Neural Networks as a Tool for Product Manufacturing Innovation in Africa

Ima Okon Essiet, Ado Dan-Isa
Dept. of Electrical Engineering
Bayero University, Kano
Nigeria
imaessiet82@gmail.com
ado_danisa@yahoo.com

Abstract—**This paper highlights the numerous advantages of process simulation using neural networks. Apart from reviewing some successful industrial applications of neural networks (specifically in the field of electrical engineering), results of the authors' research in the areas of food security and health will also be presented. The research results will show that successful neural simulation results using *Neurosolutions* software also translated to successful real-time implementation of cost-effective products with reliable overall performance of up to 90%.**

Keywords—neural network; ammonia, back-propagation, NeuroSolutions, supervised learning.

I. Introduction

Neural networks are highly parallel, dynamic systems capable of performing data manipulations by means of their state response to input information [1]. Neural networks have been used as a means of pattern recognition and data classification for decades. These networks mimic the way the human brain processes information through learning from past experience and parallel processing of information. Neural networks are particularly suited to tasks for which transformations of certain inputs to certain outputs have been established, but the transformations cannot be analytically determined due to certain constraints. Despite the fact that neural networks have been around since the 1940s, it was in the late 1980s that they began to be applied in industry. This is because the 'black box' approach to neural processing caused a lot of scepticism among researchers for nearly 50 years. However, since then, there has been a steady rise both in the frequency and areas of application of artificial neural networks from prediction and system modelling to pattern recognition systems and robust classifiers.

There are two common ways of classifying neural networks. The first is by the network structure and the second is by the training technique. In terms of network structure, networks can be described as multilayered perceptron (MLP), feedforward neural network (FNN), recurrent neural network (RNN), radial basis function (RBF), Kohonen and Hopfield and so on. Neural networks can be trained through either supervised or unsupervised (self-organized) approach. In spite of the structural and learning differences among these neural networks, they all receive information from the outside world, process this information and generate results based on the outcome of the processing. Therefore, the degree of success of a neural approach to solving a problem depends on how well it can learn and approximate relationships between inputs and outputs.

This paper provides an overview of industrial applications of neural networks while also answering pertinent questions some of which include: what problems require a solution by neural networks? What is the best neural model for a specific problem? Why are neural networks better than conventional methods? Neural applications to product development from the authors' research will also be presented with results showing the immense importance of such networks in successful product development.

II. EVOLUTION OF NEURAL NETWORKS: DISCOVERY TO INDUSTRIAL APPLICATION

The advent of the neural network began in 1943 when Walter Pitts and Warren McCulloch proposed that it was possible to mathematically model the behaviour of the biological neural system [2]. This model came to be popularly known as an *artificial* neural network or ANN. Later in 1961, the term *cybernetics* was used by Norbert Weiner to describe this new field which linked engineering, biology, control systems and brain function [3]. This interdisciplinary discovery led to the development of the Von Neumann computer. This computing model led to the development of a learning machine based on Hebb's rule [4]. The idea of the *perceptron* was proposed by Frank Rosenblatt, head of the cognitive systems division of the Cornell Aeronautical Laboratory in 1958. This perceptron formed the basis of the present day neural networks and led to the realisation of the Mark I perceptron in 1960. The Mark I could learn new skills by trial-and error by means of a neural network capable of simulating human thought processes. Widrow and Hoff proposed the adaline or adaptive linear element in 1960 which was similar to the perceptron but included a form of learning via an error correction rule. This technique was based on the least mean square (LMS) algorithm which was successfully used for echo cancellation in telephone lines. This was the earliest industrial application of neural networks [5].

The growing popularity of neural networks particularly between the 1950s and 60s was thwarted by Marvin Minsky and Seymour Papert in 1969. In their book *Perceptrons*, they proved mathematically that the perceptron could not be applied in problems that presented as nonseparable logic functions [6]. Afterwards, the 1970s were a dry period for research involving neural networks with perhaps the most notable being the discovery of back-propagation by Paul Werbos in 1974. This learning algorithm was later rediscovered in the mid-80s by Parker and Rumelhart et al [7, 8]. During this period, various forms of the neural network were realised such as the Boltzmann Machine by Sejnowski, Hinton and Ackley [9], radial basis function (RBF) networks by Broomhead and Lowe [10] and the Hopfield recurrent neural network [11].

