

Research Article

Artificial Neural Network Analysis of Sierpinski Gasket Fractal Antenna: A Low Cost Alternative to Experimentation

Balwinder S. Dhaliwal¹ and Shyam S. Pattnaik²

¹ Guru Nanak Dev Engineering College, Ludhiana, Punjab 141006, India

² National Institute of Technical Teachers' Training and Research, Chandigarh 160019, India

Correspondence should be addressed to Balwinder S. Dhaliwal; bsdhaliwal@ymail.com

Received 29 June 2013; Accepted 9 September 2013

Academic Editor: Ozgur Kisi

Copyright © 2013 B. S. Dhaliwal and S. S. Pattnaik. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Artificial neural networks due to their general-purpose nature are used to solve problems in diverse fields. Artificial neural networks (ANNs) are very useful for fractal antenna analysis as the development of mathematical models of such antennas is very difficult due to complex shapes and geometries. As such empirical approach doing experiments is costly and time consuming, in this paper, application of artificial neural networks analysis is presented taking the Sierpinski gasket fractal antenna as an example. The performance of three different types of networks is evaluated and the best network for this type of applications has been proposed. The comparison of ANN results with experimental results validates that this technique is an alternative to experimental analysis. This low cost method of antenna analysis will be very useful to understand various aspects of fractal antennas.

1. Introduction

Artificial neural networks (ANNs) have been used as efficient tools for modeling and prediction in almost all disciplines. The use of ANN has become widely accepted in antenna design and analysis applications. This is evident from the increasing number of publications in research/academic journals [1–8]. Angiulli and Versaci proposed a technique to evaluate the resonant frequency of microstrip antennas using neuro-fuzzy networks [1]. The use of ANN for the design of rectangular patch antenna is explained in [2]. Applications of ANN in various types of antennas and antenna arrays are explained in [3]. Neog et al. [4] have used a tunnel based ANN for the parameter calculation of the wideband microstrip antenna. Lebbbar et al. [5] employed a geometrical methodology based ANN for the design of a compact broadband microstrip antenna. In [6] the authors proposed an ANN to predict the input impedance of a broadband antenna as a function of its geometric parameters. Guney and Sarikaya [7] presented a hybrid method based on a combination of ANN and fuzzy inference system to calculate simultaneously the resonant frequencies of various microstrip antennas of regular geometries. An equilateral

triangular microstrip antenna has been designed using a particle swarm optimization driven radial basis function neural networks by [8]. However, the use of ANN in analysis & design of fractal antennas is at very early stage. A limited number of literatures are available in this field of antennas [9–12]. In this paper, the performance of three different ANNs on Sierpinski gasket fractal antenna analysis is investigated by means of two aspects: mean absolute error (MAE) and coefficient of correlation.

The experimental analysis of antennas involves costly setup which include anechoic chamber and equipment like vector network analyzer, synthesized signal generator, pattern recorder, and so forth, in addition to other equipment required for antenna fabrication. Availability of such resources is limited in academic institutions due to limited budget. Thus, the proposed technique is a very low cost method as it requires only a computer and software. So this method may be used as part of the lab experiments in undergraduate and postgraduate classes. The method is explained by taking an example of Sierpinski gasket fractal antenna for analysis.

The term fractal was originally coined by Mandelbrot to describe a family of complex shapes that possess an inherent

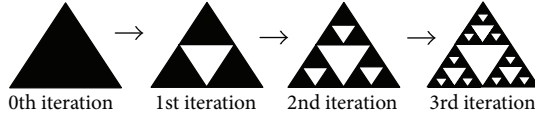


FIGURE 1: First three iterations of Sierpinski gasket antenna.

selfsimilarity in their geometrical structure [15]. Fractals represent a class of geometries with very unique properties that have attracted antenna designers in recent years. The general concept of fractals can be applied to develop various antenna elements and such antennas are known as fractal antennas [16]. Applying fractals for antenna elements result in smaller resonant antennas that are multiband and may be optimized for gain. The advantages of fractal antennas include (i) miniaturization and space-filling, (ii) multiband performance, (iii) input impedance matching, (iv) efficiency and effectiveness, and (v) improved gain and directivity [17].

