

# Artificial Neural Network Modeling of Meander Lines for Delay based Applications

Kah Seng Sam, Chan Hong Goay, Nur Syazreen Ahmad, Patrick Goh

**Abstract**—This paper presents the application of artificial neural networks (ANNs) for meander line modeling. In this work, meander lines on microstrips will be investigated to determine a correlation between the physical parameters and the propagation delay of the lines. The simulation of the meander lines is done using the Momentum electromagnetics simulator in Keysight's Advanced Design System (ADS) to generate the S-parameters which will be used in a transient simulation to determine the propagation delay. Neural network models are then created for propagation delay prediction. Finally, both the ADS and ANN results for simulated delay times of meander lines are compared to validate the performance and to justify the proposed method. Results show that the ANN model is able to accurately predict the delay of the meander lines with an accuracy above 99.5% with a speed-up of over 2000x.

**Index Terms:** Artificial Neural Network, Meander Lines, Propagation Delay, S-parameters

## I. INTRODUCTION

Artificial neural networks (ANN) have been widely used in the modeling and simulations of various fields. ANN is an information-processing system that is modeled after the neuronal structure of the human brain. ANN resembles the brain in two aspects: (1) the ability to acquire knowledge through a learning process by the network, and (2) the storage of the knowledge in the interneuron connection strengths known as synaptic weights [1]. ANNs are known to possess many desirable characteristics such as the ability to model high nonlinearities, high parallelism, fault and noise tolerance, and good generalization capabilities. A good generalization is achieved when the neural model can perform well on data that is never used during the training and development process of that model. Neural networks have been used in many microwave applications such as in modeling of microstrip filters [2-5], power amplifiers [6], transmission lines [7], and microstrip antennas [8]. More recently, ANNs have also been used in signal integrity applications such as in predicting the eye-heights and widths in high-speed digital signals [9-14].

Meander lines, also called serpentine lines are a class of

**Revised Version Manuscript Received on August 19, 2019.**

**KahSeng Sam**, School of Electrical and Electronic Engineering, UniversitiSains Malaysia, NibongTebal, 14300, Penang, Malaysia.(email: eepatrick@usm.my)

**Chan Hong Goay**, School of Electrical and Electronic Engineering, UniversitiSains Malaysia, NibongTebal, 14300, Penang, Malaysia. (email: eepatrick@usm.my)

**NurSyazreen Ahmad**, School of Electrical and Electronic Engineering, UniversitiSains Malaysia, NibongTebal, 14300, Penang, Malaysia.(email: eepatrick@usm.my)

**Patrick Goh**, School of Electrical and Electronic Engineering, UniversitiSains Malaysia, NibongTebal, 14300, Penang, Malaysia (email: eepatrick@usm.my)

transmission lines which are often used in printed circuit board to match the delays in multiple lines such as in a clock distribution circuit or in a differential signaling pair. The additional bends in the meander lines introduce an additional delay to the signal which is calculated to match the delay in the other paths for timing synchronizations. However, due to the coupling and crosstalk effects between adjacent bends, the delay of meander lines is often much shorter than the actual delay of a straight line of equal length. Thus it is important to model these effects accurately in order to precisely predict the delay of a meander line.

Recently, the modeling of meander lines has received considerable attention. Multiple studies have been done to investigate the effects of the physical parameters on the actual delay of the meander line [15-18]. While these studies have yielded guidelines on designing meander lines to fit a particular need, these guidelines are heuristics at best and rely on developers experience and engineering know-hows. Furthermore, there is still no improvement in terms of the meander line simulation and analysis.

The common standard in the industry is to perform a full wave electromagnetic (EM) modeling using a computer-aided design (CAD) tool [19]. These tools can give accurate results but they require a lot of time and high computing power. Furthermore, the design process normally involves iterating the design parameters until the final result is realized, often with only slight changes in each step. With the increasing complexity of the EM circuits, there is a need for researchers to look for alternative approximation techniques in modeling and simulation.

