APPLICATION OF SELF-ORGANIZING MAP TO INTELLIGENT ANALYSIS OF CELLULAR NETWORKS

Joseph M. Mom[1] and Cosmas I. Ani[2]
[1]Department of Electrical and Electronics Engineering, University of Agriculture, Makurdi, Nigeria
[2]Department of Electronic Engineering, University of Nigeria, Nsukka, Nigeria
E-Mail: josseff.mom@gmail.com

ABSTRACT

In this work, the efficacy and scalability of the self-organizing map (SOM) algorithm, which is a class of artificial neural network (ANN), over traditional methods of analyzing cellular network variables was shown using key performance indicators (KPIs) data collected from an operational network service provider in Nigeria. Performance trends of various cells over a period of time were evaluated and rules of significance measure extracted which could form the basis for network optimization.

Keywords: self-organizing map (SOM), cellular network, artificial neural network (ANN), key performing indicator (KPI).

INTRODUCTION

Mobile cellular networks have over the years witnessed rapid advancements. This progress makes the operations and maintenance of services more demanding as there will be corresponding increase in the number of network variables that need to be monitored and evaluated. The increase will pose serious challenges to service providers seeking to optimize their networks for efficient service delivery. The system knowledge required to optimize cellular system performance is very difficult to formalize as a mathematical model [1]. However, the behaviour of a network is usually embedded in the data collected from it [2]; therefore, in tackling these challenges so many data variables will be required to be analyzed simultaneously for emerging network trends and performance to be correctly observed. The traditional or static data analysis methods are deficient in this regard [3], as such intelligent data mining analysis methods which are capable of adjusting to changes in the environment and making logical deductions from information being processed are highly desirable for such networks. Artificial neural networks (ANNs) are a class of such intelligent methods and they have been shown to be very effective in the visualization and grouping of similarly behaving classes of data and it is also capable of handling multidimensional data simultaneously.

In recent times, artificial neural network (ANN) algorithms have been used in varying applications to provide solutions to a number of problems from given data sets [4], [5], [6], [7]. ANN is a system based on the imitation of the operation of biological neural systems. ANNs have different architectures which include Adaptive Resonance Therapy (ART), Back propagation Neural Network (BNN), Hopfield Neural Network (HNN), Kohonen's Self-Organizing map (SOM), and Perceptron and Adaline (PA). Other methodologies that are related to ANN are fuzzy logic and genetic algorithms (Gas). ANNs are basically used as pattern classifiers and as nonlinear adaptive filters, by adaptive, it means that each parameter is changed during its operation and it is deployed for solving future problems.

ANN is developed with a systematic step-by-step procedure which optimizes a criterion commonly known as the learning rule. The input/output training data is fundamental for these networks as it conveys the information which is necessary to discover the optimal operating point. In addition, a nonlinear makes neural network processing elements a very flexible system. In a typical ANN system, there is a learning phase where an input dataset is presented to the system and the desired response is set at the output. After training the input data, an error comprising the difference between the desired response and system output is compiled and fed back to the system. The system then makes adjustments severally until the desired output is obtained; this is known as the learning rule.

Mathematical model of ANN

The basic mathematical model of ANN consists mainly of the weights, summing functions and activation functions. All inputs x_j are summed together and modified by the weights w_{kj}. This activity is referred to as a linear combination. Finally, an activation function $\varphi(\cdot)$ controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be -1 and 1. Mathematically, this process is described in Figure-1.

ARPN Journal of Engineering and Applied Sciences

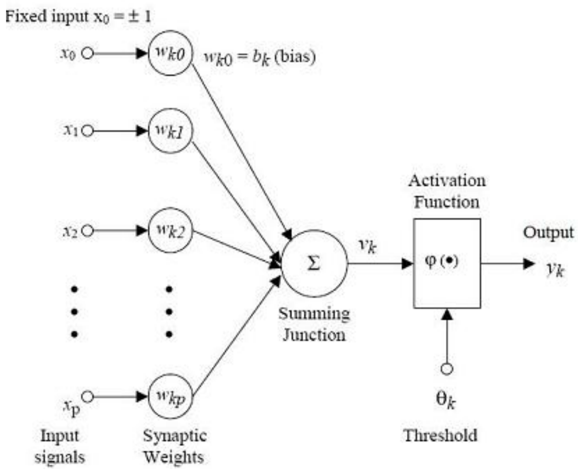

Figure-1. Mathematical model of ANN.

