Credit Risk Evaluating System Using Decision Tree – Neuro Based Model

L. G. Kabari
Computer Science Department, Rivers State Polytechnic, Bori PMB 20,Rivers State, Nigeria.

E. O. Nwachukwu
Computer Science Department, University of Port Harcourt, Port Harcourt, PMB 5323, Nigeria.

Abstract

Decision making in a complex and dynamically changing environment of the present day demands a new techniques of computational intelligence for building an equally adaptive, hybrid intelligent decision support system. In this paper, a Decision Tree-Neuro Based model was developed to handle loan granting decision support system. 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 were considered based on the guidelines of different banks in Nigeria, to validate the system. 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 result indicates that Decision Tree-Neuro Based Models with its 88% success rate and good explanatory background are successful technology that can adapt to the present day loan application evaluation in commercial banks.

Key words: *Backprpagation algorithm, Decision Tree-Neuro Based Model, MATLAB engine, Object-Oriented Analysis and Design, opacity.*

1. Introduction

Recently, with the financial crisis becoming serious, the trend of financial globalization and financial market volatility has attracted people's attention, especially banks and investors who suffered unprecedented challenges of credit risk. The credit crisis caused by American showed that international banking has been challenged because of their lack of effective methods for assessment in controlling credit risk[1]

Credit scoring is a vital activity conducted by financial institutes which involves a discrete decision making process in which the loan is assessed to avoid financial risk. However this process is often biased as it involves a degree of personal preference due to the unavailability of suitable decision making models[2].

The advancement of neural networks has been seen as one of the most exciting developments in terms of their applicability to business settings. Decision trees are also an efficient tool in evaluating options hence a hybrid of the two will be a motivating area of research making us to embark on the research to develop a model that is capable of blending the two models. The need to develop a system that can reduce loan defaulting in financial establishment is also a major motivating factor towards the research on ways of solving the problem by providing a hybrid model that can attend to the challenge.

2. Literature Review

Decisions concerning credits granting are one of the most crucial in an every banks' policy. Well-allocated credits may become one of the biggest sources of profits for any financial organizations. On the other hand, this kind of bank's activity is connected with high risk as big amount of bad decisions may even cause bankruptcy. The key problem consists of distinguishing good (that surely repay) and bad (that likely default) credit applicants.

The main investigations, in this area, are based on building credit risk evaluation models, allowing for automating or at least supporting credit granting decisions. Numerous methods for evaluating credit risk have been developed. Most of them are based on traditional statistical methods like logistic regression [3], K-nearest neigh-bor[4], classification trees[5] or neural network models[6][7][8], as well as cluster analysis[9][10][11].

Some of authors combined different models, to obtain strong general rules. In [12], authors built the decision system supporting evaluation of business credit applications, by applying integration of case based reasoning and decision rules. Such an approach

allowed for connecting two kinds of representation knowledge and for formulating rules for a set of typical examples.

3. Methodology

In this paper, an integration of Decision Tress and Neural Networks is used to form a hybrid called Decision Tree - Neuro Based Credit Risk Evaluation System (DTNBCRES).

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 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[13]. 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[14].

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 figure2 (see appendix). 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 (see appendix).

In figure2 the Bank Rule leads to parameters used for granting of the loan which are in this case the children of the Bank Rule. The numbers 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.

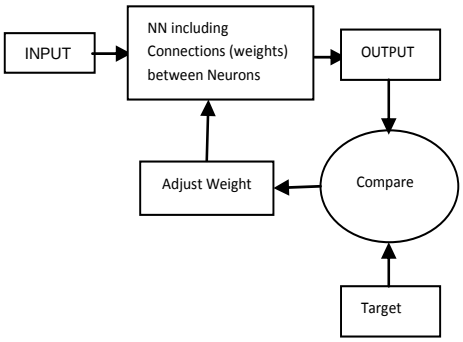

Figure1: Adjusting network based on a comparison of the output and the target

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 2010 to April 2012. All these contracts had already matured, that is to say the sample was collected after the due date of the last instalment 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 instalments.

