International Journal of Information Science 2013, 3(3): 63-69
DOI: 10.5923/j.ijis.20130303.03

A Neural Network Approach to Selection of Candidates for Electoral Offices by Political Parties

A. E. Akinwonmi*, B. M. Kuboye, A. F. Thompson

Department of Computer Science,The Federal University of Technology, Akure. P.M.B. 704

Abstract Useful governance comes from a reliable electoral process. Such begins from candidate selection within political parties. In Nigeria, factors such as electoral system, party ideology, political culture, and the organization of government, overtime had influenced candidates' selection processes for the conduct of political party primaries invariably make it subjective. Thus candidates who make it through such process are not always the party's best. Invariably, many of these candidates loose the general elections. Obviously, advancement in technology has gradually changed nearly every facet of live yet there exist exceptions in the area of democratizing the electoral system. In this paper, we present a feed forward/back propagation neural network based approach to selecting suitable candidates for elective positions before general elections. We had to enlarge the training dataset through interpolative synthesis. Our result proved that the approach was efficacious. In this report, we present an introduction to the problem and the methodology of solving the problem. Thereafter, we reported on the design, implementation and the results obtained from the system. We drew useful conclusions were later drawn from the results.

Keywords Feed Forward, Back Propagation, Feed Forward Neural Network, Candidate Rating, Candidate Selection

1. Introduction

Good governance is an upshot of a qualitative electoral process: beginning from candidate selection or nomination within political parties, permeating candidate screening by national electoral commission through to the actual contest in general elections. Factors such as electoral system, party ideology, political culture and the organization of government, overtime, had influenced candidates' selection processes[1]. In some countries gerrymandeering is used to favour certain candidates.

In the United States of America, the exact mode of which primaries or caucuses are conducted and the voting procedures for delegates is defined by each state's local office of political parties[2]. Education, career experiences and exposure of candidates can facilitate or undermine a candidate's success in the polls.Many Americans believe that Sarah Palin (ex Republican Vice Presidential candidate) was instrumental to McCain loss of election to Obama in the election that led to the latter's victory in 2009. The pre-election debates revealed Palin as a "limited to career" person, hence she went off topic numerous of times during the debates. This was more revealing as she was unable to mention a name of a single newspaper or magazine which she had read[3].

In Nigeria, the precise process of candidate selection within most parties is not well grasped by the citizenry including many politicians. This is more so because available literature on intra-party processes is largely inadequate. Moreover, the ideal of what parameters and criteria to use for candidate selection varies largely within many political parties while it is vague in some. In general, political parties usually present candidates for general elections after selections are made through intra-party elections, recommendations or appointments. In most cases the parties.

The survey by[4]on candidate selection criteria by parties further revealed that, education, professional experience, political experience, criminal records, origin, age and other basic demographics were used as criteria for selection in many cases. Certain selection criteria are specified in many cases. These may include gender representation in other cases[5]. In most successful elections, education, experiences and exposure of candidates have roles to play. These criteria are best described as rudimentary and they do not form the basis by which party delegates cast their votes. There has to be a more reasonable means of selecting a candidate from the ostensibly qualified lot within a political party.

In that regard, we believe that candidate selection can be made more objective and goal-driven if quantifiable, resultant but consequential parameters are included as criteria for candidate selection. On such, each candidate can

* Corresponding author:
tobapro@yahoo.com (A. E. Akinwonmi)

Published online at http://journal.sapub.org/ijis

be rated and assessed against a set benchmark that will be agreeable to the party members. It is expected that such rating will provide a model for assessing each ostensiblyqualified candidate's electoral prospect, personal understanding and alignment with parties manifestoes vis-à-vis electorates' expectations.

Unfortunately, not all parties have the internal democracy, the genuine or generally acceptable method of selecting candidates in Nigeria[6]. Experiences have shown some power manipulating bodies or individuals in the parties deliberately or surreptitiously block due process of selecting candidates (*ibid*). In most of the parties in Nigeria, bribery and sit-tight syndrome have been institutionalized as mode of candidate selection rather than objectivity and credibility (*ibid*). Unmistakably Nigeria's experience has regrettably not demonstrated an acceptable level of democratic compliance. Rather, "god-father" influence, money might and intimidations have characterized internal selection of candidates for elective posts in Nigeria.