Renewed interest in the field by the 1980s prompted DARPA to undertake a study of neural applications in industry. The study listed various applications of neural networks including commercial, modelling, control, image and speech recognition, and planning. Many control and classification processes have been successfully implemented using a number of neural structures (in some cases a combination of structures). Neural networks are often applied to pattern recognition and classification problems which can either be generic or specific. A generic problem involves methods which are not specific with regard to the parameters and variables used. They are therefore referred to as *domain-free* problems. A specific problem on the other hand, is defined by its parameters, values and constraints which depend on the application area in which the problem arises [12]. Other application areas of neural networks which involve heuristics include image and speech recognition, character recognition as well as planning and forecasting. The continued success of the neural network in industrial applications is the main motivation behind this paper. In addition to reviewing other work, the paper will present the authors' work in the area of neural simulations and will also discuss how successful results were translated into real-time, functional devices.

III. CHOICE OF NEURAL NETWORK STRUCTURES AND DATA REQUIREMENTS

The success of a neural network's performance depends largely on the structure of the chosen network for the specific problem and also on the parameters used to train the network. This section discusses these important concepts.

A. Neural Network Structures

There are several neural network structures that have been successfully implemented in various engineering applications. Table I shows a classification of neural network structures based on their areas of application.

- Multi-layer Perceptron (MLP): This is a structure in which the output of each neuron (node) is connected to neurons in subsequent layers in cascade without any intra-layer connections. This structure was introduced to overcome the limitations of the single-layer perceptron demonstrated by Minsky and Papert in 1969. They showed that the single-layer perceptron could only solve convex problems. However, Rumelhart et al showed that a 2-layer MLP could solve non-convex problems. Networks with 3 or more layers can essentially solve boundless problems.

  MLPs have been used in several applications such as speed control of dc motors [13, 14], induction motor fault diagnosis [15-17], induction motor control [18-20], feedback control [21-23] and fault diagnosis of robotic systems [24]. The structure of the MLP is shown in Fig. 1.

- . Recurrent Neural Network (RNN): This neural network structure is realised by feeding the network's output back into the input after a learning session (epoch) has been completed. It was first proposed by Rumelhart et al in 1986 [25]. The RNN structure can be likened to back-propagation with few hidden layers with each recurrent cycle representing exactly one instance of hidden layer activity. Examples of the RNN include Jordan, Elman and Hopfield networks and Boltzmann machines.

  Recurrent neural networks have been applied to analyse spectral content of noisy periodic waveforms common in engineering processes [26]. A Hopfield network was also employed to detect and isolate faults in linear dynamic systems [27].

- Adaptive Resonance Theory (ART) Neural Network: The ART network was developed to function as a self-organizing network by retaining knowledge of previously learned patterns. In other words, it closely resembles the biological neural network's capability of learning from past experience. The ART network comprises a comparison layer, a recognition layer and a reset element. The comparison layer consists of three (3) inputs: the recognition layer output, the input vector and a gain g_1. The output of this layer is 1 if and only if at least two of its inputs are 1. The recognition layer is used as a classifier and also has three (3) inputs: the reset element's output, a vector v_j, and a gain g_2. The recognition layer neuron with the winning combination of vector v will output a 1 if and only if $g_2 = 1$. All other combinations will output 0.

  Hence, the recognition layer classifies the input vector. The ART network has been successfully applied in sensor pattern interpretation [28].

- Kohonen Network: This is another form of self-organizing network whose topology is different from that of the ART network. In Kohonen networks, the output nodes are ordered in the form of an array determined by the user. The ordering process involves selecting which set of output nodes are neighbours [29]. When learning patterns are presented to the network, output nodes are adapted in such a way that the order of the input space is replicated at the output. In other words, learning patterns which are close to each other in the input space must be mapped unto output units which are also close to each other. Assume the input vector is I^N and a sample $\mathbf{s}(t)$ is presented to the network. The winning node $\mathbf{u}$ is adapted using the Equation (1):

$$w_o(t+1) = w_o(t) + \eta d(o,u)(s(t) - w_o(t)) \quad (1)$$

The collective learning scheme in Equation (1) ensures that input signals which are near to each other are mapped unto neighbouring neurons. Therefore, the topology in the input signals is preserved in the mapping. The Kohonen network has been extensively used in classification and pattern recognition applications [30].