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 13 influential variables over the loan decision. The definition and recording of the variables are given in Table1 (see appendix). On the other hand; the output for the neural network was 1 for good applications or 0 for bad applications.

3.4. Decision Tree Modelling

The operation of DTs are based on the ID3 or C4.5 divide-and-conquer algorithms[15] 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 (GL) and negative examples (BL) of some target concept, the entropy of *S* relative to this Boolean classification is

$$Entropy(s) = -P_{GL}\log_2(P_{GL}) - P_{BL}\log_2(P_{BL}) \dots (1)$$

Where P_{GL} is the proportion of positive examples in *S* and P_{BL} 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

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

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.

$$IG(S,X) = Entropy(S) - \sum_{x} \frac{|S_x|}{|S|} Entropy(S_x) \dots\dots\dots\dots\dots (3)$$

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

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

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 Modelling

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.

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

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.

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[16] in equation6,7,8,9.

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

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

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

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

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. 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 as predicted. As an example, two 500 vector data files were generated that represent 500 loan applicants. The data for these cases is given in loan1tran.mat. An additional 500 vectors was generated for testing both modes. This data is given in loan1tst.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.

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 was built, loan500.mat that was able to achieve the desired rms error rate of 0.01 for loan processing. This is shown in Figure3 below.

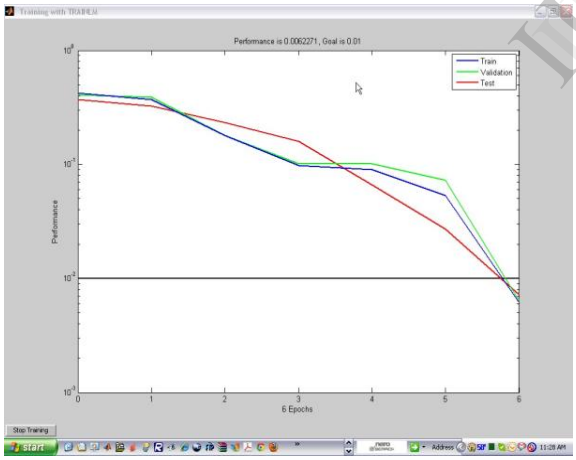

Figure 3: Training Curve for loan500

These networks were tested on the corresponding 500 vector test sets and the result is below in figure4. As the criterion for selection/denial was linearly separable, represented by a threshold of 0.5, these results clearly indicate the success of the DTNBCRES in solving the problems for the given data.

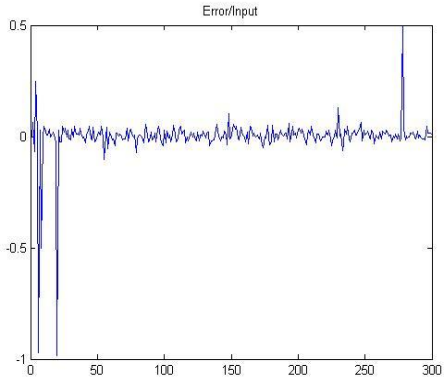

Figure4: Error Curve for loan500

4.1. Performance of the System Using Random Test

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 DTNB. Table2 and Table3 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. The result gives an accuracy rate of 88% for Decision Tree-Neuro Based (DTNB), 75% for Neural Networks(NN) and 68% for Decision Tree(DT).

4.2. Performance Comparison with Decision Tree and Neural Networks

The system was developed in a way that different cases can be executed using Decision (DT) alone, Neural Networks(NN) alone and DTNBCRES(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 figure5.

Explanations: From Figure5, the ROC curve for DTNB 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.

Figure5 again shows that the performance of DTNB is in agreement with other classification software, but performs better. This relative advantage of DTNB over others is as a result of combining decision tree and neural networks.