Undoubtedly, a substantial and gradual technological lift has been witnessed in nearly every facet of our lives as a nation, there still exist exceptions in the sphere of democratization of the electoral system. Several efforts made hitherto to upgrade the traditional voting system had failed. Introduction of E-voting has consistently and cleverly been resisted even by the assembly. Sadly, the customary voting system, with all its inherent shortcomings and pitfalls has been inefficient in the overall voting time and cost. It has remained very unreliable at its best when compared with an E-voting system which is accurate, verifiable, democratic, private, and convenient and in which "polls are popping up all over the World Wide Web, with a few clicks of your mouse"[7].

The hand-marked, hand-collated and hand-appraised traditional political selection process can only get more cumbersome and unnecessarily time-sapping. Worse still, such methods are infested with human errors and vulnerable to premeditated manipulations and rigging by individuals involved in the processes. Thus the integrity of the processes is grossly jeopardized.

Fair enough, computers have proved very efficient at solving problems with direct or known solutions through conventional problem analysis and coding[8]. However when an explicit description of the solution to a problem is not feasible or where complex conventional approach to solving problems is inadequate, Artificial Neural Networks (ANNs) can be used. The network's ability to accumulate knowledge about objects and processes using learning algorithms makes their application at solving complex problems very promising and attractive ([9],[10]). Since they can learn fromexample or experience, ANNs have been used in solving decision-making problems including: classification, generalization, pattern recognition, learning, non-linearity, abstraction, and analysis of noisy or incomplete data[[11],[12],[13],[14].

Moreover, neural networks can adapt themselves during training based on examples of similar problems even without a desired solution to the problem[15]. If sufficiently and carefully trained, a neural net is able to associate the problem data to the solutions, relate inputs to outputs, and offer a feasible solution to a novel problem *(ibid)*.

In this paper we present a feed forward / backward error propagation neural network based method to implement selection of candidates for electoral positions based on benchmarked criteria.

1.1. Related Studies

Feed forward/back propagation neural networks have been used in solving character matching problems which good results[16].[17] Presents an expert system for evaluation of the unemployed at certain offered posts. The system used Neuro-Fuzzy techniques to analyze a corporate database of unemployed and enterprises profile data. Thereafter, a Sugeno type Neuro-Fuzzy inference system was used to process the matching of an unemployed with an offered job.

Moreover,[18] using a Feed Forward/BackPropagation neural network engine, designed and implemented a NN based system for recognizing musical notes.

In the area of using synthetic data,[19] presented a method for artificial interpolation of data for training purpose in classifier. The result shows that a classifier trained as such did not perform less than that trained on real data. It also proved that such use of synthetic data was safe.

2. Methodology

We studied the selection criteria and manifestoes of some political parties in the country in the past and at present. We selected ten criteria across party boundaries which were used to assess candidates.

From candidates' particulars we selected information about education, previous experience and exposure. Whereas from party manifestoes, we selected specific electoral promises including agricultural inputs, free and compulsory education, provision of portable water, provision of electricity, employment generation, good health care delivery and provision of good road networks.

We trained a feed forward/back propagation neural network with digitalized benchmarked scores as a target for comparison, targeting an output of 1. The network configuration was stored for the target candidate.

The network was then run using the digitalized scores of candidates. Output values were recorded and compared with the target. A selection order was generated from the output based on how a candidate compares with the benchmark. Output values ranges from 0 to 1. The candidate with the closestvalue to 1 is selected as the candidate for the position.

3. Design and Implementation

We applied object-oriented features of Visual Basic to enhance modular techniques in the design of basic functional components. There are four main functional modules of the Candidate selection system (Figure 1). These include the data entry module, digitalization module, the feed forward neural engine and the output module. Other auxiliary components were coded into the system to enhance data collation, accessibility and testing. All components, though integrated, are reusable and separable.