The interval activity of the neuron can be shown from Figure-1 to be:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j \qquad (1)$$

The output of the neuron, y_k , would therefore be the outcome of some activation function on the value of v_k. As mentioned previously, the activation function acts as a squashing function, such that the output of a neuron in a network is between certain values (usually 0 and 1, or -1 and 1). In general, there are three types of activation functions, denoted by $\varphi(\cdot)$. First, there is the threshold function which takes on a value of 0 if the summed input is less than a certain threshold value v and the value 1 if the summed input is greater than or equal to the threshold value.

$$\varphi(v) = \begin{cases} 1, & if\ v \geq 0 \\ 0, & if\ v < 0 \end{cases} \qquad (2)$$

Secondly, there is the piecewise-linear function. This function again can take on the values of 0 and 1, but can also take on values between that depending on the amplification factor in a certain region of linear operation.

$$\varphi(v) = \begin{cases} 1, & v \geq \frac{1}{2} \\ v, & -\frac{1}{2} > v > \frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \qquad (3)$$

Thirdly, there is the sigmoid function. This function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function.

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)} \qquad (4)$$

The architecture of each ANN is based on very similar building blocks which perform the processing. The analysis that will now follow is based on the self-organizing map (SOM) algorithm.

The SOM algorithm

The SOM algorithm in its basic form consists of a regular grid of map units or neurons [3], [8]. They are connected to the neighbouring units using, for instance, a rectangular or hexagonal neighbourhood. Each map unit, denoted here by i , is represented by a prototype vector m_i . The dimension of the prototype vectors is equal to the dimension of the input data. First, the prototype vectors are initialized, for instance, to random values. Then, during the training the values of the prototype vectors are adapted to follow the properties of the input data. Training of the SOM is divided into two alternating steps, typically thousands of times each. First, one data vector x from the training set is randomly selected and the corresponding best-matching unit (BMU) c is determined. The prototype vector of the BMU, denoted by m_c , is the one nearest to the data sample. In other words, it minimizes the Euclidean distance between x and m_c .

$$c = \arg \min_{i} \|x - m_i\| \qquad (5)$$

In the next step, the prototype vectors of the winner and its neighbours are moved towards the data vector. It should be noted that the neighbourhood is defined in terms of the lattice structure, not according to the distances between data samples and prototype vectors in the input space. The update step is performed by applying

$$m_i(t+1) = m_i(t) + \alpha(t) h_c(t,i)[x(t) - m_i(t)] \qquad (6)$$

Where $\alpha(t)$ is the learning rate and $h_c(t,i)$ is the neighbourhood function of the algorithm. The last term in the square bracket is proportional to the gradient of the squared Euclidean distance $d(x, m_i) = \|x - m_i\|^2$. The learning rate $\alpha(t) \in [0\ 1]$ is usually a monotonically decreasing function of time. A good candidate is $\alpha(t) = \alpha_0 \left(1 - \frac{t}{T}\right)$, where, α_0 is the initial value for the learning rate and T is the total number of training iterations. A very frequently used form for the neighbourhood function $h_c(t,i)$ is the Gaussian one centered on the winner map unit c .

$$h_c(t,i) = exp\left(-\frac{\|r_i - r_c\|^2}{2\sigma(t)^2}\right) \qquad (7)$$

ARPN Journal of Engineering and Applied Sciences

Where r_c depicts the coordinates of the winner unit c and r_i denotes the coordinates of an arbitrary unit i on the discrete output lattice of the map and $\sigma(t)$ is the width of the neighbourhood. It is necessary that $h_c(t,i) \rightarrow 0$ when $t \rightarrow \infty$ for the algorithm to converge. During learning, the learning rate and the width of the neighbourhood function are decreased, typically in a linear fashion. The map converges to a stationary distribution, which approximates the density of the data.

Methodology of the study

The flow chart of the methodology for the performance analysis is shown in Figure-2. Data will be collected, processed, trained and clustered to enable summarization of network trends that can be used to predict network behaviour.

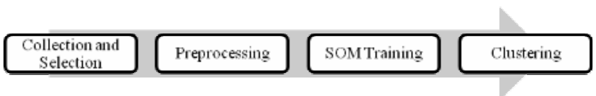

Figure-2. Methodology for data analysis.