Various fractal geometries have been explored by researcher in past two decades. However, the Sierpinski gasket fractal antenna geometry has been investigated most widely than any other geometry since its introduction in 1998 by Puente-Baliarda et al. [13]. The Sierpinski triangle, also called the Sierpinski gasket, is a fractal geometry named after the Polish mathematician Waclaw Sierpinski who described it in 1915. The first three iterations of Sierpinski gasket are shown in Figure 1. The first iteration gasket is constructed by subtracting central inverting triangle from a main triangle shape. After the subtraction three equal triangles remain on the structure, each one being half of the size of the original. If the same subtraction procedure is repeated on remaining triangles, the 2nd iteration is obtained and similarly if iteration is carried out an infinite number of times, the ideal Sierpinski gasket is obtained [18]. The ideal Sierpinski gasket is a selfsimilar structure and each one of its three main parts has exactly the same shape as that of the whole object but reduced by a factor of two [13].

The Sierpinski triangle can also be generated by the use of the iterated function system (IFS) which is generally used to construct selfsimilar fractal shapes. The IFS is based on a set of affine transforms and if the side length “ s ” of the base equilateral triangle is known, then the following iterated function system gives the selfsimilar shape of Sierpinski gasket [18]:

$$\begin{aligned} f_1(x) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} x, \\ f_2(x) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \\ f_3(x) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} x + \begin{bmatrix} 0.250 \\ 0.433 \end{bmatrix}. \end{aligned} \quad (1)$$

The Sierpinski antenna is a multiband antenna and the number of bands depends on the number of iterations “ n ” [13]. The base triangular shape which is also called 0th iteration has single fundamental resonance frequency. The first fractal iteration has two resonant frequencies and so on.

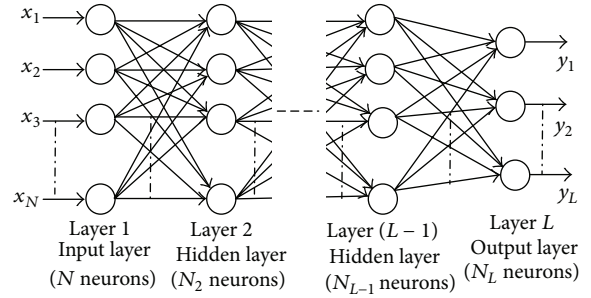


FIGURE 2: Basic structure of MLPNN.

Also, the actual value of resonant frequencies depends on dielectric constant “ ϵ_r ” and on the thickness “ t ” of the substrate and side length “ s .” So to determine the resonant frequencies f_n in a particular iteration, the values of ϵ_r , t , s and number of iterations “ n ” must be known. An expression for predicting the resonant frequency of this antenna has been proposed in 2008 by Mishra et al. [14].

The rest of the paper is organized as follows. Section 2 briefly describes the ANN types which are being investigated. Section 3 describes the details of ANN models, results, and the performance comparison. The conclusion is presented in Section 4.

2. Artificial Neural Networks: Brief Description

The brief of three types of networks, namely, multilayer perceptron neural network (MLPNN), radial basis function neural network (RBFNN), and general regression neural network (GRNN) which are used in this work, are described in the following sections.

2.1. Multilayer Perceptron Neural Network. Multilayer perceptron neural network (MLPNN) is a widely used neural network structure in antenna applications. It consists of multiple layers of neurons that are connected in a feed forward manner [6]. The base structure of an MLPNN is shown in Figure 2.