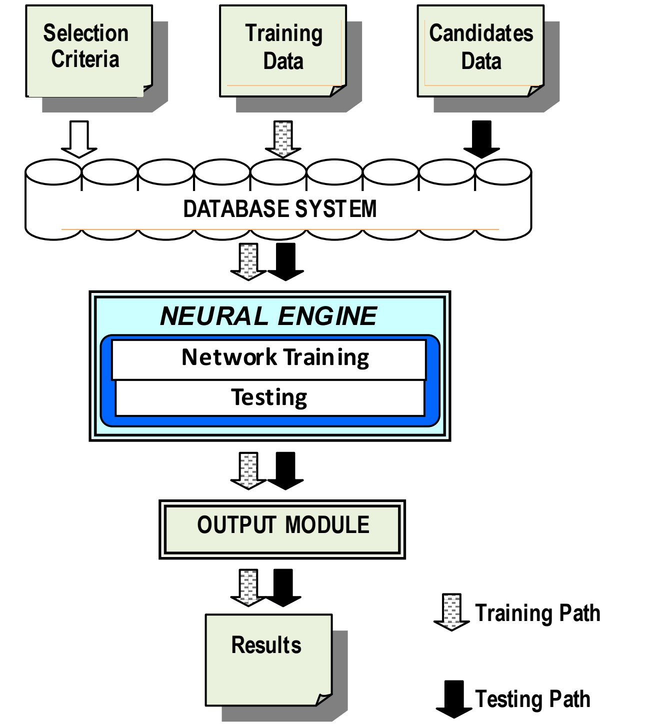

Figure 1. Conceptual diagram of the neural-network-based candidate selection software

At one end, records of selection criteria, candidates' data and benchmarked data were managed on a portable database system. Recorded scores were submitted to the digitalization module which digitizes and concatenate the binary output for each candidate. The digital scores were thereafter submitted to the neural network module as input signals for training the network or testing as applicable. The neural network module had the features to save the network configuration for trained networks for subsequent testing and to submit the result of the tested input to the output module wherein it could be displayed or printed. The conceptual diagram of the system is shown in figure 1 below.

3.1. Setting up the Network

3.1.1. Training Set

Reports from pattern recognition investigations have shown that the classifier trained on the most data are give better performance[20],[21],[22]. However, implementing such approach suffers some setbacks which include the high cost of large training sets. Moreover such dataset are difficult to obtain in many cases.

The major challenge encountered in this investigation was the scanty nature of available training datasets obtained from section 2 above. Worse still such may not be representative of the universe of discourse. In addition,[19] showed thatsuch datasets may be imbalance, as one class may be represented by too insufficient samples while others have too many.Fair enough, Nonnemaker's investigation in[19] offered a method to ease this problem through the generation of artificial (synthetic) data. The result of that work and others alsoin[22],[23] and[24] show that training on interpolated data issafe. Such training reportedly gave a good performance when tested on the pure samples. Consequently, we had to synthetically enlarge the training set for better result. Thus we obtained synthetic derivations from interpolation of the sample scores obtained from section 2 above.

A variant of the method used by Nonnemakerwas adapted and presented as follows:

i. For each selection criterion, werandomly chose 20"seed" samples of the collected training data.

ii. For each of the 20 seed points of that selection criterion we derived a training point in a convex space represented by the seeds points by randomly assigning20 weights w1, w2…w20such that w1 +w2 +…w20 = 1

iii. We multiplied each seed point s1, s2…s20by its corresponding weight w1, w2…w20 vialinearalgebra

iv. We calculated the new training point, t, by summing s1w1, s2w2,… s20w20 over all points

v. We repeated steps ii to iv to obtain each training points needed for each criterion.

vi. We repeated step i for every set of training points.