In this work, ANNs are used to model the physical relationships between the meander lines and the delay. Once the neural network is properly trained, it can be used repeatedly to provide fast and accurate solutions for microwave design problems. This simplifies the design process and the trained neural network can provide very accurate results many times faster than conventional EM simulations.

## II. MULTILAYER PERCEPTRON

Multilayer perceptron (MLP) is a class of feedforward ANN with an input layer, an output layer, and at least one hidden layer. Fig. 1 shows an example of a three layered MLP



with two input neurons, four hidden neurons and an output neuron. Computations are performed from the input layer to the output layer and each neuron has an associated weight and bias values which are eventually summed up to produce a weighted sum of its inputs which is then passed through an activation function to determine if a neuron should fire. Usually, the neurons of the hidden layers have non-linear activation functions which are continuously differentiable such as the tansig function defined as follows:

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

where  $\sigma(x)$  denotes the activation function and its output ranges from -1 to 1. The input and output neurons on the other hand normally have linear activation functions.

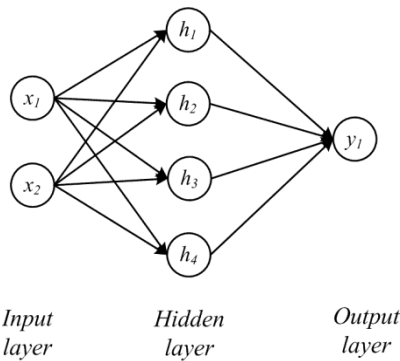


Fig. 1. Example MLP structure

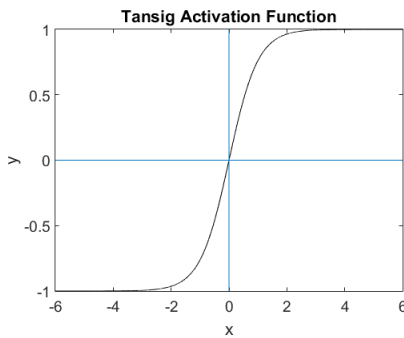


Fig. 2. Tansig activation function

Before a neural network can be used, it must be properly trained. There are a variety of ways to train a neural network such as by using the gradient descent, conjugate gradient, Newton method or Levenberg-Marquardt algorithm. The readers are referred to the reference [20] for more information about computation and learning of neural networks. In this paper, Levenberg Marquardt is used as the training algorithm. The neural network weights are initialized using the Nguyen-Widrow method [21]

One of the most important parts during neural network design is to decide the number of hidden neurons. If the number of neurons in the hidden layer is too small, the network may not be powerful enough to meet the desired requirements. This problem is called underfitting and it can be indicated by poor training and validation performances. On the other hand, a large number of hidden neurons can cause very long training and recall time [22]. Also, a neural network with too many hidden neurons is prone to overfitting,

where it is able to perform very well on training data but has large errors on testing data [23]. The overfitting problem can be indicated by good training performance but poor validation performance. A good learning is achieved when the neural network has good training and validation performances. Hence, there is a tradeoff in the number of hidden neurons to achieve a correct balance between performance and simulation cost. In this work, an iterative approach is used to find the optimum number of hidden neurons. First,  $N_{trial}$  neural models with number of hidden neurons  $H = H_{min}$  (typically 1) are created. The process is then repeated by increasing  $H$  by 1 until  $H = H_{max}$ ,

$$H_{max} = \text{ceil}\left(\frac{(N_{train} - 1)(O)}{I + O + 1}\right) - 1 \quad (2)$$

where  $N_{train}$  is the total number of training samples,  $O$  is the number of output neurons,  $I$  denotes the number of input neurons and  $\text{ceil}(\cdot)$  is a rounding towards positive infinity.

