Covenant Journal of Informatics and Communication Technology (CJICT) Vol. 4, No. 1, June, 2016

Evaluation of Manufactured Product Performance Using Neural Networks

Ima Okon Essiet

Dept. of Electrical Engineering, Bayero University, Kano, Nigeria
imaessiet82@gmail.com

Abstract: This paper discusses some of the several successful applications of neural networks which have made them a useful simulation tool. After several years of neglect, confidence in the accuracy of neural networks began to grow from the 1980s with applications in power, control and instrumentation and robotics to mention a few. Several successful industrial implementations of neural networks in the field of electrical engineering will be reviewed and 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.

1. Introduction

The parallel processing capability of the human brain is the most powerful characteristic which has been applied to the design of neural networks. Parallelism is defined as the brain's ability to respond to an external stimulus in several different ways. Neural networks are designed to find the best-fitting solution to a problem given a number of specific 'objectives'. The objectives can be selected to minimize some cost, for example, or enhance utilization of limited resources. Neural networks are highly parallel, dynamic systems capable of performing data manipulations by means of their state response to input information (Lampinen and Heikkonen). Neural networks have been utilized in order to implement 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, neural networks can be described in various forms based on the arrangement and interaction between the layers of neurons in the network. Some of these neural interactions may be cyclical or feedback, feed forward or recurrent in nature (to mention a few). 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.

2. 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 (McCulloch & Pitts, 1943). 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 (Weiner, 1961). 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 (Hebb, 1949). 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. The adaptive linear element was proposed by Bernard Widrow and Ted Hoff 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 to eliminate echoes in analogue telephony. This was the first documented application of neural networks in industry (Widrow, 1988).

The growing popularity of neural networks particularly between the 1950s and early 60s was thwarted when Marvin Minsky and Seymour Papert published a book titled *Perceptrons* in 1969. In it, they proved mathematically that the perceptron could not be applied in problems that presented as nonseparable logic functions (Minsky & Papert, 1969). 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 (Parker, 1986; Rumelhart, Widrow, & Lehr, 1994). During this period, various forms of the neural network were realised such as the Boltzmann Machine by Sejnowski, Hinton and Ackley (Hinton & Sejnowski, 1986), radial basis function (RBF) networks by Broomhead and Lowe (Broomhead & Lowe, 1988) and the Hopfield recurrent neural network (Hopfield, 1982).

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. Fig 1 shows engineering applications of neural networks since 1988. As seen from Fig 1, many control and classification processes have been successfully implemented using a number of neural structures (in some cases a combination of structures). The figure also shows that 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 (Kasabov, 1996). 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.

3. 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.

3.1 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 neural structure in which the output of each element or node is connected to nodes in forward layers in series without any intra-layer connections. This structure was introduced to overcome the limitations of the single-layer perceptron as revealed in *Perceptrons*. 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 (Rabaai & Kotaru, 2000; Venyagamoorthy & Harley, 1999), induction motor fault diagnosis (Chow, Mangum, & Yee, 1991) (Chow, Sharpe, & Hung, 1993; Filipetti, Franceschini, Tassoni, & Vas, 2000), induction motor control (Burton & Harley, 1998; Huang, Chen, & Huang, 1999; Wishart & Harley, 1995), feedback control (Er & Liew, 1997; Hashimoto, Kubota, Sato, & Harashima, 1992; Ozaki, Suzuki, Furuhashi, Okuma, & Uchikawa, 1991) and fault diagnosis of robotic systems (Vemuri & Polycarpou, 1997). The topology of the multi-layer perceptron is depicted in Figure 2.

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 (Rumelhart, Hinton, & Williams, 1986). 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 used in the spectral analysis of periodic waveforms used to describe engineering phenomena (Bhat, Minderman, McAvoy, & Wang, 1990). Fault detection and isolation in realtime engineering systems has also been achieved using Hopfield networks (Sirinivasan & Batur, 1994).

ARTnet: The adaptive resonance theory 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 (Whiteley, Davis, Mehrotra, & Ahalt).

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 (Krose & Smagt, 1996). 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 near to each other in the input matrix must be matched to output units which are also close to each other. Assume the input vector is $\boldsymbol{I}^{\mathbf{N}}$ and a sample $\mathbf{s}(t)$ is presented to the network. The winning node $\mathbf{u}$ is adapted using the Equation (1):

$$w_k(t+1) = w_k(t) + \eta d(k,u)(s(t) - w_k(t)) \quad (1)$$

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 (Sardy & Ibrahim, 1996).

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 (Meireles, Almeida, & Simoes, 2003). PNNs have been used in the identification of transients in nuclear power plants (Attieh, Gribok, Hines, & Uhrig, 2000).