Data collection and selection

The key performance indicators (KPIs) data were collected from a well-established operational cellular network in Nigeria. The study area chosen for this work is the Nsukka geographical region shown in Figure-3, which is a proportional representation of the data collected. This region is situated in Enugu State, South Eastern Nigeria, and is serviced by a BSC, 28 BTSs with 28 sites divided into three sectors given a total of 84 cells. The data which is henceforth called "Nsukka Data" is a 84 x 5 matrix mean KPI values collected over a period of 12 weeks. The KPIs used are call setup success rate (*cssr*), drop call rate (*dcr*), handover success rate (*hosr*), traffic channel congestion (*tchcong*) and busy hour traffic (*bhtraffic*).

Figure-3. Nsukka geographical region.

Data preprocessing

The data was preprocessed in Matlab using the SOM Toolbox [9]. Scaling of the data was done using the range method, where the values were scaled between 0 and 1 to remove the effect of large numbers on the performance of the SOM.

Data training

The prototype vector was trained using batch algorithm and the best matching units corresponding to the data set were obtained called codebook vectors. These codebook vectors are mapped to corresponding map units in different data clusters. Figure-4 show an 8 x 8 size component planes of the data clusters of the cells obtained for *cssr, dcr, hosr, tchcong,* and *bhtraffic*. Similarly behaving cells are grouped together, or near each other, the component planes depict a map of data clusters, where a group of cells are mapped to a cell position based on its trained characteristics and aggregate performance in relation to other cells. The bar codes on the right of the component planes show the denormalized value of the KPI data vectors.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

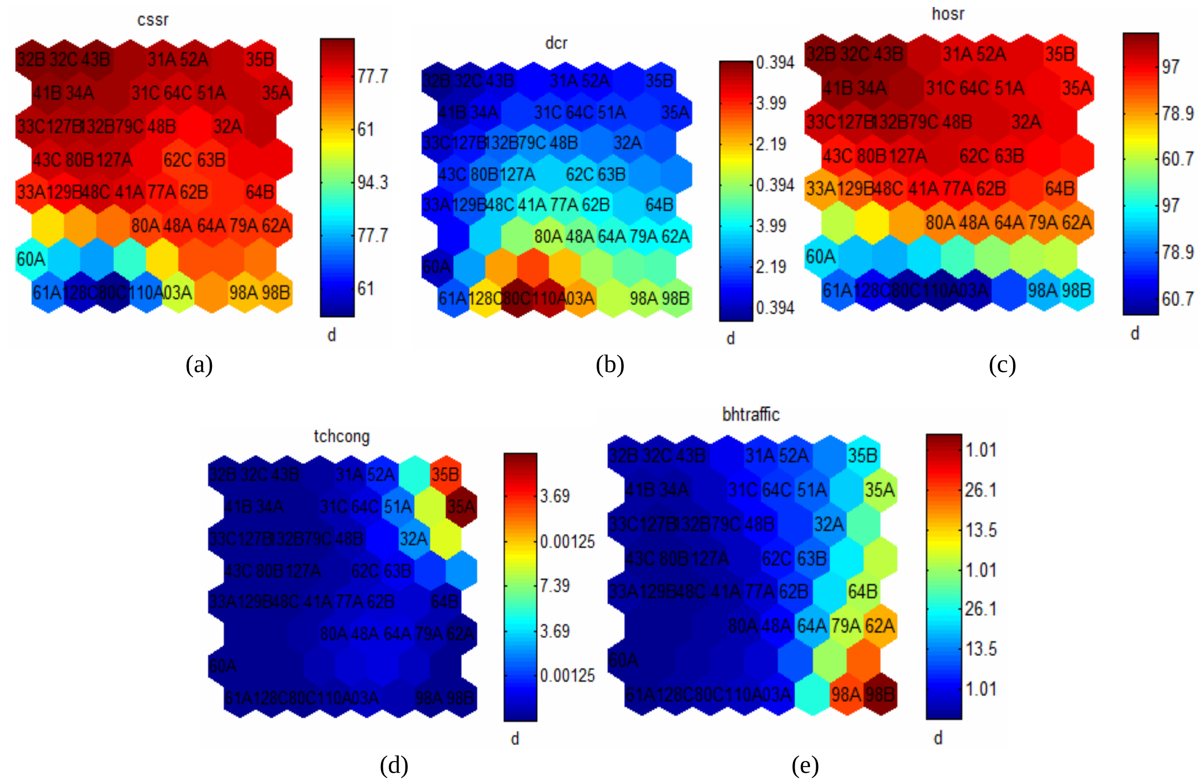

Figure-4. Component planes of data vectors (a) *cssr* (b) *dcr* (c) *hosr* (d) *tchcong* (e) *bhtraffic*.