For each network, a test error is calculated and the neural model with the lowest validation error is selected as the final design. Typically, each neural network training session starts with different initial conditions due to the randomness present in the weight initialization. Hence, it is important to create multiple neural models so that a good generalization can be found. The pseudo-code for determining the number of hidden neurons is shown below:

```

for  $H = H_{min} : dH : H_{max}$ 
  for index = 1 :  $N_{trials}$ 
    net = neural model with  $H$  hidden neurons
    Design net with training data.
    Compute validation data from validation data.
  end
end
bestNeuralModel = net with lowest validation error
    
```

### III. MEANDER LINES

Serpentine lines have been widely used in printed circuit boards (PCB) to provide the required timing delays for timing margin management. Nevertheless, as the processor speed and clock frequency increases, conventional circuits such as delay lines have to be considered as distributed elements. Full-wave electromagnetics simulation tools provide an accurate analysis of these microwave structures, but they are also computationally expensive and slow. In [15], a detailed study about meander lines is conducted. From the study, it can be observed that a straight line has more delay compared to serpentine segments designed with the same conductor length. This is due to the cross-talk coupling between the segments. In the case of the straight line structures, the electric fields are mostly directed to the ground. On the other hand, in serpentine segments, due to cross-talk effects, the electric fields get coupled to the adjacent segment causing the

reduction in delay.

Fig. 3 shows an example of a top view of a meander microstrip line structure. As discussed above, implementation of the meander line without careful design considerations may lead to timing violations due to unaccounted delay variations. Some of the critical parameters to be considered when designing the meander line include the line length, line width, spacing, number of bends, and types of bends. Furthermore, the substrate thickness, dielectric of the substrate, and rise time of signals will also affect the propagation delay. In this work, the segment length and segment spacing are varied, while the other parameters are held constant. This is typical in a design scenario where the board materials and signaling speed are decided beforehand based on the other processes. The readers are referred to references [24-26] for more background on meander line.

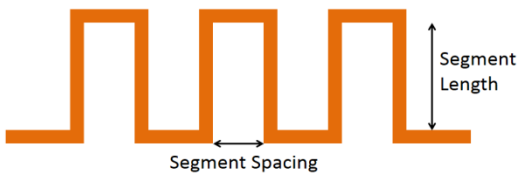


Fig. 3. Meander line structure

#### IV. DATA COLLECTION AND NEURAL MODEL TRAINING

In order to train the ANN, multiple meander lines are simulated using the Momentum EM simulator in Keysight's Advanced Design System (ADS). In each case, the S-parameters which represent the behavior of the lines are extracted. In this work, two cases are investigated; (a) Case 1: Effect of segment length on propagation delay time and (b) Case 2: Effect of segment spacing on propagation delay time. In all simulations, the impedances of both ports are specified at 50 Ω. The width of the line is also calculated such that the impedance of the line is 50 Ω to match the impedances of the ports. The number of turns for the meander lines is set to 12 bends. The type of bend chosen is the right-angle bend as it is more easily fabricated in the industry.

The microstrip substrate parameters are set where the substrate thickness is 0.813 mm, the relative dielectric constant,  $\epsilon_r$  is 3.38, relative permeability is 1, line width is 1.84 mm and the thickness of the conductor is 0.03556 mm. The layout of the meander line from the Momentum EM software is shown in Fig. 4. A frequency domain simulation is performed from 0 GHz to 10 GHz with a step of 0.01 GHz. Fig. 5 shows the plot of the S-parameters from the Momentum EM simulation compared with a simulation using a circuit level modeling. While the plots both show the same characteristics, there is a slight difference as the circuit model does not account for the couplings as accurately as the full-wave Momentum simulation.