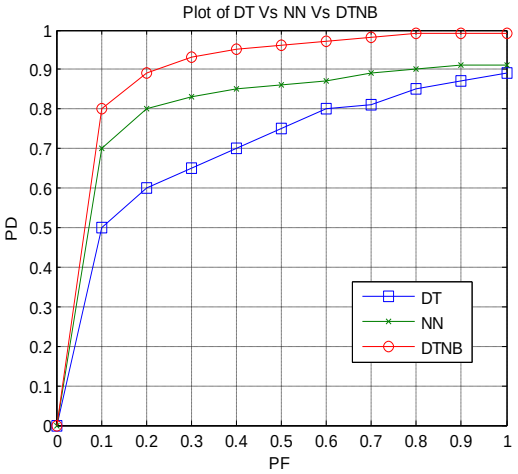

Figure5: Comparison of DTNB with NN and DT

5. Conclusion

Granting of loans by financial institutions (banks or home loan business) particularly in a complex and dynamically changing environment of the present day requires new techniques of computational intelligence for building adaptive, hybrid intelligent credit risk evaluation system. The research work thus proposed Decision Tree - Neuro Based Credit Risk Evaluation System with 88% accuracy for decision support. The developed system can provide explanation why a particular customer was rejected or granted knowing that a customer may protest his/her rejection.

This proposal in a way attempt to solve the present global credit crisis which the international banking has been challenged because of their lack of effective methods for assessment in controlling credit risk

The works thus contribute the following to the existing body of knowledge:-

Design a Decision Tree-Neuro Based architectural topology to implement credit risk evaluation system.

Developed a Decision Tree-Neuro Based model with 88% accuracy that can adequately decide if customers applying for loan should be granted or not.

6. Recommendation

The contribution in this work is recommended 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 a multi tribal nation which consequently often resulted to nepotism, tribalism and corruption, there 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.

References

[1] Chiu, J., Yan, Y., Xuedong, G. and Chen, R.(2010):A New Method for Estimating Bank Credit Risk. 2010 International Conference on Technologies and Application of Artificial Neural Networks, Taiwan. Pp. 503-507.

[2] Palihakkara, V. and Peiris, M.(2011): An Investigation in to Application of Decision Making Models For Credit Scoring In A State Sector Bank, Proceedings of the International Symposium on the Analytic Hierarchy Process 2011.

[3] Steenackers A., and Goovaerts M.J. (1989): A credit scoring model for personal loans. Insurance Mathematics & Economics, 8, 31-34.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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.
[12] Staefanowski J., and Wilk S(2001): Evaluating Business Credit Risk by Means of Approach – Integrating Decision Rules and Case-Based Learning. International Journal of Intelligent Systems in Accounting, Finance and Management, 10, 97-114.
[13] Kumar, K., Sundar, G. and Thakur, M.(2012):Extracting Explanation from Artificial Neural Networks, International Journal of Computer Science and Information Technologies(IJCSIT), Vol. 3 (2) , Pp: 3812-3815
[14] UKessay.com:Advantages and Limitation of Neural Networks. Retrieved on 11/06/2013 from: http://www.ukeassay.com/essays/education
[15] Quinlan, J. R.(1987): Simplifying Decision Trees, International Journal of Man-Machine Studies, vol. 27, Pp.221-234, 1987.
[16] Haykin, S. (1999). Neural Networks: a Comprehensive Foundation. Second Edition. Prentice Hall.