The performance several cell sites over a period of six months is summarized in Table-1; the classification is done on the basis of cells with best, average and worst performance. Generally, the best performing cells will be the ones which minimize or maximize the chosen KPI based on established benchmarks by the Nigerian Communications Commission [10].

Table-1. Summarized cell performance ranges.

KPI	Best	Average	Worst
cssr	Cells at the top left (> 92%)	Cells at the middle right (85% - 92%)	Cells at bottom left (< 70%)
dcr	Cells at the top left (< 0.13%)	Cells at the middle and bottom right (0.13 - 2.5%)	Cells at the bottom middle (> 2.5%)
hosr	Cells at the top (> 90%)	Cells at the middle (80% - 90%)	Cells at the bottom (< 80%)
tchcong	Cells at the left and bottom right (<0.02%)	Cells at the middle right (0.02% - 3%)	Cells at the top right (>3%)
bhtraffic	Cells at the left (2)	Cells at the right (2 - 15)	Cells at the bottom extreme right (>15)

Data clustering analysis

The map vectors were further clustered using the k-means clustering algorithm to obtain behavioural patterns. The number of clusters in NsukkaData is 4, this was determined using the Davies-Bouldin index [11]. Figures 5(a) and 5(b) show the behaviour clusters and group of cells with similar behavioural patterns grouped together in 4 clusters. In Figure-5(c), a corresponding histogram representation of the behavioural clusters is shown; the bar colour coding indicates from left to right, *cssr*, *dcr*, *hosr*, *tchcong* and *bhtraffic*.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

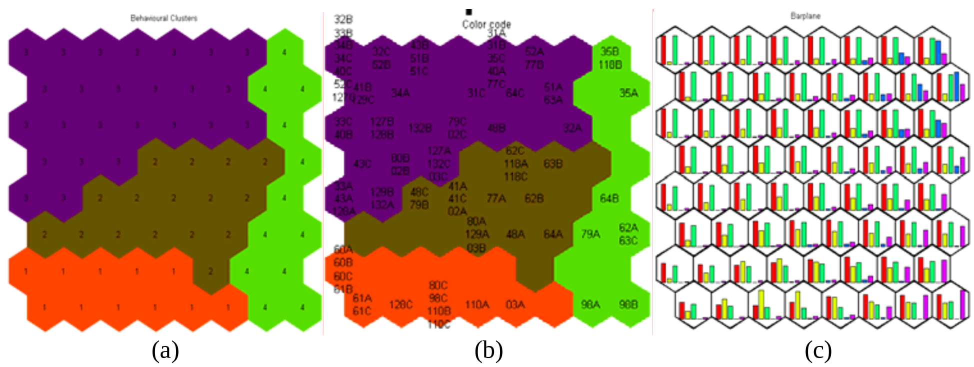

Figure-5. Clustering and classification of cells.

After clustering the cells, it can be seen from Figures 5(a) and 5(b) that cells occupying the same map unit has the same overall performance as characterized by values for *cssr*, *dcr*, *hosr*, *tchcong* and *bhtraffic* as shown in Figure-5(c). This visualization can be seen at a glance and in relation to other cells, this contrast sharply with the traditional method where several plots must be observed together to see the inherent behaviour of these cells

Each class is an aggregate categorization of cells with similar behaviour based on the multiple KPIs combined together. From the information obtained from Figures 4 and 5, rules particular for each behaviour class can be extracted as shown in Table-2. These rules optimize the given significance measure and can form the basis for optimization of cell performances to give optimal results, as the characteristics of best performing cells can be used to correct worst performing cells. Table-2 show the KPI rules that optimize the given significance measure

Table-2. SOM rules for data clusters.

KPI/Rule	Low	High	Rule indices
cssr	0.92	1.00	0.92 <= *cssr* < 1.00
dcr	0.00	0.13	0.00 <= *dcr* < 0.13
hosr	0.85	1.00	0.85 <= *hosr* < 1.00
tchcong	0.00	0.06	0.00 <= *tchcong* < 0.06
bhtraffic	0.00	0.10	0.00 <= *bhtraffic* < 0.10

The relative spread of the KPIs in relation to one another can be visualized in the form of histograms and scatter plots as in Figure-11.