We convertedeachset of criteriato binary and concatenate as shown in table 2 of section 4. For *n* criteria for each candidate, the concatenated equivalence is inputted to the input neurons ($p_1,p_2,...p_n$) of the network and processed by the neural network in the configuration shown below:

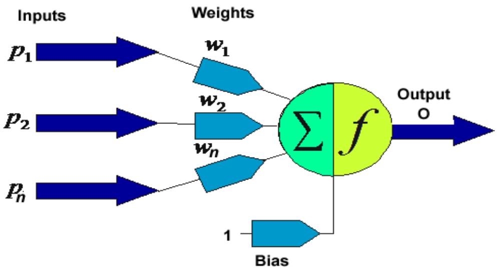

Figure 2. Conceptual neuron (adopted[25])

The output of the network*O*is given as

$$O = f(p_1w_1 + p_2w_2 ...p_nw_n + b) \quad (1)$$

Where $p_1, p_2,...p_n$are neural inputs

$w_i, w_2,...w_n$are weights on dendrites

b is bias

Each neuron within the network is a simple processing unit such that when given one or more inputs an output is produced. Every input has an associated weight modifying the strength or contribution of each input. The neuron sums all inputs and calculates an output to be passed on to the next neuron. Each neuron is also associated with an activation function which is usually a non-linear function. A sigmoid function is used in this work.

$$Y_i = 1/(1 + e)^{-X_i} \quad (2)$$

Where Y_j=Activation

[25] presented a mathematical approach to the back-propagation Algorithm. It also shows that neural units are interconnected. These connections relate to the edges of the basic directed graph. There are weights (strength) associated with each connection denoted by w_{ij} the weight of the connection from unit u_i to unit u_j. This approach allows the pattern of network connectivity to be represented by a weight matrix w whose elements are the weights w_{ij}. A positive weight represents an excited connection whereas a negative weight represents an inhibited connection. Where there are more layers of neurons, the pattern of connectivity characterizes the architecture of the network as follows:

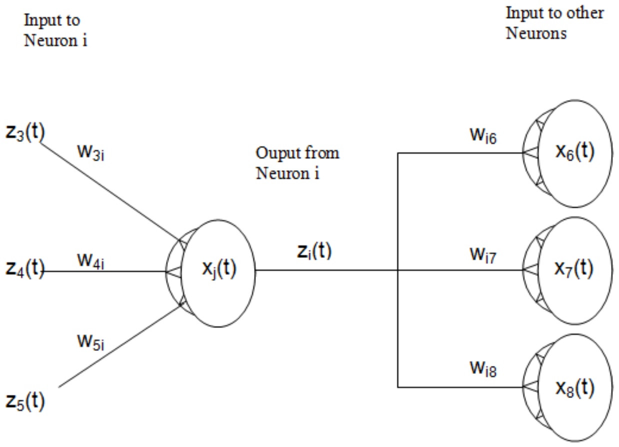

Figure 3. Conceptual diagram of interconnected neurons (adopted[25])

The total weighted input x_j:

$$x_j = \sum y_i w_{ij} \quad (3)$$

Where y_i is the activity level of the jth unit in the previous layer, w_{ij} is the weight (strength) of the connection between the ith and the jth unit.

The activation y_j using sigmoid function of the total weighted input is

$$y_j = 1/(1 + e)_8^{-x_j} \quad (4)$$

The network computes the error E, which is

$$E = ½\sum(y_i - d_i)^2 \quad (5)$$

Where y_j is the activation of the j^{th} unit in the top layer and dj is the desired output of the j^{th} unit.

The rate of change of error with the activation of an output unit (A) is the difference between the actual and the desired activity.

$$A_j = \frac{\partial E}{\partial E} \quad (6)$$

$$A_j = y_j - d_j \quad (7)$$

3.2. Choosing the Number of Hidden Neurons

We used the approaches reported in[16] to choose the number of hidden neurons. This is precisely obtained by dividing the sum of input neurons and output neurons by 2. For this research, 40 input neurons and 1 output neuron per data item were used.

$$N_h = (N_i + N_o)/2 \quad (8)$$

Where N_h = No. of Hidden Neurons
N_i = No. of Input Neurons
N_o = No of Output Neurons
$N_h = (40 + 1)/2$
$N_h \approx 21$

After setting the parameters above, the following steps were taken:

Training involved finding the error value for each Output Node which is

$$E_x = (A_x * (1 - A_x) * (D_x - A_x)) \quad (9)$$

Where E_x= Output Neuron Error on neuron x
A_x=Actual OutputActivation on x
D_x=Desired Outputs from x

The Hidden Neuron Error was calculated for each Neuron by totalling each dendrites weight multiplied by its output neurons error. The resultant value was then multiplied by the current neuron's activation.

