that will minimize confusion when processing the modules. The netinf module creates trains and tests the ANNs. There are several network parameters that utilize default values; however, these may be edited to achieve more favorable results. The results obtained; however, are promising. For example, ANNs were built, loan500.mat and eye500.mat that were able to achieve the desired rms error rate of 0.01 for loan processing and eye test diagnosis, respectively. This is shown in Figs. 3 and 4, below.

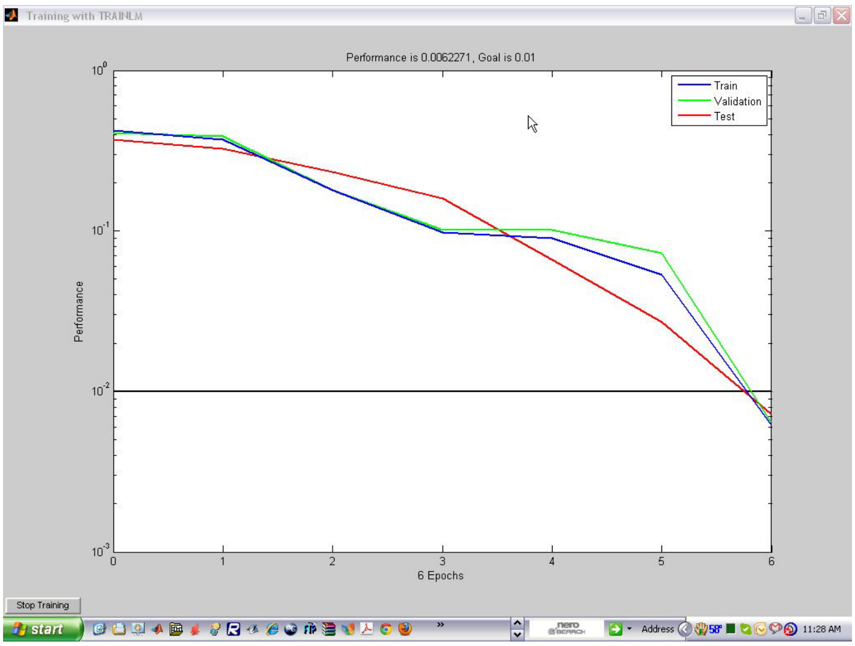

Figure 3: Training Curve for loan500

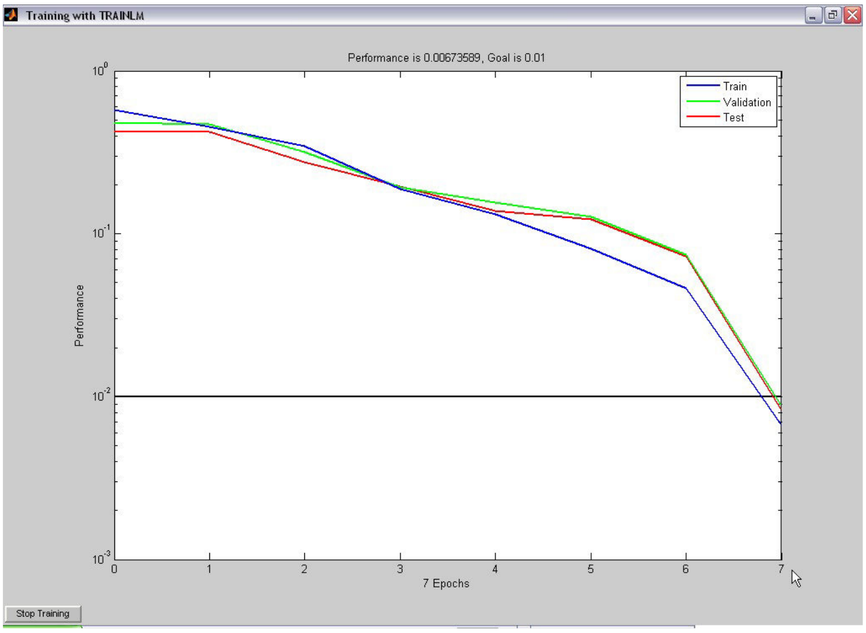

Figure 4: Training Curve for eye500

These networks were tested on the corresponding 500 vector test sets and the results are below in figure 5 and 6. respectively.