Probabilistic Neural Network (PNN): This neural structure is similar to the MLP. They differ from MLPs in terms of activation functions (usually exponential functions) and synaptic patterns. Unlike the MLP, the hidden neurons are usually not fully connected. This fewer numbers of connections makes this form of neural network easy to train. The PNN operates in parallel with the signal flowing in one direction only [31]. PNNs have been used in the identification of transients in nuclear power plants [32].

- Radial Basis Function (RBF) Networks: This network structure consists of RBF nodes as process units. The input nodes all connect to these hidden units with the output nodes being summations [31]. RBF networks are particularly suitable for fault diagnosis applications because they are fast to train compared to MLPs for instance. They have been used to train a robotic hand [33], for generator system control [34], power electronic drives and digital signal processors [35].
- Polynomial Neural Networks: Polynomial networks are termed 'plastic networks' because their structure is formed during the training process. As a result, no two applications can have the same polynomial network structure. The automatic feature selection capability of polynomial networks makes them useful in control applications in which plant order is unknown. Polynomial networks have been used to implement filter design [36].

B. Data Requirements

The data requirements for successful implementation of a neural network for a specific application answer the important questions of when, how and exactly which problems require a neural-based solution. Problems that are heuristic in nature are typically good candidates for neural solutions. Other neural-solution based problems include those that require classification, regression or pattern recognition of large solution spaces requiring a generic solution. The following are some of the important data requirements which must be ascertained to obtain a neural structure likely to solve the problem:

- Network Structure: This answers the question regarding whether a single-layer, multi-layer or recurrent network should be employed. A single-layer network (the adaline, for example) is particularly suitable for logic-based applications. Some logic tasks such as the XOR implementation require an additional layer for obtaining accurate output. Multi-layer networks (such as the MLP) are suitable for classification and pattern recognition purposes. Recurrent networks have been successfully applied to data recognition tasks such as handwritten character recognition [37].
- Activation Function: The manner in which neurons interact with each other in the neural network is determined by the choice of activation function. The activation function is usually a non-decreasing function of the total unit inputs such that the neuron outputs are as close as possible to their individual activation levels [29]. There are several types of activation functions. The most common are the linear, semi-linear and sigmoid functions. The combination of both activation and output functions determine the transfer function [38].
- Interconnected Weights: These connect the layers of the neural network and are used to adjust layer output(s). Assume two interconnected neurons a and b have a weight connection W_{ab} such that $\{W_{ab} \in 0.......1\}$. The closer the connected weight value is to 1, the greater the importance of that connection.
- Data Flow: Here it has to be determined whether or not data flow is to be recurrent. In other words, the designer should ascertain whether data flow within the network layers should be fed forward continually or fed back at a point.
- Input Signals: There are a number of input signals that a neural network can process. These include binary, bipolar and continuous values. Binary input values can be either 0 or 1, while bipolar values are either -1 or 1. Continuous real numbers within a certain range constitute continuous input signals.

Over the years, the MLP has been the most commonly used NN structure. As at the year 1995, the percentage of network utilisation was as follows: MLP, 81.2%; Hopfield, 5.4%; Kohonen, 8.3%; others, 5.1% [39]. Fig. 2 shows the utilisation in power engineering as at 2007. The MLP utilisation is about 33%, which is the highest percentage.

TABLE I. CLASSIFICATION OF NEURAL NETWORK STRUCTURES BASED ON THEIR APPLICATION AREAS[31]

Functional Characteristics	Structure
Pattern Recognition	MLP, Hopfield, Kohonen, PNN
Associative Memory	Hopfield, recurrent MLP, Kohonen
Optimization	Hopfield, ART, CNN
Function Approximation	MLP, CMAC, RBF
Modeling and Control	MLP, recurrent MLP, CMAC, FLN, FPN
Image Processing	CNN, Hopfield
Classification (including Clustering)	MLP, Kohonen, RBF, ART, PNN