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 (Meireles, Almeida, & Simoes, 2003). 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 (Dini & Failli, 2000), for generator system control (Flynn, McCloone, & Irwin, 1997), power electronic drives and digital signal processors (Dote, Strefezza, & Suyitno, 1993).

Polynomial Neural Networks: Polynomial networks are termed ‘plastic networks’ because their structure is determined during network training. As a result, no two applications can have the same polynomial network structure. The flexible nature of polynomial networks makes them useful in control applications in which the dimension of the plant is not known. Polynomial networks have been used to implement filter design (Silva, Bose, & Pinto, 1999).

3.2 Data Requirements

The data requirements for successful implementation of a neural network 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 used in data recognition applications such as handwritten character recognition (Frinken, Fischer, Manmatha, & Bunke) and the results have been quite encouraging.

- 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 selected such that the network's overall behaviour remains stable throughout the training process i.e. the neuron outputs are as close as possible to their individual activation levels (Krose & Smagt, 1996). There are several types of activation functions. The most common are the linear, semi-linear and sigmoid functions. The network's transfer function is a ratio of the output function to its activation function (Essiet, 2014).
- Interconnected Weights: These connect the layers of the neural network and are used to adjust layer output(s). Assume two interconnected neurons *m* and *n* have a weight connection W_{mn} such that $\{W_{mn} \in 0.......1\}$. The significance of weight connections is determined by the proximity of their absolute value to unity.
- Data Flow: This specifies the direction in which information flows between network layers. In some neural structures, data flow is from input to output in one forward direction, while in others, it may be fed back from output to input or within the same layer.
- Input Signals: Input signals are of various types such as binary, bipolar or continuous. Binary inputs can be in either one of two states, while bipolar inputs are concerned with the direction of the inputs. In other words, inputs can be in one direction or in the exact opposite. Inputs represented by a string of real values constitute continuous inputs.

Over the years, the MLP has

been the most commonly used NN structure. As at 1995, the ratio of neural network utilisation was: MLPs recorded 81%; Hopfield was 5% and Kohonen, 8% (Haykin, 1995). Figure 3 shows the utilisation in power engineering as at 2007. The MLP utilisation is about 33%, which is the highest percentage.

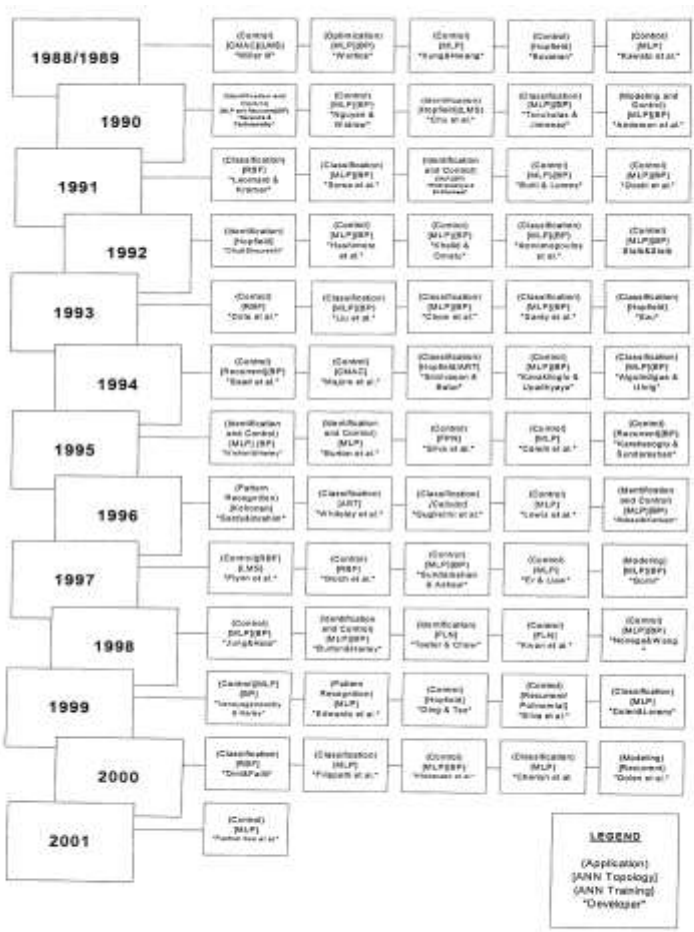

Figure 1. Documented Applications of Neural Networks in Industry since 1980s (Meireles, Almeida, & Simoes, 2003)

4. 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.

4.1 Power Systems Security Assessment

Security assessment with regard to power systems is defined as the process of determining the presence and extent of interference to the normal operation of power systems. It also involves determining system stability in response to certain external perturbations in either its present or future state (Kalyani & Swarup, 2009). 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 (Kalyani & Swarup, 2009). 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 simulated in IEEE 14 bus, 30 bus and 57 bus test systems respectively (Haque & Kashtiban, 2007). 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 data test samples

DO_k= target output

AO_k= actual neural network output as given by the neural network

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

4.2 Neural Simulation of Food Classification and Tooth Decay Sensor Circuits

*T*his 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 Figure 4.