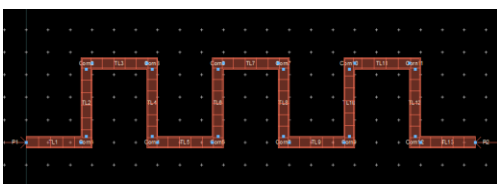


Fig. 4. Layout of the meander line structure

In order to determine the delay of the lines, the S-parameters are then used in a step voltage transient simulation as shown in Fig. 6. A step voltage source with a minimum voltage of 0 V, maximum voltage of 1 V and rise time of 1 ns is used. The voltage waveform at port 1 and port 2 are captured to determine the propagation delay of the meander line. This is visualized in Fig. 7. During simulation, several physical parameters of the microstrip lines are varied and data is collected as input-target pair sets that are used during training, validation and testing of the ANNs.

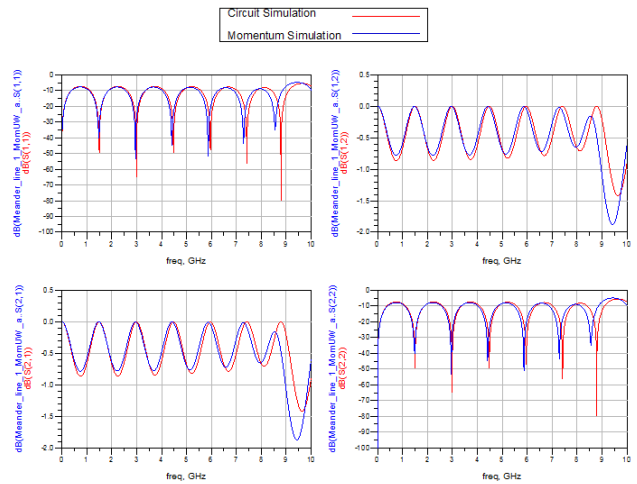


Fig. 5. S-parameters from circuit model simulation and Momentum EM simulation

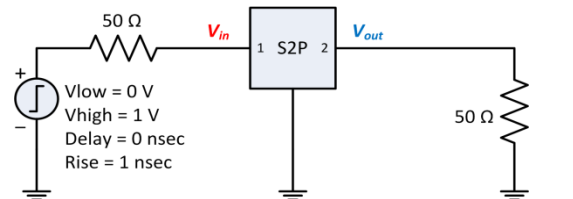


Fig. 6. Transient simulation setup of the meander line

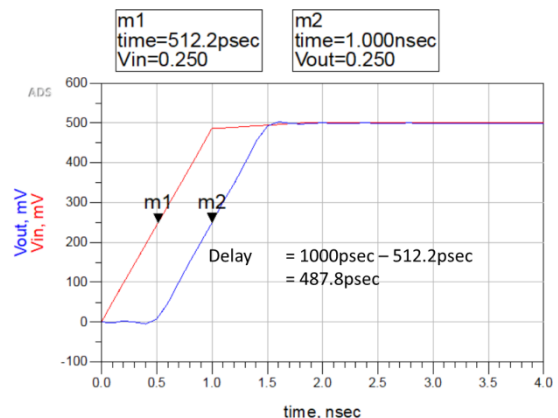


Fig. 7. Calculation of delay from the voltage waveform at port 1 ( $V_{in}$ ) and port 2 ( $V_{out}$ )

## V. NEURAL NETWORK TRAINING AND PERFORMANCE COMPUTATION

MATLAB is used to build and train the neural networks. Early stopping is used to improve the generalization capability of the neural networks. The early stopping method requires users to separate their data into three groups, namely training, validation and testing data. In this work, the data collected are randomly split into training, validation and testing sets according to a specified split ratio. Training data is used to update the weights and biases of the neural model during the training process. Validation data is used to terminate the training process if the validation error increases continuously for several iterations. Testing data is used for an unbiased performance evaluation of the neural models. The collected data are arranged as follows:

$$\text{input} = [x_1 \quad x_2 \quad x_3 \quad K], \quad (3)$$

$$\text{target} = [t_1 \quad t_2 \quad t_3 \quad K]. \quad (4)$$

The universal approximation theorem [27-28] states that a feed-forward network with a single hidden layer containing a finite number of neurons can approximate any continuous functions to any degree of accuracy. Thus in this work, a three layered MLP which has only one hidden layer with a tansig activation function in the hidden layer and a linear output activation function is used. In addition, the Levenberg-Marquardt backpropagation algorithm [29] is used to train the neural network models. The Levenberg-Marquardt algorithm has a benefit of faster convergence. This advantage is especially noticeable if very accurate training is required. For example, results in [30] show that the Levenberg-Marquardt algorithm is able to obtain lower mean square errors than any of the other algorithms tested.