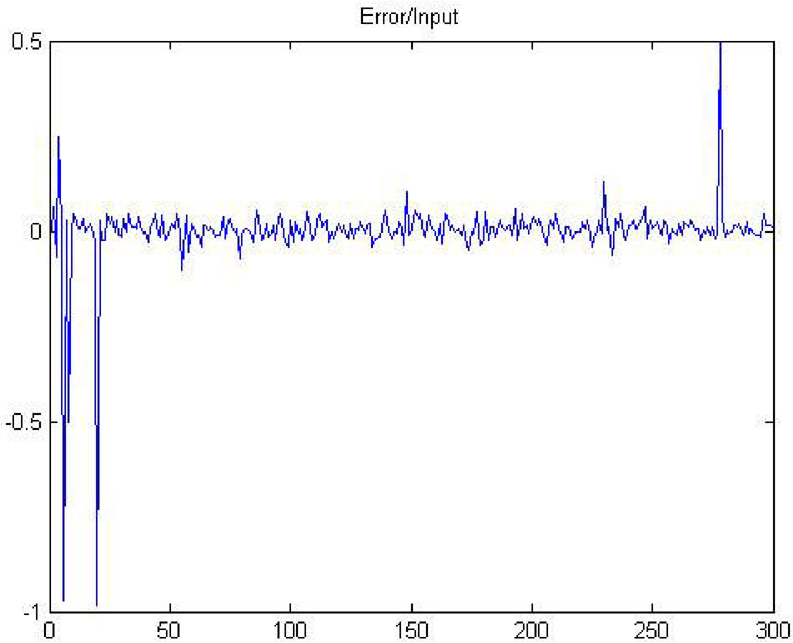

Figure 5: Error Curve for loan500

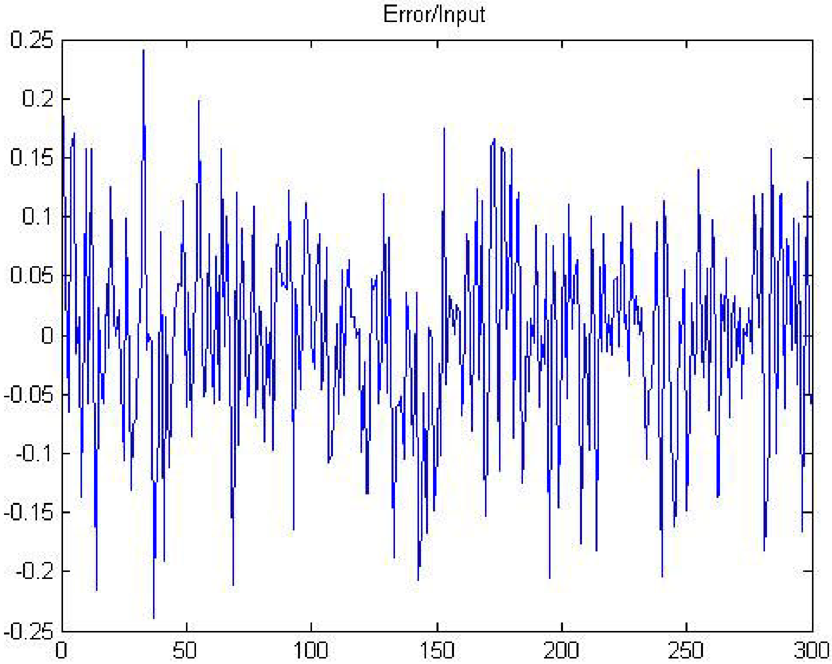

Figure 6: Error Curve for eye500

As the criterion for selection/denial and disease presence/absence was linearly separable, represented by a threshold of 0.5, these results clearly indicate the success of the LDSS in solving the problems for the given data.