Figure 2 depicts that the network has L layers. The first layer is the input layer and the last layer (L th layer) is the output layer. The other layers, that is, layers from 2 to $L - 1$ are hidden layers. Each layer has a number of neurons [19]. In the input and output layers, the number of neurons are equal to the number of inputs and outputs, respectively. The number of neurons in the hidden layers is chosen using a trial and error process so that accurate output is obtained. Usually, the MLPs with one or two hidden layers are commonly used for antenna applications. Each neuron processes the inputs to produce output using a function called activation function. Input neurons normally use a relay activation function and simply relay (pass) the external inputs to the hidden layer neurons. The most commonly used activation function for hidden layer neuron is the sigmoid function defined as

$$\sigma(r) = \frac{1}{1 + e^{-r}}. \quad (2)$$

The connection between the units in subsequent layers has an associated weight which is computed during training using error backpropagation algorithm [20]. The backpropagation learning algorithm consists of two steps. In the first step, the input signal is passed forward through each layer of the network. The actual output of the network is calculated and this output signal is subtracted from desired output signal to generate an error signal. In the second step, error is fed backward through the network from the output layer, through the hidden layers. The synaptic weights between each layer are updated based on the computational error. These two steps are repeated for each input-output pair of training data and process is iterated for a number of times until the output value converges to a desired solution [21]. The details of MLPNN can be further seen from [22–25].

2.2. Radial Basis Function Neural Networks. Radial basis function neural network (RBFNN) has several special characteristics like simple architecture and faster performance [26]. It is a special neural network with a single hidden layer. The hidden layer neurons use radial basis function to apply a nonlinear transformation from the input space to the hidden space [27]. The nonlinear basis function is a function of the normalized radial distance between the input vector and the weight vector. The most widely used form of radial basis function is the Gaussian function given by [26]

$$\Phi_j(x) = \exp\left(\frac{-\|x - \mu_j\|^2}{2\sigma_j^2}\right), \quad (3)$$

where x is d -dimensional input vector with elements x_i and μ_j is vector determining the centre of basis function Φ_j and has elements μ_{ji} . The symbol σ_j denotes width parameter whose value is determined during training. The RBFNN involves two-stage training procedure. In the first stage, the parameters governing the basis functions, that is, centers (μ_j) and widths (σ_j), are determined using the relatively fast, unsupervised methods. The unsupervised training methods use only the input data and not the target data. In second stage of training, the weights of the output layer are determined [27]. For these networks, output layer is a linear layer, so fast linear supervised methods are used. Due to this reason these networks have faster performances [26]. The number of neurons in all layers of RBFNN, that is, input, output, and hidden layers, is determined by the dimensions and number of input-output data sets. The basic structure of RBFNN is shown in Figure 3. RBFNN can be explored in detail from [25–29].

2.3. General Regression Neural Network. General regression neural network (GRNN) introduced by Specht in 1991 is a one-pass learning algorithm network with a highly parallel structure [30]. It is a memory based network that provides estimates of continuous variables and converges to underlying linear or nonlinear regression surface. Nonlinear regression analysis forms the theoretical base of GRNN. If joint probability density function of random variables x and y is $f(x, y)$ and measured value of x is X , then the regression

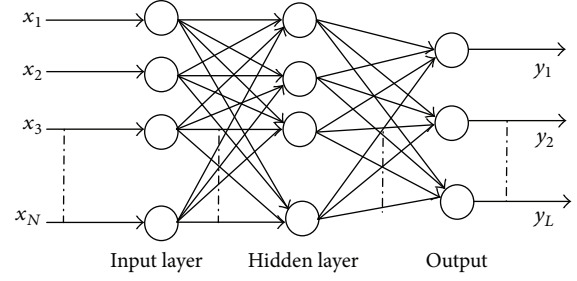


FIGURE 3: Basic structure of RBFNN.

of y with respect to X (which is also called conditional mean) is [31]

$$\hat{Y}(X) = E\left[\frac{y}{X}\right] = \frac{\int_{-\infty}^{\infty} y \cdot f(X, y) dy}{\int_{-\infty}^{\infty} f(X, y) dy}, \quad (4)$$

where $\hat{Y}(X)$ is the prediction output of Y when input is X .