The performances of the neural networks are rated by using the coefficient of determination,  $R^2$ :

$$R^2 = 1 - \frac{\text{mse}(t_N - y_N)}{\frac{1}{O} \sum_{i=1}^O \text{var}(t_{N,i})} \quad (5)$$

where  $\text{mse}(t_N - y_N)$  is mean squared error between normalized targets,  $t_N$  and normalized outputs  $y_N$  of the neural model, and  $\text{var}(t_{N,i})$  is the variance of the  $i$ th output neuron,  $t_{N,i}$  defined as follows:

$$\text{mse}(t_N - y_N) = \frac{\sum_{i=1}^O \sum_{j=1}^{\text{length}(t_{N,i})} (t_{N,i,j} - y_{N,i,j})^2}{\sum_{i=1}^O \text{length}(t_{N,i})}, \quad (6)$$

$$\text{var}(t_{N,i}) = \frac{1}{\text{length}(t_{N,i})} \sum_{j=1}^{\text{length}(t_{N,i})} (t_{N,i,j} - \overline{t_{N,i}})^2, \quad (7)$$

where  $\text{length}(\cdot)$  is the length of the vector. The mean of targets,  $\overline{t_{N,i}}$  is defined as follows:

$$\overline{t_{N,i}} = \frac{1}{\text{length}(t_{N,i})} \sum_{j=1}^{\text{length}(t_{N,i})} t_{N,i,j}. \quad (8)$$

The training data is normalized to the range of [-1, 1] to improve training performance.

## VI. SIMULATION RESULTS

This section discusses the results obtained from the ANN simulations. The results are analyzed in terms of the accuracy and the time it takes to train the neural network for a different number of hidden neurons and a different number of inputs/data points. The results involve two cases which are delay times versus segment length and delay times versus segment spacing. All simulations are done on a 2.6 GHz 3rd Gen. Intel Core i5 processor with 4 GB of RAM.

### A. Case 1: Delay Times Versus Segment Length

For Case 1, 130 samples are simulated from ADS where 100 of them are used to train and validate the neural models and the rest is used for unbiased performance evaluation. The segment length is varied from 2 mm to 11.9 mm.  $H_{max}$  is calculated to be 19 during training. 10 models are trained for each number of hidden neurons which totals up to 190 network models being created.

The best design has 12 hidden neurons, a validation performance of 0.9990 and a testing performance of 0.9995. The average training time of each neural model is only 0.1358 second which is very short. Fig. 8 shows that the ANN predictions closely match the ADS simulation results. It can be observed that the increment of meander line segment length results in shorter propagation delay, even though the total length of the line remained the same. For longer segment length, the cross-coupling between adjacent segments is amplified, which provide another alternative path for the signal to bypass the full length of the line. Other than that, we also study the effect of training sample size on the performance of the neural model. We trained the neural model with training data of different sizes and then test it with 30 testing samples. As can be seen from Fig. 9, the training process of the neural model converges at a training data size of 40. After that, increasing the training data size has little impact on the test performance of the neural model.