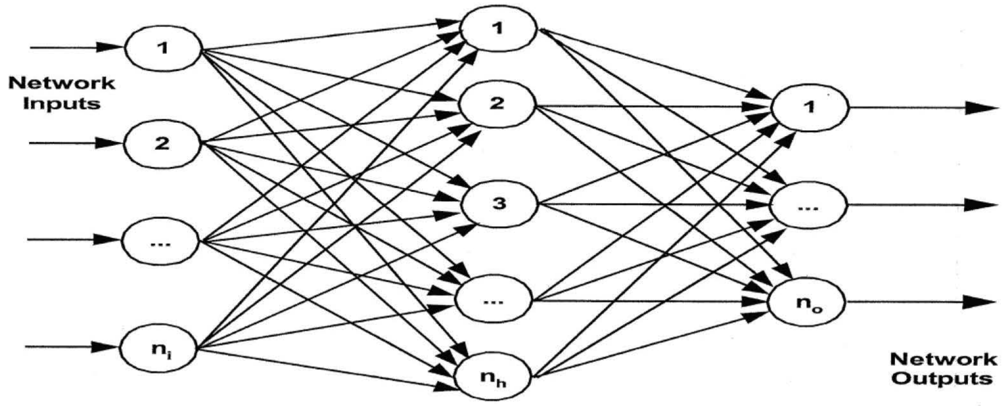

Fig 1. Structure of the MLP [31]

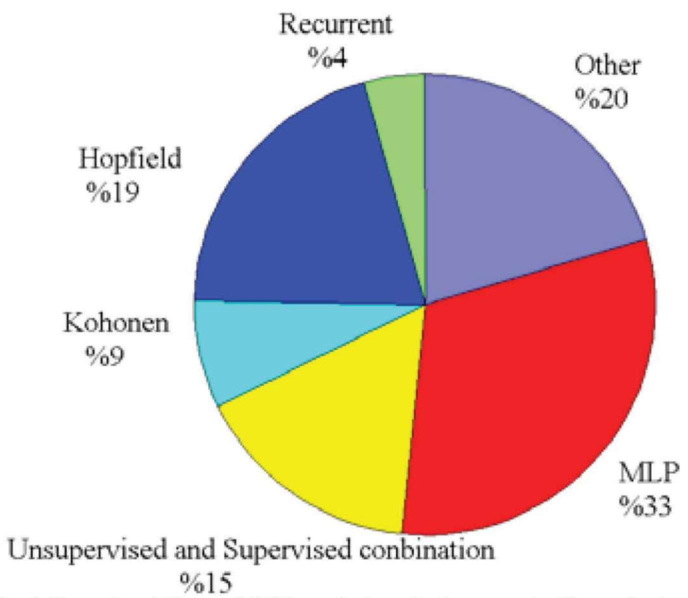

Fig. 2. Proportional Usage of NN Types in Security Assessment of Power Systems [40]

III. NN APPLICATIONS IN ENGINEERING

Since the 1980s, application of neural networks to the analysis of engineering-based problems has been on the increase. This phenomenal growth is attributed to the success of results obtained from neural simulations over the years. Today, neural networks are being used in control, power and telecommunications engineering as well as robotics and instrumentation. This section outlines the use of neural networks in power system security assessment and data classification for electronic sensor circuits.

A. Security Assessment of Power Systems

Security assessment is the process of determining the presence and extent of interference to the normal operation of power systems. It also involves determining system robustness to a set of contingencies in its present or future state [41]. There are basically two forms of security assessment: static and transient. The neural approach was proposed because traditional methods of security evaluation based on load flow and transient stability analysis are unrealistic for real-time implementation [41]. Table IV shows the simulation results.

Training and testing vectors for the NN training were obtained by sequential forward selection (SFS) method. NN model implementation was carried out in IEEE 14 bus, 30 bus and 57 bus test systems repectively. Results obtained showed that PNN and ARTNN gave high classification accuracies of 100% and 96% respectively. Mean squared error (MSE) and classification accuracy are evaluated according to Equations (2) and (3).