For GRNN, Nose-Filho et al. [32] have shown that $\hat{Y}(X)$ can be calculated by

$$\hat{Y}(X) = \frac{\sum_{i=1}^n Y^i \exp(-D_i^2/2\sigma^2)}{\sum_{i=1}^n \exp(-D_i^2/2\sigma^2)}, \quad (5)$$

where X^i and Y^i are sample values of random variables x and y , n is number of samples in training set, σ is smoothing parameter, and D_i^2 is the Euclidean matrix defined by

$$D_i^2 = (X - X^i)^T (X - X^i). \quad (6)$$

The estimate $\hat{Y}(X)$ can be visualized as a weighted average of all of the observed values, Y_i , where each observed value is weighted exponentially according to its Euclidean distance from X .

The basic structure of GRNN is shown in Figure 4. The input units are merely distribution units, which provide all of the inputs (scaled measurement variables) to all of the neurons in the second layer called pattern units. The number of neurons in input layer and output layer of GRNN is equal to the dimensions of the input data and output data, respectively. The pattern unit consists of neurons equal to number of sample in training set and each pattern unit is dedicated to one cluster centre. When a new vector, X , is entered into the network, it is subtracted from the stored vector representing each cluster center. The sum of squares or the absolute values of the differences is calculated and fed into a nonlinear activation function which is generally an exponential function.

The outputs of pattern units are passed on to third layer neurons known as summation units. There are two types of neurons in summations units: numerator neurons whose number is equal to output dimensions and one denominator neuron. The first type generates an estimate of $f(x)K$ by adding the outputs of pattern units weighted by the number

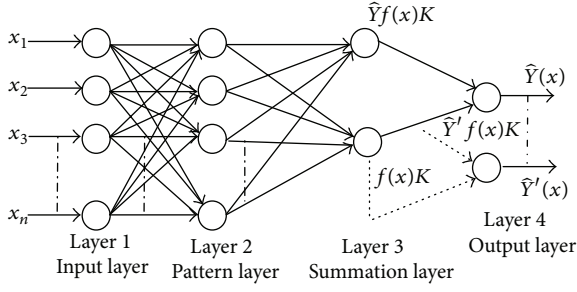


FIGURE 4: Basic structure of GRNN.

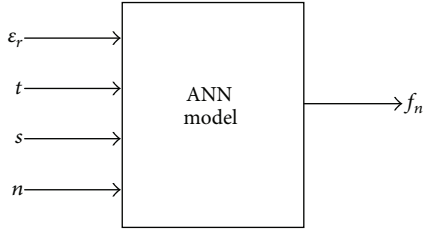


FIGURE 5: Proposed ANN Model.

of observations each cluster centre represents. The second type generates estimate of $\hat{Y}f(x)K$ and multiplies each value from a pattern unit by the sum of samples associated with cluster center X^i . K is a constant determined by the Parzen window. The desired estimate of Y is yielded by output unit by dividing $\hat{Y}f(x)K$ by $f(x)K$ [32]. The detailed explanation of GRNN can be seen from [30–32].

3. The Proposed ANN Analysis of the Sierpinski Gasket Fractal Antenna

In presented work, three above discussed ANN models have been implemented and compared for Sierpinski gasket fractal antenna. In each model, the parameters dielectric constant “ ϵ_r ,” the thickness “ t ” of the substrate, side length “ s ” and the number of iterations “ n ” of the antenna are taken as inputs and the resonant frequency (f_n) is taken as outputs. A set of 45 values of inputs along with desired output values are taken for training. The block diagram of the proposed models is depicted in Figure 5. Thus, the networks models have four inputs and one output with intermediate hidden layers. The parametric details of the models are shown in Figure 5.

3.1. Multilayer Perceptron Neural Network Parameters. For the training of MLPNN, 4 input neurons, 15 neurons in hidden layer, and 1 output neuron are used. The *trainlm* function is used as training function and learning rate is selected as 0.2. *Trainlm* is a network training function that updates weight and bias values according to the Levenberg-Marquardt algorithm. The architecture used for the analysis is shown in Figure 6.