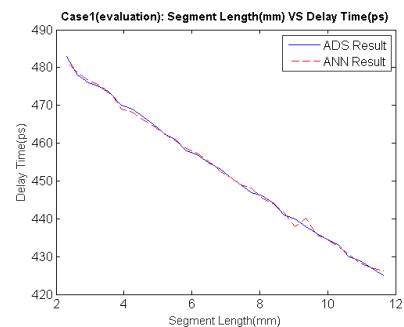
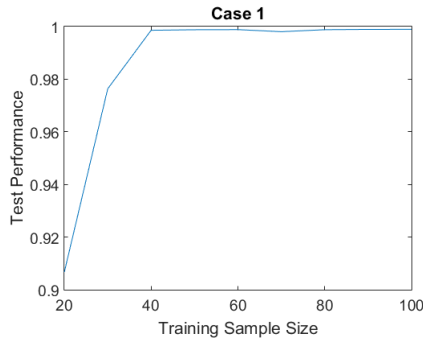
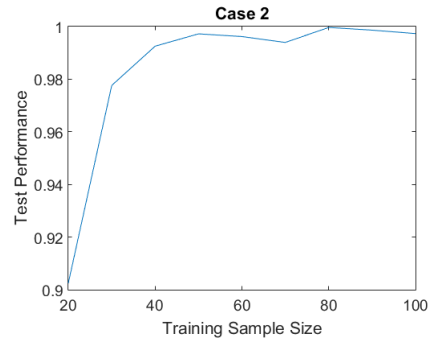


Fig. 8. Calculation of delay from the voltage waveform at port 1 ( $V_{in}$ ) and port 2 ( $V_{out}$ )



**Fig. 9. Testing performance versus training sample size for Case 1**



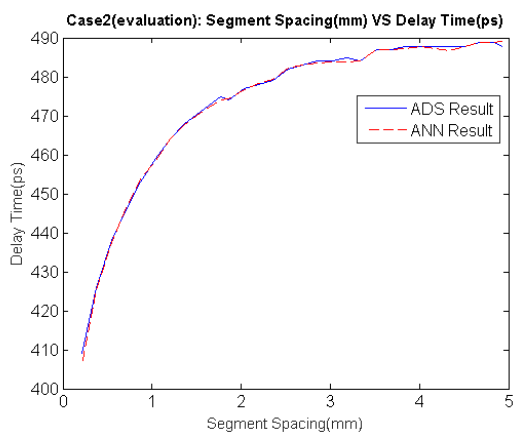
**Fig. 11. Testing performance versus training sample size for Case 2**

### B. Case 2: Delay Times Versus Segment Spacing

For Case 2, 190 new networks are created to model the meander lines. 100 samples are used for training and validation and an additional 30 samples are generated to evaluate the performances of the neural models. In this case, the segment spacing is varied from 0.05 mm to 5 mm.

The best design has 11 hidden neurons, a validation performance of 0.9995 and a testing performance of 0.9990. The average training time of each neural model is 0.1991 second. Fig. 10 shows a good agreement between ANN predictions and ADS simulation results. In this case, the delay time decreases with a reduction in the segment spacing. This is due to the coupling effect between the segments. As the spacing decreases, the coupling between adjacent lines increases. It is observed that when the segment spacing is between 5 mm to 2 mm the propagation delay reduces slowly. However, the propagation delay reduces significantly when segment spacing is below 2 mm. As in case 1, we trained the neural model with training data of different sizes and then test it with 30 testing samples. As can be seen from Fig. 11, the training process of the neural model converges at a training data size of 50.

Finally, as a comparison of simulation time, the total time from generating the S-parameters using Momentum and using the generated S-parameters to determine the delay of the line in a transient simulation is about 95 s, while the time for the trained ANN to determine the delay for a given physical parameter was only 0.04 s. This gives over 2000× improvement in speed when using the proposed ANN modeling.



**Fig. 10. Calculation of delay from the voltage waveform at port 1 ( $V_{in}$ ) and port 2 ( $V_{out}$ )**

## VI. CONCLUSION

In this work, artificial neural networks have been used to model the delay in meander lines. Due to the cross coupling between adjacent traces of the lines, the delay of the meander line structure is not easily predicted using normal delay equations. On the other hand, accurate electromagnetics solvers are slow and computationally expensive. Results obtained from the proposed method indicate that the neural models are able to provide accurate (above 99.5% accuracy) and fast (over 2000× speed-up) solutions compared to using conventional EM simulation tools, thus making it an attractive alternative to circuit designers. Future work will focus on modeling using higher order ANNs to include other design parameters of the lines.