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(E_k)^2 \quad ; E_k = \left|DO_k - AO_k\right| \tag{2}$$

Where n=no of samples in data set

DO_k=desired output obtained from offline simulation

AO_k=actual output obtained from NN trained classifier

$$CA(\%) = \frac{no\ of\ samples\ classified\ correctly}{total\ no\ of\ samples\ in\ data\ set} \times 100\% \tag{3}$$

B. Neural Simulation of Food Classification and Tooth Decay Sensor Circuits

This section focuses on the authors' research in the application of neural network to the real-time realisation of electronic circuits for the classification of food condition and identification of tooth decay respectively. The neural software used in the simulation is NeuroSolutions version 5 training software. The purpose of the neural simulations is to select an appropriate sensor capable of yielding the most accurate data classification for the applications mentioned above. In both cases, an MLP was used to obtain the best simulation results. The results obtained in this work answer the three most important questions raised in the introduction section viz: the task associated with both applications is that of data classification, which is particularly suited to neural networks. From both literature and experimentation, the MLP has been found to give the most accurate classification results. The neural approach has been found to be better than conventional methods in these applications because it is cost-effective in terms of both time and resources. Conventional approach would basically involve a trial-and-error approach to selecting best-performing sensor(s). Normalized data samples were obtained from constructed ammonia sensor circuits using a TGS 2602 metal oxide semiconductor (MOS) sensor. Tables II and III show simulation results for food and tooth decay data classifications respectively. Overall accuracy of 92.3% and 85% for the food and tooth decay classification neural networks respectively have resulted in the real-time implementation of electronic circuits capable of performing these classifications as shown in Fig. 3.

Further research is presently being carried out to improve the accuracy of the constructed circuits by fine tuning training data for the neural network model. From research carried out so far, the MLP has yielded the most accurate classification results. It was also observed that MLP models with more than two process layers resulted in delayed convergence, likely due to over fitting of training data points in relation to the target data.

TABLE II. NEURAL SIMULATION RESULTS FOR FOOD CLASSIFICATION NN

	Bad (predicted)	Not Bad (predicted)	Accuracy %
Bad (actual)	92.3	7.3	92.3
Not Bad (actual)	7.7	92.3	92.3
Overall Accuracy			**92.3**

TABLE III. NEURAL CLASSIFICATION RESULTS FOR TOOTH DECAY BREATH SAMPLES

	Kidney failure (predicted)	Non kidney failure (predicted)	Accuracy %
Kidney failure (actual)	9	1	90.0
Non kidney failure (actual)	2	8	80.0
Overall Accuracy			85.0

TABLE IV. CLASSIFICATION RESULTS OF TRANSIENT SECURITY ASSESSMENT ON TRAIN SET AND TEST SET [40]

	Train Set					Test Set			
	samples	MLP	LVQ	PNN	ARTMAP	MLP	LVQ	PNN	ARTMAP
		440	440	440	440	40	40	40	40
IEEE 14 Bus	CA(%)	72.7	92.5	100.0	100.0	95.0	85.0	97.5	92.5
	MSE	0.273	1.275	0.000	0.000	0.05	0.450	0.025	0.075
IEEEb30 Bus	samples	2242	2242	2242	2242	52	52	52	52
	CA(%)	91.882	90.187	99.732	100.000	96.154	90.385	90.385	84.615
	MSE	0.081	0.268	0.003	0.000	0..039	0.096	0.096	0.154
IEEE 57 Bus	samples	1687	1687	1687	1687	77	77	77	77
	CA(%)	78.601	95.791	100.00	100.0	84.416	90.909	98.701	96.104
	MSE	0.214	1.548	0.000	0.000	0.156	0.766	0.013	0.039

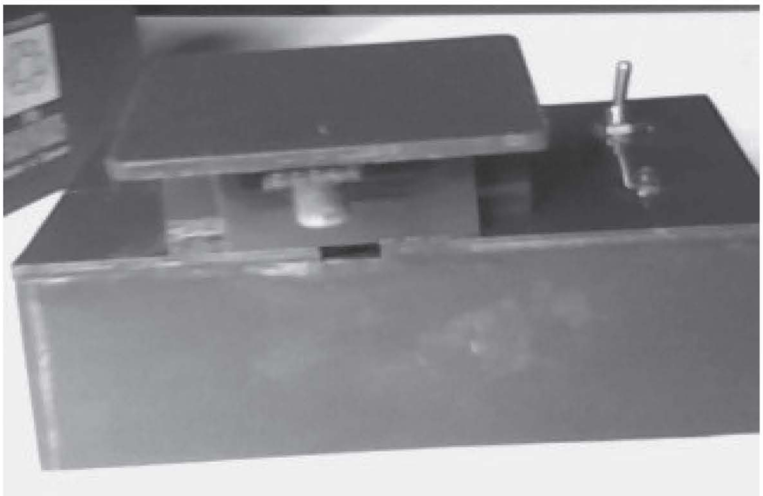

Fig. 3. Real-time Implementation of Electronic Circuit for Food and Tooth Decay Sample Classification