3.2. Radial Basis Function Neural Networks Parameters. For the training of RBFNN, the value of spread constant has been

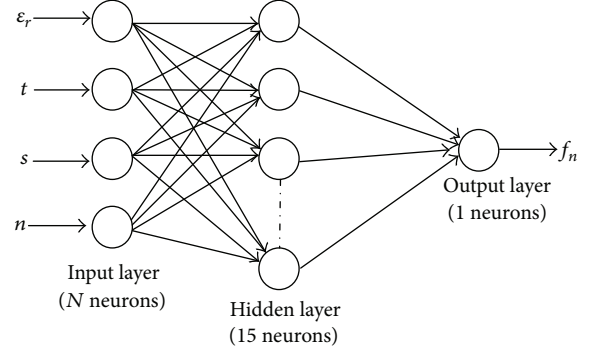


FIGURE 6: Architecture of MLPNN.

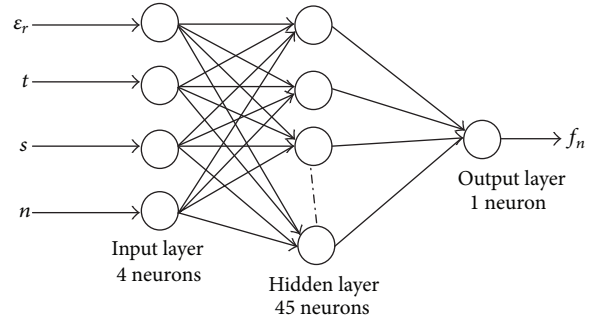


FIGURE 7: Architecture of RBFNN.

selected as 1.05. The value of spread constant should be large enough so that several radial basis neurons always have fairly large outputs at any given moment. This makes the network function smoother and results in better generalization for new input vectors occurring between input vectors used in the design. However, for very large value of spread constant, each neuron is effectively responding in the same large area of the input space. As there are 45 sets of training data so number of neurons in hidden layer is 45. The detailed architecture is depicted in Figure 7.

3.3. General Regression Neural Network Parameters. For the training of GRNN, the value of spread constant has been selected as 0.85. In GRNN, the transfer function of first hidden layer is radial basis function. For small spread constant, the radial basis function is very steep so the neuron with the weight vector closest to the input will have a much larger output than other neurons. The network will tend to respond with the target vector associated with the nearest design input vector. For large values of spread constant the radial basis function's slope becomes smoother and several neurons can respond to an input vector. The network then acts as if it is taking a weighted average between target vectors whose design input vectors are closest to the new input vector. As in case of RBFNN, the number of neurons in hidden layer is 45. The architecture is shown in Figure 8.

The trained ANN models are used to predict resonant frequencies (f_0, f_1, f_2, f_3, f_4) of 4th iteration Sierpinski gasket antenna with parameters $\epsilon_r = 2.5$, $s = 102.7683$ mm,

TABLE 1: Simulation results of all three ANN models. Absolute error calculated by subtracting from experimental results.

Resonant frequency	Experimental results (GHz) [13]	Theoretical results [14]		MLPNN results		RBFNN results		GRNN results	
		Output (GHz)	Absolute error	Output (GHz)	Absolute error	Output (GHz)	Absolute error	Output (GHz)	Absolute error
f_0	0.520	0.5140	0.0060	0.5122	0.0078	0.5123	0.0077	0.8922	0.3722
f_1	1.740	1.7420	0.0020	1.7362	0.0038	1.7360	0.0040	1.9097	0.1697
f_2	3.510	3.4840	0.0260	3.4712	0.0388	3.4717	0.0383	3.9282	0.4182
f_3	6.950	6.9680	0.0180	6.9437	0.0063	6.9434	0.0066	7.6243	0.6743
f_4	13.89	13.9340	0.0440	13.8865	0.0035	13.8867	0.0033	11.8325	2.0575

TABLE 2: Values of performance measures.