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 I. Classification of Neural Network Structures based on their application areas (Meireles, Almeida, & Simoes, 2003)

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

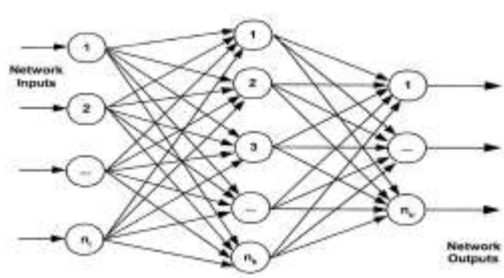

Figure 2. Structure of the MLP (Meireles, Almeida, & Simoes, 2003)

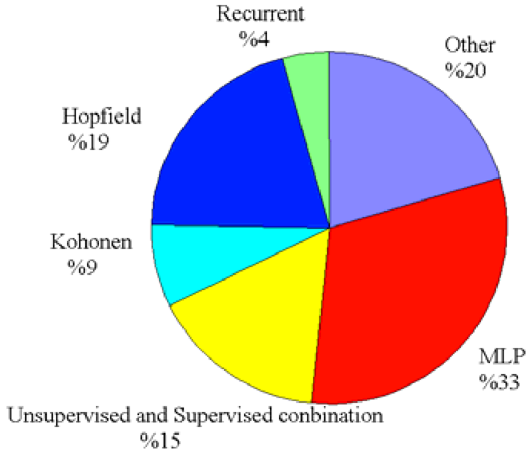

Figure 3. Proportional usage of NN types in Security Assessment of Power Systems (Haque & Kashtiban, 2007)

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

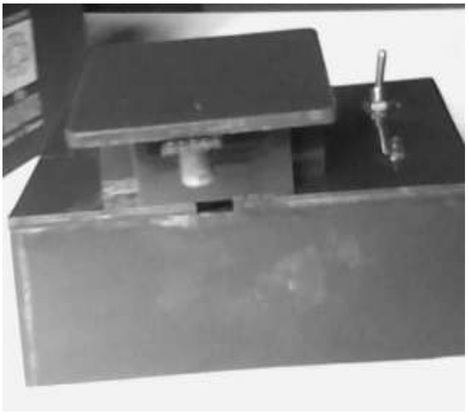

Figure 4. Real-Time Implementation of Electronic Circuit for Food and Tooth Decay Sample Classification

5. 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

Attieh, I., Gribok, A., Hines, J., & Uhrig, R. (2000). Pattern Recognition Techniques for Transient Detection to Enhance Nuclear Reactors. *Proc. 25th CNS/CNA Ann. Student Conf.* Knoxville.

Bhat, N., Minderman, P., McAvoy, T., & Wang, N. (1990). Modeling Chemical Process Systems via Neural Computation. *IEEE Contr. Syst. Mag.*, 26-28.

Broomhead, D., & Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Network. *Complex Syst.*, 325-354.

Burton, B., & Harley, R. (1998). Reducing the Computational Demands of Continually Online-trained Artificial Neural Networks for System Identification and Control of Fast Processes. *IEEE Trans. Ind. Electron.*, 591-595.

Chow, M., Mangum, P., & Yee, S. (1991). A Neural Network Approach to Real-time Condition Monitoring of

Induction Motors. *IEEE Trans. Ind. Electron.*, 449-452.

Chow, M., Sharpe, R., & Hung, J. (1993). On the Application and Design of Artificial Neural Networks for Motor Fault Detection - Part II. *IEEE Trans. Ind. Electron.*, 191-196.

Dini, G., & Failli, F. (2000). Planning Grasps for Industrial Robotized Applications using Neural Networks. *Robot. Comput. Integr. Manuf.*, 455-460.

Dote, Y., Strefezza, M., & Suyitno, A. (1993). Neuro Fuzzy Robust Controllers for Drive Systems. *Proc. IEEE Int. Symp. Industrial Electronics*, (pp. 232-239).

Er, M., & Liew, K. (1997). Control of Adept One SCARA Robot using Neural Networks. *IEEE Trans. Ind. Electron*, 764-766.

Essiet, I. (2014). Neural Analysis of Hedonic Tone, Intensity and Detectability of Ammonia Odorant for the Identification of Tooth Decay. *Research Journal in Engineering and Applied Sciences*, 29-33.

Filipetti, F., Franceschini, G., Tassoni, C., & Vas, P. (2000). Recent Developments of Induction Motor Drives Fault Diagnosis using AI Techniques. *IEEE Trans. Ind. Electron.*, 996-1004.