We then used the Output Error to adjust the weights of the dendrites backward (Back-Propagation) from the Output Neurons to the Input Neurons dendrites. In[16] the new weight of each hidden-output dendrite value was set as shown in equation 10.

$$w_x := H_x * L_x * E_x + w_x \quad (10)$$

Where H_x= HiddenActivation
L_x = Learning Rate
E_x = Output Neuron Error on neuron x
w_x= Dendrite Weight

Above step was repeated with new Weights for each Input-Hidden Dendrite. At this point, the Network was configured for training. Fundamentally, one the reasons for training the network is to derive the input weight and dendrite strength for each neuron at which the network learns from training with training data (benchmark). Once these are derived, the network is saved into a *.nM#) file.

3.3. Training the Network

The network was initially tested with a linearly separable problem, in which it learned at about 2000 epochs. This gave a preliminary proof that the network was properly coded and that it could learn. The network started learning after being trained for about 500 epochs. We limited our configuration to 2000 epochs in order to avoid over-learning and memorization of data.

3.4. Running the Network

The training data (benchmark) and other candidates' data (data not used during training) were used to run the network.

The outputs were then saved into the database.

4. Results

Table 1 shows the score of each candidate in decimal.

The decimal candidate scores on each criterion in Table 1 were converted to binary numbers as shown in Table 2.

Table 2 below shows the candidate data, the digitalize data (input) and the output from the neural network.The last column shows the concatenated binary scores for the input into the neural engine. The corresponding output signals are shown in Table 3.

These output values show how each tested candidate stood compared with the benchmark. The results show that CAN006 compares best with the benchmark hence the selection order was 1.

All other candidates compared with the training data as reflected in their corresponding selection order. Our result showed CAN006 as the best fitting for the benchmark out of all candidates.

Table 1. Candidate Scores on all Criteria C_1 -C_{10} (Decimal)

CanID	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
CAN001	5	7	6	4	7	5	8	4	6	6
CAN002	5	7	5	4	5	5	6	4	7	6
CAN003	7	5	5	6	5	8	6	4	7	5
CAN004	8	5	8	7	5	5	6	4	8	5
CAN005	7	5	7	7	5	6	6	4	6	5
CAN006	8	9	9	9	8	9	9	8	9	8
CAN007	8	5	8	7	5	5	6	4	5	7
CAN008	7	6	6	7	4	5	6	6	5	8
CAN009	5	7	6	8	4	6	5	6	5	7
CAN010	8	9	8	9	8	9	9	8	9	8
CAN011	8	9	9	9	8	9	9	8	9	4

Table 2. Candidate Scores in Binary (Criteria 1 to 10 Obtained from Table 1) and Concatenated Binary Scores

CanID	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	Neural Network Inputs
CAN001	0101	0111	0110	0100	0111	0101	1000	0100	0110	0110	0101011101100100011101011000010001100110
CAN002	0101	0111	0101	0100	0101	0101	0110	0100	0111	0110	0101011101010100010101010110010001110110
CAN003	0111	0101	0101	0110	0101	1000	0110	0100	0111	0101	0111010101010110010110000110010001110101
CAN004	1000	0101	1000	0111	0101	0101	0110	0100	1000	0101	1000010110000111010101010110010010000101
CAN005	0111	0101	0111	0111	0101	0110	0110	0100	0110	0101	0111010101110111010101100110010001100101
CAN006	1000	1001	1001	1001	1000	1001	1001	1000	1001	1000	1000100110011001100010011001100010011000
CAN007	1000	0101	1000	0111	0101	0101	0110	0100	0101	0111	1000010110000111010101010110010001010111
CAN008	0111	0110	0110	0111	0100	0101	0110	0110	0101	1000	0111011001100111010001010110011001011000
CAN009	0101	0111	0110	1000	0100	0110	0101	0110	0101	0111	0101011101101000010001100101011001010111
CAN010	1000	1001	1000	1001	1000	1001	1001	1000	1001	1000	1000100110001001100010011001100010011000
CAN011	1000	1001	1001	1001	1000	1001	1001	1000	1001	0100	1000100110001001100010011001100010010100