Performance measure	Theoretical results	MLPNN results	RBFNN results	GRNN results
Mean absolute error (MAE)	0.0192	0.0120	0.0120	0.7384
Coefficient of correlation, R	1.0000	1.0000	1.0000	0.9899

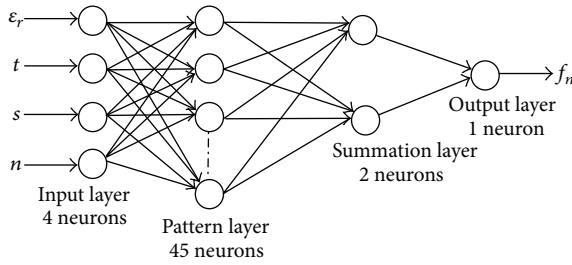


FIGURE 8: Architecture of GRNN.

and $t = 1.588$ mm. ANN results are compared with the experimental and theoretical results as shown in Table 1. The absolute error values are calculated by comparing with measured results of Puente-Baliarda et al. [13]. Also the theoretical results obtained using the expressions of Mishra et al. [14] are provided for comparison.

The performance comparison of different types of ANNs is important to find the best suitable model for specific problem. In recent years, a number of studies are done on performance comparison of ANNs for a range of applications [19, 21, 26, 33–39]. The performance of the proposed models is compared on the basis of two different measures that is, (1) mean absolute error (MAE), which gives the quality of training the model. The smaller the value of MAE of a model better will be its performance, and (2) coefficient of correlation (R) is the other measure of strength of the linear relationship developed by a particular model. The values of R close to 1.0 indicate a good model performance. The quantitative performance is given in Table 2. The values of MAE and R for theoretical results are also shown for comparison purposes.

The comparison between these three algorithms on the basis of MAE shows that the MLPNN and RBFNN have smallest error values and even better than those of theoretical results. It may be seen that the values of R are equal to 1 for MLPNN, RBFNN, and theoretical results and it is sufficiently high for GRNN indicating satisfactory performance by all models.

When both performance criteria are considered together, the most satisfactory model is RBFNN, which has better performance than MLPNN and GRNN and it is even accurate than theoretical method in predicting resonant frequencies.

4. Conclusion

The application of the artificial neural networks method in Sierpinski gasket fractal antenna analysis is evaluated in this work. The performances of the three neural networks are evaluated to realize the possible applications of ANN in fractal antenna design. The obtained ANN results are compared with the published experimental results which show percentage of error less than 1.5% for RBFNN. Due to fast adaptive properties of ANN, the simulation time required is very less. It has been observed that RBFNN method outperforms the theoretical method in accuracy. Among the different neural models compared, the RBFNN is found as the best suitable model for this type of antenna analysis. Thus, ANN approach to Sierpinski fractal antenna analysis seems to be a low cost, accurate, and computationally fast approach. The same concept can be explored for other fractal geometries.

References

- [1] G. Angiulli and M. Versaci, "Resonant frequency evaluation of microstrip antennas using a neural-fuzzy approach," *IEEE Transactions on Magnetics*, vol. 39, no. 3 I, pp. 1333–1336, 2003.
- [2] R. K. Mishra and A. Patnaik, "Designing rectangular patch antenna using the neurospectral method," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 8, pp. 1914–1921, 2003.
- [3] A. Patnaik, D. E. Anagnostou, R. K. Mishra, C. G. Christodoulou, and J. C. Lyke, "Applications of neural networks in wireless communications," *IEEE Antennas and Propagation Magazine*, vol. 46, no. 3, pp. 130–137, 2004.
- [4] D. K. Neog, S. S. Pattnaik, D. C. Panda, S. Devi, B. Khuntia, and M. Dutta, "Design of a wideband microstrip antenna and the use of artificial neural networks in parameter calculation," *IEEE Antennas and Propagation Magazine*, vol. 47, no. 3, pp. 60–65, 2005.