4.1 Performance of the system using Random Tests

In testing the system randomly, forty different tests from the data sets for testing the system were carried out at random and the result compared with the actual data in file collected from the bank. This data set was used for the Decision Tree, Neural Networks and LDSS. Table3 and Table4 for Bank A and Bank B respectively, show the result indicating TRUE POSITIVE(TP) where system suggested grant loan and the actual grand loan, TRUE NEGATIVE(TN) where the system suggested don't grant and the actual is don't grant, FALSE POSITIVE(FP) where the system suggested grant loan but actual is don't grant, FALSE NEGATIVE(FN) where the system suggested don't grant but actual is grant loan.

Table3: Result from Bank A

Decision Tree			Neural Networks			LDSS		
TP = 5	FP = 1	Total = 6	TP = 7	FP = 1	Total = 8	TP = 9	FP = 1	Total = 10
FN = 5	TN = 9	Total = 14	FN = 3	TN = 9	Total = 12	FN = 1	TN = 9	Total = 10
Total = 10	Total = 10	Total = 20	Total = 10	Total = 10	Total = 20	Total = 10	Total = 10	Total = 20

Table4: Result from Bank B

Decision Tree			Neural Networks			LDSS		
TP = 7	FP = 4	Total = 11	TP = 8	FP = 4	Total = 12	TP = 9	FP = 2	Total = 11
FN = 3	TN = 6	Total = 9	FN = 2	TN = 6	Total = 6	FN = 1	TN = 8	Total = 9
Total = 10	Total = 10	Total = 20	Total = 10	Total = 10	Total = 20	Total = 10	Total = 10	Total = 20

$$Accuracy = \frac{TP + TN}{Total\ Number\ of\ dta\ used} \ldots\ldots\ldots\ldots\ldots\ldots\ldots.(10)$$

For Decision Tree, $accuracy = \frac{5+9+7+6}{40} = 0.68$, for Neural Networks, $accuracy = \frac{7+9+8+6}{40} = 0.75$ and for LDSS, $accuracy = \frac{9+9+9+8}{40} = 0.88$.

4.2 Performance Comparison with decision Tree alone and Neural Networks alone.

The system was developed in a way that different cases can be executed using Decision(DT) alone, Neural

Networks(NN) alone and LDSS(the Decision Tree-Neuro Based System). The performance of the Decision Tree-Neuro based system was compare with the performance of Neural Networks and Decision Tree alone using the receiver operating characteristics (ROC) curve from Table2 and Table3 using MATLAB software package(MATLABR2009b). The result is as shown in figure7.