Table 3. Result of CandidatesShowing Actual Output Values from Neural Network and Selection Order for Tested Candidates

CanID	Neural Network	Selection Order
CAN001	7.694887E-02	8
CAN002	9.628364E-02	7
CAN003	3.775475E-02	9
CAN004	0.753539	4
CAN005	2.052132E-02	11
CAN006	0.994003	1
CAN007	0.390748	5
CAN008	3.577022E-02	10
CAN009	0.172161	6
CAN010	0.993768	2
CAN011	0.993068	3

5. Conclusions

In this research, we are cognisant of the limitations that may likely be posed by three factors: insincerity of some candidates about their electoral promises;unavailability of funds to implement the contents of some party's manifestoes after successful elections andsubjective influence on selection process by "god fathers" as observed within some parties. We noted that these challenges are idiosyncratic and human–created. Therefore, the trend might not be observable in all parties or other spheres of application.

Sincesolving themwas not part of the aims ofthis research, information about them were neitherdeliberately nor directly processed in anyway.Hence, givena sane and safelevel of generalisation within the universe of discourse, this research demonstrated the applicability of synthetic training data to NNfor solving selection problem. The neural network has been able to establish a useful method ofselecting candidates for elective offices in Nigeria. There is also the possibility of usageacross political platforms as it is not strongly tied to a specific political system or elective post.It is also believed that the experiment offers an extensible template that may be applicable to solving problems in other spheres of candidate selection for posts aside political offices.

We do believe that if this system is put to use by political parties, killings, maiming, intimidations, bribery and mediocrity may be reduced in our society (Nigeria and other countries with fairly similar political construct).

REFERENCES

[1] Candidate selection within political parties, Retrieved February 1 2013 from http://aceproject.org/ace-en/topics/pc/ Candidate Selection within Political Parties — ACE Electoral Knowledge Network.htm.

[2] Pollick, M. (2013). How does a party choose a presidential candidate? Retrieved February 1 2013 from wiseGEEK, http://topics.wisegeek.com/ How Does a Party Choose a Presidential Candidate.htm. Revised By: A. Joseph, Edited By: Bronwyn Harris

[3] Johnson J., (2008). Why John McCain lost the election. Retrieved April 13, 2010, from (http://www.associatedcontent.com/article/1187924/ why_john_mccain_lost_the_election.html)

[4] Ayoade J., Okunade B., Vicente P., Bates R., Bratton M., Collier P., Widner J., (2010). The election process in Nigeria: Candidate selection, finance and outcomes . Retrieved April 13, 2010 from http://www.iig.ox.ac.uk/research/07-election-process-nigeria.htm

[5] Candidate selection (2010). Retrieved April 13, 2010, from Candidate Selection International Knowledge Network of Women in Politics (iKNOW Politics).mht at http://www.iknowpolitics.org/taxonomy/term/16

[6] Orji, C. (2010). Candidate imposition tears EbonyiPDP, retrieved April 16 2010 from http://www.nigerianbestforum.com/index.php?topic=68880.0.