IV. CONCLUSION

The paper has examined the applicability of neural networks in engineering research and practice. It has been shown that the level of confidence in the results of neural simulations in engineering-based applications is on the increase. As a result, neural networks are expected to become more commonplace in years to come. Results from the authors' research also show that neural simulation results can also be used to realise reliable devices for real-time application in health and food security. One important advantage of the neural approach is the prediction of the performance of the physical system via simulation.

REFERENCES

[1] J. Heikkonen and J. Lampinen, "Building Industrial Applications with Neural Networks," Helsinki: Laboratory of Computational Engineering, Helsinki University of Technology.

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bulletin of Mathcmatical Biophysics, vol. 5, pp. 115-133, 1943.

[3] N. Weiner, Cybernetics. Cambridge, Massachusetts: MIT Press, 1961.

[4] D. O. Hebb, The Organization of Behaviour. New York: Wiley, 1949.

[5] B. Widrow, "DARPA Neural Network Study," Armed Forces Communication and Electronics Association, Fairfax, VA1988.

[6] M. L. Minsky and S. Papert, Perceptrons: An Introduction to Computational Geometry. Cambridge, MA: MIT Press, 1969.

[7] D. B. Parker, "A comparison of algorithms for neuron-like cells " in Neural Networks for Computing, J. S. Denker, Ed. New York: American Institute of Physics, 1986, pp. 328-332.

[8] D. D. Rumelhart, B. Widrow, and M. A. Lehr, "The basic ideas in neural networks," Commun. ACM, vol. 37, pp. 89-92, 1994.

[9] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," in The PDP Research Group, D. Rumelhart and J. McClelland, Eds. Cambridge, MA: MIT Press, 1986.

[10] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive network," Complex Syst., vol. 2, pp. 323-354, 1988.

[11] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in Nat. Acad. Sci., 1982, pp. 2446-2556.

[12] N. K. Kasabov, Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering. Cambridge, MA: MIT Press, 1996.

[13] A. Rabaai and R. Kotaru, "Online identification and control of a DC motor using learning adaptation of neural networks," IEEE Trans. Ind. Applicat., vol. 36, pp. 936-941, 2000.

[14] G. K. Venayagamoorthy and R. G. Harley, "Experimental studies with a continually-online-trained artificial neural network controller for a turbo generator," in Int. Joint. Conf. Neural Networks, Washington DC, 1999, pp. 2160-2163.

[15] M. Y. Chow, P. M. Mangum, and S. O. Yee, "A neural network approach to real-time condition monitoring of induction motors " IEEE Trans. Ind. Electron., vol. 38, pp. 449-452, 1991.

[16] M. Y. Chow, R. N. Sharpe, and J. C. Hung, "On the application and design of artificial neural networks for motor fault detection - Part II," IEEE Trans. Ind. Electron., vol. 40, pp. 191-196, 1993.

[17] F. Filipetti, G. Franceschini, C. Tassoni, and P. Vas, "Recent developments of induction motor drives fault diagnosis using AI techniques," IEEE Trans. Ind. Electron., vol. 47, pp. 996-1004, 2000.

[18] B. Burton and R. G. Harley, "Reducing the computational demands of continually online-trained artificial neural networks for system identification and control of fast processes," IEEE Trans. Ind. Applicat., vol. 34, pp. 591-595, 1998.

[19] C. Y. Huang, T. C. Chen, and C. L. Huang, "Robust control of induction motor with a neural network load torque estimator and a neural network identification," IEEE Trans. Ind. Electron., vol. 46, pp. 992-997, 1999.

[20] M. Wishart and R. G. Harley, "Identification and control of induction machines using artificial neural networks," IEEE Trans. Ind. Applicat., vol. 31, pp. 614-619, 1995.