Appendix

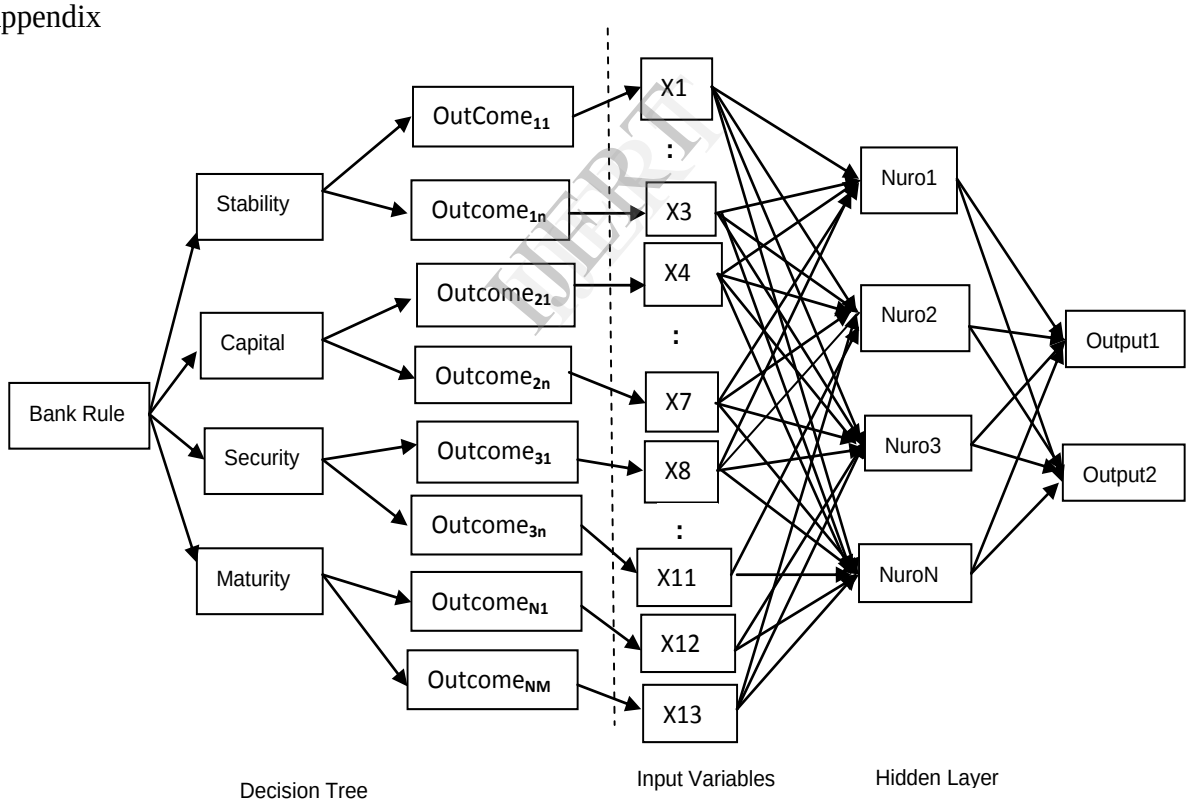

Figure2: Decision Tree Neuro-Based Architecture

Table1. Loan Granting Decision Factor

Parameter	Variable code	Variable description	Variable Explanation
Stability	X_1	Accommodation	1 if applicant has his resident, 0 if in rented place
	X_2	Job Experience	0 if (exp < 2 years), 1 for (2 years ≤exp < 8 years) and 2 (exp ≥ 8years)
	X_3	Place of work	1 of company is accredited by the bank, 0 otherwise
Capital	X_4	Loan Size	1 if (100,000≤ LS≤350,000) and X5=2, 0 otherwise Or 1 if (100,000≤ LS≤250,000) and X_4=1, 0 otherwise
	X_5	Account type	2 for payroll account, 1 for self account, 0 otherwise
	X_6	Debt balance ratio	1 for good DBR and 0 otherwise
	X_7	Income	0 if income < ₦ 40,000, 1 for (₦40,000<income< ₦ 100,000) , 2 if N100,000<income<250,000, 3 for (income≥ ₦2500,000).
Security	X_8	Residency	1 resident in Nigeria, 0 otherwise
	X_9	Guarantor	1 if applicant has guarantor, 0 otherwise
	X_{10}	Collateral	1 if collateral exist, 0 otherwise
	X_{11}	Social security	1 if applicant has social security, 0 otherwise
Maturity	X_{12}	Age	0 if age < 20, 2 if 20<age<35, 3 if 35<age<45, 1otherwise
	X_{13}	Marital Status	1 if married, 0 otherwise

Table2: Result from Bank A

Decision Tree			Neural Networks			DTNB		
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

Table3: Result from Bank B

Decision Tree			Neural Networks			DTNB		
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

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

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 DTNB, $accuracy = \frac{9+9+9+8}{40} = 0.88$.