[7] Cranor., L. F., (1996). Electronic Voting, Computerized polls may save money, protect privacy. Retrieved April 16, 2010 from www.acm.org/crossroads/xrds2-4/voting.html

[8] Stergiou, C. and Siganos, D. Neural networks (1996), Retrieved April 17, 2010 from http://www.doc.ic.ac.uk/~nd/ surprise_96/journal/vol4/cs11/report.html#Why use neural networks

[9] Luo, S., Wen, J., Huang, H. (2002). Knowledge-increasable learning behaviours research of Knowledge-increasable Learning Behaviors Research of Neural Field. Proceedings of the first international conference on machine learning and cybernetics, Beijing, 4-5 November 2002. pp 586 -590

[10] Miller, C. B., Chen D., Chen H. H., Sun G. Z. and Lee Y. C. (1992). Learning and extracting finite state automata with second-order recurrent neural networks, neural computation. May 1992, Vol. 4, No. 3, Pages 393-405. Posted Online March 13, 2008. Retrieved April 16, 2010 from http://www.mitpressjournals.org/doi/abs/

[11] Lein, K., 1997. Environmental Decision Making. Blackwell Science, Biddeford, Maine, p. 213.

[12] Lek, S., Belaud, A., Baran, P., Dimopoulos, I., Delacoste, M., 1996. Role of some environmental variables in trout abundance models using neural networks. Aquat. Living Resour. 9, 23-/29.

[13] Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J., Aulagnier, S., 1996. Application of neural networks to modellingnonlinear relationships in ecology. A.A. Ibarra et al. / Ecological Modelling 160 (2003) 281-/290 289 Ecol. Model. 90, 39-/52.

[14] Lek, S., Giraudel, J.L., Gue´gan, J.F., 2000. Neuronal networks: algorithms and architectures for ecologists and evolutionary ecologists. In: Lek, S., Gue´gan, J.F. (Eds.), Artificial neuronal networks: application to ecology and evolution. Springer-Verlag, Berlin, pp. 3 -/27.

[15] Carpenter, G. A. and Grossberg, S., 1998. The art of adaptive pattern recognition. self-organizing by a neural network. 0018-9l~~/88/0300-0077$000~ c 1988 IEEE. pp 77 – 88.

[16] Chesnut, C. 2004. "Chesnut tablet pc ocr with neural network AI". Retrived April 13, 2010, from http://www.generation5.org/content/2004/aiTabletOcr.asp.

[17] Drigas A., Kouremenos S., Vrettos S., Vrettaros, J., Kouremenos, D., (2004). An expert system for job matching of the unemployed, Expert Systems with Applications 26 (2004) 217–224.

[18] Akinwonmi, A.E., Adewale, O.S. Alese, B.K. and Adetunmbi. O.S. 2008. "Design of a neural network based optical character recognition system for musical notes". Pacific Journal of Science and Technology. 9(1):45-58.

[19] Nonnemaker, J. E. (2008). The safe use of synthetic data in classification a thesis presented to the Graduate Committee of Lehigh University in Candidacy for the Degree of Doctor of Philosophy in Computer Science & Engineering Dept. Lehigh University December, 2008. pp 42 -48

[20] T. K Ho and H. S. Baird. Large-scale simulation studies in image pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(10):1067–1079, 1997.

[21] Simard, P., Le Cun, Y., Denker, J., and Victorri, B.

Transformation invariance in pattern recognition - tangent distance and tangent propagation. In G. B. Orr Miller and K-R, editors, Neural Networks: Tricks of the Trade, volume Chapter 12. Springer, 1998.

[22] T. Varga and Bunke H. Comparing natural and synthetic training data for off-line cursive handwriting recognition. In IEEE, editor, 9th International Workshop on Frontiers in Handwriting Recognition, 2004.

[23] Young, J. (2008). Synthetic tables: a prototype for creating synthetic census tables. In Graham, P., Young, J., & Penny, R. (2008). Preface: methods for creating synthetic data, the official statistics system, Wellington, Official Statistics Research Series, Vol 3

[24] Young, J. (2008). Using Bayesian networks to create synthetic data. In Graham, P., Young, J., & Penny, R. (2008). Preface: Methods for Creating Synthetic Data, The Official Statistics System, Wellington, Official Statistics Research Series, Vol 3 ISSN 1177-5017

[25] Papadourakis, G. Ph.D.,(2013)Introduction to neural networks. Retrieved on 12 January 2013 From: www.ohio.edu/people/starzykj/network/Class/ee690/EE690_CellularNN_Reiforcement/Slides/CNN/NeuralNets%2520 overview.ppt