[21] M. J. Er and K. C. Liew, "Control of adept one SCARA robot using neural networks," IEEE Trans. Ind. Electron., vol. 44, pp. 763-766, 1997.

[22] H. Hashimoto, T. Kubota, M. Sato, and F. Harashima, "Visual control of robotic manipulator based on neural networks," IEEE Trans. Ind. Electron., vol. 39, pp. 411-496, 1992.

[23] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma, and Y. Uchikawa, "Trajectory control of robotic manipulators using neural networks," IEEE Trans. Ind. Electron., vol. 38, 1991.

[24] A. T. Vemuri and M. M. Polycarpou, "Neural network-based robust fault diagnosis in robotic systems," IEEE Trans. Neural Networks, vol. 8, pp. 1411-1420, 1997.

[25] D. D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 534-536, 1986.

[26] N. V. Bhat, P. A. Minderman, T. McAvoy, and N. S. Wang, "Modeling chemical process systems via neural computation," IEEE Contr. Syst. Mag., vol. 10, pp. 25-29, 1990.

[27] A. Sirinivasan and C. Batur, "Hopfield/ART-I neural network-based fault detection and isolation," IEEE Trans. Neural Networks, vol. 5, pp. 891-899, 1994.

[28] J. R. Whiteley, J. F. Davis, A. Mehrotra, and S. C. Ahalt, "Observations and problems applying ART2 for dynamic sensor pattern interpretation," IEEE Trans. Syst., Man, Cybern. A, vol. 26, pp. 424-436, 1996.

[29] B. Krose and P. v. d. Smagt, An introduction to neural networks, 8th ed. Amsterdam: University of Amsterdam, 1996.

[30] S. Sardy and L. Ibrahim, "Experimental medical and industrial applications of neural networks to image inspection using an inexpensive personal computer," Opt. Eng., vol. 35, pp. 2183-2186, 1996.

[31] M. R. G. Meireles, P. E. M. Almeida, and M. G. Simoes, "A comprehensive review for industrial applicability of artificial neural networks," IEEE Trans. Ind. Electron., vol. 50, pp. 585-599, 2003.

[32] I. K. Attieh, A. V. Gribok, J. W. Hines, and R. E. Uhrig, "Pattern recognition techniques for transient detection to enhance nuclear reactors' operational safety," in Proc. 25th CNS/CNA Ann. Student Conf., Knoxville, 2000.

[33] G. Dini and F. Failli, "Planning grasps for industrial robotized applications using neural networks," Robot. Comput. Integr. Manuf., vol. 16, pp. 452-463, 2000.
[34] D. Flynn, S. McLoone, G. W. Irwin, M. D. Brown, E. Swidenbank, and B. W. Hogg, "Neural control of turbogenerator systems," Automatica, vol. 33, pp. 1962-1971, 1997.
[35] Y. Dote, M. Strefezza, and A. Suyitno, "Neuro fuzzy robust controllers for drive systems," in Proc. IEEE Int. Symp. Industrial Electronics, 1993, pp. 230-240.
[36] L. E. B. Silva, B. K. Bose, and J. O. P. Pinto, "Recurrent neural network-based implementation of a programmable cascaded low-pass filter used in stator flux synthesis of vector-controlled induction motor drive," IEEE Trans. Ind. Electron., vol. 46, pp. 662-665, 1999.
[37] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural network," Bern: University of Bern, Switzerland, pp. 1-12.
[38] I. O. Essiet, "Neural analysis of hedonic tone, intensity and detectability of ammonia odorant for the identification of tooth decay," Research Journal in Engineering and Applied Sciences, vol. 3, pp. 29-33, 2014.
[39] S. Haykin, Neural networks: A comprehensive foundation, 2nd ed. New York: Prentice-Hall, 1995.
[40] M. T. Haque and A. M. Kashtiban, "Application of neural networks in power systems: A review," Int. Journal of Electrical, Computer, Electronics and Communications Engineering, vol. 1, 2007.
[41] S. Kalyani and K. S. Swarup, "Study of neural network models for security assessment in power systems," Int. Journal of Research and Reviews in Applied Sciences, vol. 1, 2009.