- [5] S. Lebbar, Z. Guennoun, M. Drissi, and F. Riouch, "A compact and broadband microstrip antenna design using a geometrical-methodology-based artificial neural network," *IEEE Antennas and Propagation Magazine*, vol. 48, no. 2, pp. 146–154, 2006.
- [6] Y. Kim, S. Keely, J. Ghosh, and H. Ling, "Application of artificial neural networks to broadband antenna design based on a parametric frequency model," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 669–674, 2007.
- [7] K. Guney and N. Sarikaya, "A hybrid method based on combining artificial neural network and fuzzy inference system for simultaneous computation of resonant frequencies of rectangular, circular, and triangular microstrip antennas," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 659–668, 2007.
- [8] V. S. Chintakindi, S. S. Pattnaik, O. P. Bajpai, and S. Devi, "PSO driven RBFNN for design of equilateral triangular microstrip patch antenna," *Indian Journal of Radio and Space Physics*, vol. 38, pp. 233–237, 2009.
- [9] B. S. Dhaliwal, S. S. Pattnaik, and S. Devi, "Design of Sierpinski gasket fractal patch antenna using artificial neural networks," in *Proceedings of the IASTED International Conference on Antennas, Radar, and Wave Propagation (ARP '09)*, pp. 76–78, July 2009.
- [10] P. H. D. F. Silva, E. E. C. Oliveira, and A. G. D'Assunção, "Using a multilayer perceptrons for accurate modeling of quasi-fractal patch antennas," in *Proceedings of the International Workshop on Antenna Technology (iWAT '10)*, pp. 1–4, Lisbon, Portugal, March 2010.
- [11] P. Arora and B. S. Dhaliwal, "Parameter estimation of dual band elliptical fractal patch antenna using ANN," in *Proceedings of the International Conference on Devices and Communications (ICDeCom '11)*, pp. 1–4, February 2011.
- [12] A. Anuradha, A. Pattnaik, and S. N. Sinha, "Design of custom-made fractal multi-band antennas using ANN-PSO," *IEEE Antennas and Propagation Magazine*, vol. 53, no. 4, pp. 94–101, 2011.
- [13] C. Puente-Baliarda, J. Romeu, R. Pous, and A. Cardama, "On the behavior of the sierpinski multiband fractal antenna," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 4, pp. 517–524, 1998.
- [14] R. K. Mishra, R. Ghatak, and D. Poddar, "Design formula for Sierpinski gasket pre-fractal planar-monopole antennas," *IEEE Antennas and Propagation Magazine*, vol. 50, no. 3, pp. 104–107, 2008.
- [15] D. H. Werner, R. L. Haupt, and P. L. Werner, "Fractal antenna engineering: the theory and design of fractal antenna arrays," *IEEE Antennas and Propagation Magazine*, vol. 41, no. 5, pp. 37–59, 1999.
- [16] J. P. Gianvittorio and Y. Rahmat-Samii, "Fractal antennas: a novel antenna miniaturization technique, and applications," *IEEE Antennas and Propagation Magazine*, vol. 44, no. 1, pp. 20–36, 2002.
- [17] N. S. Song, K. L. Chin, D. B. B. Liang, and M. Anyi, "Design of broadband dual-frequency microstrip patch antenna with modified sierpinski fractal geometry," in *Proceedings of the 10th IEEE Singapore International Conference on Communications Systems (ICCS '06)*, pp. 1–5, Singapore, November 2006.
- [18] D. H. Werner and S. Ganguly, "An overview of fractal antenna engineering research," *IEEE Antennas and Propagation Magazine*, vol. 45, no. 1, pp. 38–57, 2003.
- [19] S. Shrivastava and M. P. Singh, "Performance evaluation of feed-forward neural network with soft computing techniques for hand written English alphabets," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1156–1182, 2011.
- [20] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design—from theory to practice," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 4, pp. 1339–1350, 2003.