Flynn, D., McCloone, S., & Irwin, G. e. (1997). Neural Control of Turbogenerator Systems. *Automatica*, 1964-1970.

Frinken, V., Fischer, A., Manmatha, R., & Bunke, H. (n.d.). A Novel Word Spotting Method Based on Recurrent Neural Network. 1-10.

Haque, M., & Kashtiban, A. (2007). Applications of Neural Networks in Power Systems: A Review. *Int. Journal of Electrical, Computer, Electronics and Communications Engineering*.

Hashimoto, H., Kubota, T., Sato, M., & Harashima, F. (1992). Visual Control of Robotic Manipulator based on Neural Networks. *IEEE Trans. Ind. Electron.*, 418-492.

Haykin, S. (1995). *Neural Networks: A Comprehensive Foundation.* New York: Prentice Hall.

Hebb, D. (1949). *The Organization of Behaviour*. New York: Wiley.

Hinton, G., & Sejnowski, T. (1986). Learning and Relearning in Boltzmann Machines. In *The PDP Research Group, D.* Cambridge, Massachusetts: MIT Press.

Hopfield, J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Nat. Acad. Sci.*, pp. 2446-2500.

Huang, C., Chen, T., & Huang, C. (1999). Robust Control of Induction Motor with a Neural Network Load Torque Estimator and a Neural

Network Identification . *IEEE Trans. Ind. Electron.*, 992-997.

Kalyani, S., & Swarup, K. (2009). Study of Neural Network Models for Security Assessment in Power Systems . *Int. Journal of Research and Reviews in Applied Sciences*.

Kasabov, N. (1996). *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering.* Cambridge, Massachusetts: MIT Press.

Krose, B., & Smagt, P. (1996). *An Introduction to Neural Networks.* Amsterdam: University of Amsterdam.

Lampinen, J. and Heikkonen J. (n.d.). *Building Industrial Applications with Neural Networks*. Helsinki, Finland: Laboratory of Computational Engineering, Helsinki University of Technology.

McCulloch, W., & Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 115-133.

Meireles, M., Almeida, P., & Simoes, M. (2003). A Comprehensive Review for Industrial Applicability of Artificial Neural Networks. *IEEE Trans. Ind. Electron.*, 587-596.

Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry.* Cambridge, Massachusetts: MIT Press.

Ozaki, T., Suzuki, T., Furuhashi, T., Okuma, S., & Uchikawa, Y. (1991). Trajectory Control of Robotic Manipulators using Neural Networks. *IEEE Trans. Ind. Electron.*

Parker, D. (1986). A Comparison of Algorithms for Neuron-like Cells. In J. Denker, *Neural Networks for Computing* (pp. 326-335). New York: American Institute of Physics.

Rabaai, A., & Kotaru, R. (2000). Online Identification and Control of a DC Motor using Learning Adaptation of Neural Networks. *IEEE Trans. Ind. Applicat.*, 936-941.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning Representations by Back-propagation Errors. *Nature*, 534-536.

Rumelhart, D., Widrow, B., & Lehr, M. (1994). The Basic Ideas in Neural Networks. *Commun. ACM*, pp. 89-92.

Sardy, S., & Ibrahim, L. (1996). Experimental Medical and Industrial Applications of Neural Networks to Image Inspection using an Inexpensive Personal Computer. *Opt. Eng.*, 2184-2186.

Silva, L. E., Bose, B. K., & Pinto, J. O. (1999). 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.*, 663-665.

Sirinivasan, A., & Batur, C. (1994). Hopfield/ART-Neural Network-based Fault Detection and Isolation. *IEEE Trans. Neural Networks*, 893-897.

Vemuri, A., & Polycarpou, M. (1997). Neural Network-based Robust Fault-diagnosis in Robotic Systems. *IEEE Trans. Neural Networks*, 1415-1419.

Venyagamoorthy, G., & Harley, R. (1999). Experimental Studies with a Continually-Online-Trained Artificial Neural Network Controller for a Turbo Generator. *Int. Joint. Conf. Neural Networks*, (pp. 2160-2163). Washington DC.

Weiner, N. (1961). *Cybernetics.* Cambridge, Massachusetts: MIT Press.

Whiteley, J., Davis, J., Mehrotra, A., & Ahalt, S. (428-435). Observations and Problems Applying ART2 for Dynamic Sensor Pattern Interpretation. *IEEE Trans. Syst., Man, Cybern. A*, 1996.

Widrow, B. (1988). DARPA Neural Network Study. *Armed Forces Communication and Electronics Association*.

Wishart, M., & Harley, R. (1995). Identification and Control of Induction Machines using Artificial Neural Networks. *IEEE Trans. Ind. Applicat.*, 616-619.