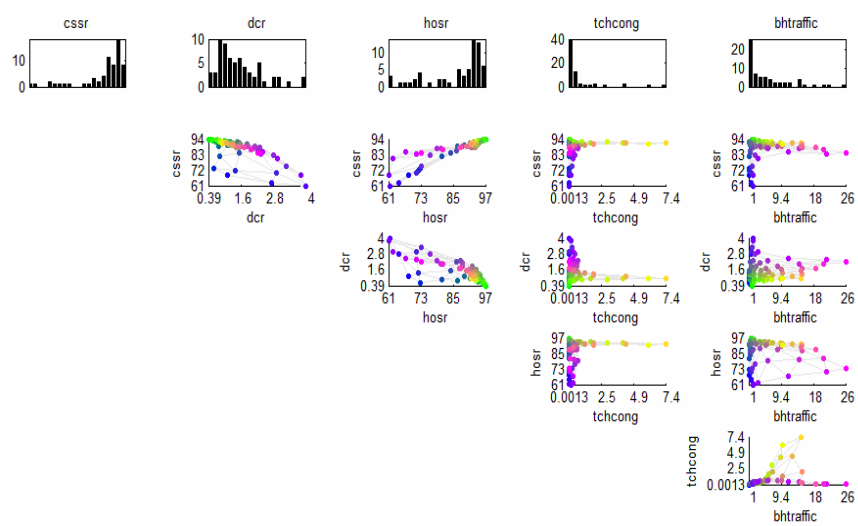

Figure-11. Histograms and scatterplots of Nsukka data.

CONCLUSIONS

Starting from a multidimensional data obtained from a live operational cellular network, the SOM was used to intelligently analyze and clearly bring out behavioural classes and performance of cells within the network. This approach performed better as compared to traditional means of analysis. The results of this analysis can be further used to optimize the performance of the network where qualities of better performing cells can be copied and used to 'correct' less performing ones. Additional data can also be easily classified based on the learning ability of the SOM.

ACKNOWLEDGEMENTS

This work has been sponsored in part by grant from the TETFund, Abuja-Nigeria. We acknowledge the contributions of members of the Communication Research Group of the Department of Electrical and Electronics Engineering, University of Agriculture, Makurdi-Nigeria and the Department of Electronic Engineering, University of Nigeria, Nsukka.

REFERENCES

[1] P. Lehtimaki and K. Raivio. 2005. A SOM based approach for visualization of GSM network performance data. Helsinki University of Technology, Laboratory of Computer and Information Science, P.O. Box 5400, FIN-02015 HUT, Finland.

[2] J. Han and M. Kamber. 2006. Data Mining: Concepts and Techniques, 2nd Edition, Morgan Kaufmann Publishers, San Francisco, CA 94111.

[3] J. Laiho, K. Raivio, P. Lehtimaki, K. Hatonen and O. Simula. 2005. Advanced analysis methods for 3G cellular networks. IEEE Transactions on wireless communications. 4(3).

[4] K. Hatonen, P. Kumpulainen and P. Vehvilainen. 2003. Pre- and post-processing for mobile network performing data. In: Proceedings of seminar days of Finnish Society of Automation, (automation 03). Helsinki, September, 9-11.

[5] B. Haider, M. Zafrullah and M.K. Islam. 2009. Radio frequency optimization and QoS evaluation in operational GSM network. In: Proceedings of the World Congress on Engineering and Computer Science. Vol. I WCECS 2009, October 20-22, San Francisco, USA.

[6] S. Kyriazakos, G. Karetsos, E. Gkroustiotis, C. Kechagias and P. Fournogerakis. 2001. Congestion study and resource management in cellular networks of present and future generation. IST Mobile Summit 2001, Barcelona, Spain, 9-12 September.

[7] G.A. Bareto, J.C.M. Mota, L.G.M. Souza, R.A. Frota and L. Aguayo. 2005. Condition monitoring of 3G cellular networks through competitive neural models. IEEE Transactions on Neural Networks. 16(5), September.

[8] J. Laiho, P. Vehvilainen, A. Hoglund, M. Kylvaja, K. Valkealahti and T. Bout. 2006. Advanced Analysis Methods and Radio Access Network Autotuning. Radio Network Planning and Optimization for UMTS, 2nd Edition. J. Laiho, A. Wacker and T. Novosad (Eds.). John Wiley and Sons, Ltd.

[9] J. Vesanto, J. Himberg, E. Alhoniemi and J. Parhankangas. 2006. SOM toolbox for Matlab 5. Tech. Rep. A31, Helsinki University of Technology, Laboratory of Computer and Information Science. Available: http://www.cis.hut.fi/projects/somtoolbox

[10] 2008. Nigerian Communication Commission, Quality of Service Regulations.

[11] D.L. Davies and D.W. Bouldin. 1979. A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence. 1(2): 224-227, April.