- [21] A. Jain and A. Kumar, "An evaluation of artificial neural network technique for the determination of infiltration model parameters," *Applied Soft Computing Journal*, vol. 6, no. 3, pp. 272–282, 2006.
- [22] V. Bourdès, S. Bonnevey, P. Lisboa et al., "Comparison of artificial neural network with logistic regression as classification models for variable selection for prediction of breast cancer patient outcomes," *Advances in Artificial Neural Systems*, vol. 2010, Article ID 309841, 11 pages, 2010.
- [23] K. Y. Huang and K. J. Chen, "Multilayer perceptron for prediction of 2006 world cup football game," *Advances in Artificial Neural Systems*, vol. 2011, Article ID 374816, 8 pages, 2011.
- [24] P. T. Pearson, "Visualizing clusters in artificial neural networks using morse theory," *Advances in Artificial Neural Systems*, vol. 2013, Article ID 486363, 8 pages, 2013.
- [25] S. Haykins, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, USA, 1998.
- [26] P. Singh and M. C. Deo, "Suitability of different neural networks in daily flow forecasting," *Applied Soft Computing Journal*, vol. 7, no. 3, pp. 968–978, 2007.
- [27] H. Sarimveis, P. Doganis, and A. Alexandridis, "A classification technique based on radial basis function neural networks," *Advances in Engineering Software*, vol. 37, no. 4, pp. 218–221, 2006.
- [28] S. Mehrabi, M. Maghsoudloo, H. Arabalibeik, R. Noormand, and Y. Nozari, "Application of multilayer perceptron and radial basis function neural networks in differentiating between chronic obstructive pulmonary and congestive heart failure diseases," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6956–6959, 2009.
- [29] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.
- [30] D. F. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.
- [31] D. Tomandl and A. Schober, "A modified general regression neural network (MGRNN) with new, efficient training algorithms as a robust "black box"-tool for data analysis," *Neural Networks*, vol. 14, no. 8, pp. 1023–1034, 2001.
- [32] K. Nose-Filho, A. D. P. Lotufo, and C. R. Minussi, "Short-term multinodal load forecasting using a modified general regression neural network," *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2862–2869, 2011.
- [33] K. L. Du, A. K. Y. Lai, K. K. M. Cheng, and M. N. S. Swamy, "Neural methods for antenna array signal processing: a review," *Signal Processing*, vol. 82, no. 4, pp. 547–561, 2002.
- [34] R. Shavlt and I. Taig, "Comparison study of pattern-synthesis techniques using neural networks," *Microwave and Optical Technology Letters*, vol. 42, no. 2, pp. 175–179, 2004.
- [35] M. O. Elish, "A comparative study of fault density prediction in aspect-oriented systems using MLP, RBF, KNN, RT, DENFIS and SVR models," *Artificial Intelligence Review*, 2012.
- [36] P. Jeatrakul and K. W. Wong, "Comparing the performance of different neural networks for binary classification problems,"

in *Proceedings of the 8th International Symposium on Natural Language Processing (SNLP '09)*, pp. 111–115, Bangkok, Thailand, October 2009.

- [37] A. Katidiotis, K. Tsagkaris, and P. Demestichas, “Performance evaluation of artificial neural network-based learning schemes for cognitive radio systems,” *Computers and Electrical Engineering*, vol. 36, no. 3, pp. 518–535, 2010.
- [38] M. Mohamadnejad, R. Gholami, and M. Ataei, “Comparison of intelligence science techniques and empirical methods for prediction of blasting vibrations,” *Tunnelling and Underground Space Technology*, vol. 28, no. 1, pp. 238–244, 2012.
- [39] A. Rawat, R. N. Yadav, and S. C. Shrivastava, “Neural network applications in smart antenna arrays: a review,” *AEU—International Journal of Electronics and Communications*, vol. 66, no. 11, pp. 903–912, 2012.