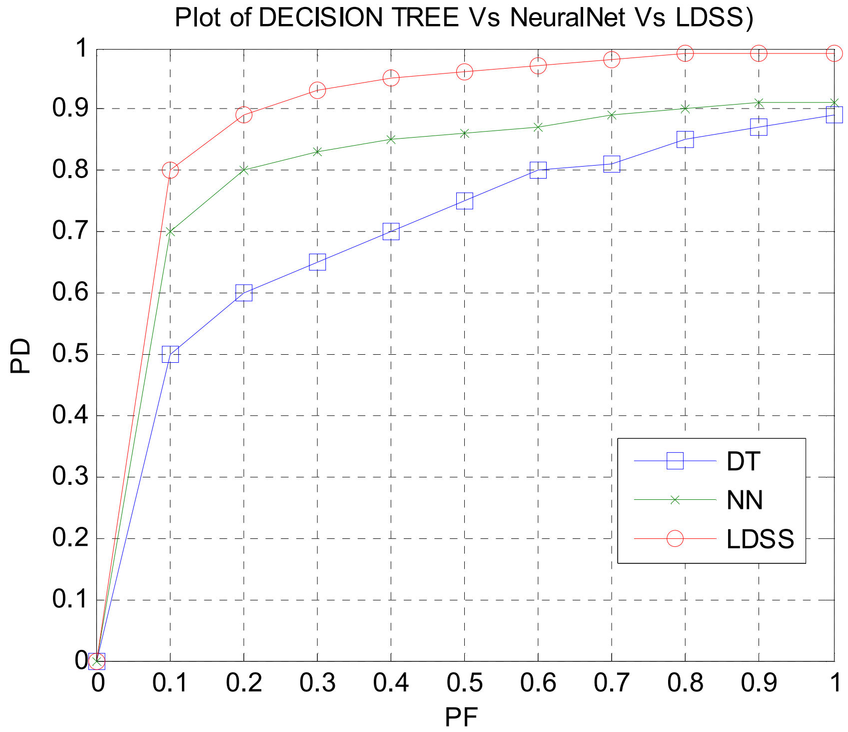

Figure 7: Comparison of LDSS with NN and DT

Explanations: From Figure 7, the ROC curve for LDSS was able to correctly classify over 80% of customers as bad or good without causing false alarms. This is followed by the ROC curve for neural networks with 70% classification with no false alarms. The least is the ROC curve for decision tree with 50% detection. Figure 7 again shows that the performance of LDSS is in agreement with other classification software, but performs better. This relative advantage of LDSS over others is as a result of combining decision tree and neural networks.

5.0 Conclusion

Decision making, particularly in a complex and dynamically changing environment of the present day is a difficult task that requires new techniques of computational intelligence for building adaptive, hybrid intelligent decision support systems. The research work thus proposed Decision Tree- Neuro based Decision Support System with 88% accuracy for decision support. The developed system can provide explanation why a particular customer was selected knowing that a customer may protest his rejection.

The works thus contribute the following to Knowledge:-

1. Design a Decision Tree-Neuro Based architectural topology to implement a decision support system.
1. Design of a Decision Tree-Neuro Based model that can adequately decide if customers applying for loan should be granted or not.
2. By changing variables in the system can be used to as clinical decision support system to diagnosis eye diseases.

6.0 Recommendations

The contribution in this work is recommended:-

1. To financial institutions, particularly in Nigeria for loan granting applications. Given the fact that humans are not good at evaluating loan applications together with the fact that Nigeria is full of nepotism, tribalism and corruption, necessitates the need for robust knowledge tool using Decision Tree – Neuro based Decision support system to assist banks in credit risk evaluation for the sustainability of the banks and Nigerian economy bearing in mind that the loan officer may be asked for explanations why certain applicants are chosen in preference to others.

2. To medical experts (ophthalmologists) as an aid in the decision making process and confirmation of suspected cases. Also, a non expert will still find the work useful in areas where prompt and swift actions are required for the diagnosis of a given eye disease covered in the system. Medical practitioners who operate in areas where there are no specialist (ophthalmologist) can also rely on the system for assistance.

References

[1] Gallinari, P. (1998):Predictive models for sequence modelling, application to speech and character recognition.Adaptive Processing of Sequences and Data Structures, Lecture Notes in Computer Science Volume 1387, 1998, pp 418-434

[2] Gevaert, W., Senov, G. and Mladenov, V.(2010). Neural Networks for Speech Recognition. Journal of Automatic Control, University of Belgrade, vol. 20:1-7

[3] Matan, O., Kiang, R. K., Stenard, C. E., Boser, B.,Denkar, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. and Le Cun, Y.(1990). Hand Written Character Recognition Using Neural Networks Architectures. Proceeding of the 4th USPS Advanced Technology Conference, Washington D.C., Pp 1003-1011.

[4] Ralston, A. and Reilly, E. D.(2000). Encyclopedia of Computer Science, London : Nature Pub. Group. Ref. QA76.15 .E48 2000.

[5] Glorfeld, L.W. & Hardgrave, B.C. (1996): An improved method for developing neural networks: The case of evaluating commercial loan creditworthiness. Computer Operation Research, 23 (10), Pp: 933-944.

[6] Handzic, M. & Aurum, A. (2001). Knowledge discovery: Some empirical evidence and directions for future research. Proceedings of the 5th. International Conference on Wirtschafts Informatics (WI'2001), 19-21 September, Augsburg, Ge rmany.

[7] Bigus, J.P. (1996): Data mining with neural networks: Solving business problems from application development to decision support. McGraw Hill, USA.

[8] Marakas, G.M. (1999): Decision support systems in the twenty-first century. Prentice Hall, New Jersey, USA.

[9] Smith, K.A. (1999): Introduction to neural networks and data mining for business applications. Eruditions Publishing, Australia.

[10] Thomas, L.C. (1998): A survey of credit and behavioural scoring: Forecsting financial risk of lending to customers. Retrieved June 5, 2002, from www.bus.ed.ac.uk/working_papers/full_text/crc9902.pdf

[11] Steenackers A.,and Goovaerts M.J.(1998): A credit scoring model for personal loans. Insurance Mathematics & Economics, 8, 31-34.

[12] Henley W.E., Hand D.E.(1997). Construction of a k-nearest neighbor credit-scoring system. IMA Journal of Mana-gement Mathematics, 8, 305-321.

[13] Davis, R. H., Edelman, D. B., and Gammerman, A. J.(1992):Machine Learning Algorithms for Credit-Card Applications,. IMA Journal of Mathematics Applied in Business and Industry (4), Pp: 43-51.

[14] Desai V.S., Crook J.N., Overstreet G.A. Jr(1996): On comparison of neural networks and linear scoring models in the credit union environment. European Journal of Operational Research, 95(1), Pp: 24-37.

[15] Baesens B., Setiono R., Mues C, Vanthienen J**.**(2003): Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation, Management Science, Volume 49 , Issue 3, March 2003, Pp:312 – 329.

[16] West, D.(2000): Neural Network Credit Scoring Models, Computers & Operations Research, vol. 27, no. 11-12, pp. 1131-1152.

[17] Chi G., Hao J., Xiu Ch., Zhu Z. (2001): Cluster Analysis for Weight of Credit Risk Evaluation Index. Systems Engineering-Theory Methodology, Applications, 10(1), Pp. 64-67.

[18] Lundy M.(1993): Cluster Analysis in Credit Scoring. Credit Scoring and Credit Control. New York: Oxford Uni-versity Press.

[19] Luo Y.Z., Pang S.L., S.(2003): Fuzzy Cluster in Credit Scoring. Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003, 2731-2736.

[20] Baesens, B., Van, T., Gestel, M. Stepanova et al.,(2005):Neural network Survival Analysis for Personal Loan Data, Journal of the Operational Research Society, vol. 56, no. 9, Pp. 1089- 1098, Sep.

[21] Bakpo, F. S. and Kabari, L. G. (2011):Diagnosing Skin Diseases using an Artificial Neural Network, Artificial Neural Networks - Methodological Advances and Biomedical Applications, kenji suzuki (ed.), isbn: 978-953-307-243-2, intech, Available from: http://www.intechopen.com/articles/show/title/diagnosing-skin-diseases-using-an-artificial-neural-network

[22] Kabari, L. G and Nwachukwu, E. O. (2012): Neural Networks and Decision Trees For Eye Diseases Diagnosis, Advances in Expert Systems, Petrica Vizureanu (Ed.), ISBN: 978-953-51-0888-7, InTech, Available from: http://www.intechopen.com/books/advances-in-expert-systems/neural-networks-and-decision-trees-for-eye-diseases-diagnosis

[23] Kumar, D. S., Sathyadevi, G. and Sivanesh, S.(2011): Decision Support System for Medical Diagnosis Using Data Mining, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1,ISSN (Online): 1694-0814,www.IJCSI.org.

[24] UKessay.com(2013):Advantages and Limitation of Neural Networks. Retrieved from http://www.ukeassay.com/essays/education on 11/06/2013.

[25] Quinlan, J. R.(1987): Simplifying Decision Trees, International Journal of Man-Machine Studies, vol. 27, Pp.221-234, 1987.

[26] Chandra R., and Omlin C. W.(2007): Knowledge Discovery Using Artificial Neural Networks For A Conservation Biology Domain, Proceedings of the 2007 International Conference on Data Mining, Las Vegas, USA, In Press, 2007.

[27] Haykin, S. (1999): Neural Networks: a Comprehensive Foundation. Second Edition. Prentice Hall.