## VII. ACKNOWLEDGMENT

This work is supported by the Universiti Sains Malaysia under the Research University Grant (RUI) grant no. 1001/PELECT/8014011.

## REFERENCES

1. S. Haykins, *Neural Networks A Comprehensive Foundation*, Upper Saddle River, NJ, USA: Prentice Hall, 1994.
2. Y. Cao, S. Reitzinger and Q. J. Zhang, "Simple and efficient high-dimensional parametric modeling for microwave cavity filters using modular neural network", *IEEE Microwave and Wireless Components Letters*, vol. 21, no. 5, pp. 258-260, 2011.
3. F. Feng, V. Gongal-Reddy, C. Zhang, J. Ma and Q. Zhang, "Parametric modeling of microwave components using adjacent neural networks and pole-residue transfer functions with EM sensitivity analysis", *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 6, pp. 1955-1975, 2017.
4. H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, Q. J. Zhang, "Smart modeling of microwave devices", *IEEE Microwave Mag.*, vol. 11, no. 3, pp. 105-118, May 2010.
5. W. Na, W. Liu, L. Zhu, F. Feng, J. Ma and Q. Zhang, "Advanced extrapolation technique for neural-based microwave modeling and design", *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 10, pp. 4397-4418, 2018.
6. T. Liu, S. Boumaiza, F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifier using real-balued time delay neural networks", *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 3, pp. 1025-1033, Mar. 2004.
7. G. Xue-lian, C. Zhen-nan, F. Nan, Z. Xiao-yu, and H. Jian-hong, "An artificial neural network model for s-parameter of microstrip line," in *2013 Asia-Pacific Symposium on Electromagnetic Compatibility (APEMC)*, May 2013, pp. 1-4.
8. R. Varma and J. Ghosh, "Design of electromagnetically-coupled microstrip antennas using neural networks", in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 185-189, 2016.



9. W. T. Beyene, "Application of artificial neural networks to statistical analysis and nonlinear modeling of high-speed interconnect systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 1, pp. 166-177, Jan. 2007.
10. N. Ambasana, D. Gope, B. Mutnury and G. Anand, "Eye-height/width prediction from S-parameters using bounded size training set for ANN," in *2014 IEEE Electrical Design of Advanced Packaging & Systems Symposium (EDAPS)*, Bengaluru, India, Dec. 2014, pp. 17-20.
11. N. Ambasana, G. Anand, B. Mutnury and D. Gope, "Eye height/width prediction from S-parameters using learning-based models," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 6, no. 6, pp. 873-885, Jun. 2016.
12. N. Ambasana, G. Anand, D. Gope and B. Mutnury, "S-parameter and frequency identification method for ANN-based eye-height/width prediction," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 5, pp. 698-709, 2017.
13. C. H. Goay, P. Goh, N. S. Ahmad and M. F. Ain, "Eye-height/width prediction using artificial neural networks from S-parameters with vector fitting," *Journal of Engineering Science and Technology*, vol. 13, no. 3, pp. 625-639, 2018.
14. T. Lu, J. Sun, K. Wu and Z. Yang, "High-speed channel modeling with machine learning methods for signal integrity analysis," *IEEE Transactions on Electromagnetic Compatibility*, vol. PP, no. 99, pp. 1-8, 2018.
15. A. Datta, N. Datta, T. Bandyopadhyay and S. Sinha, "Signal integrity analysis of serpentine traces in IC packages," in *2016 IEEE Dallas Circuits and Systems Conference (DCAS)*, 2016.
16. M. Zhang, H. Tan, "Accurate delay extraction of serpentine lines for next-generation high-density DRAMs," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 5, no. 12, Dec. 2015.
17. J. Shin and T. Michalka, "Comprehensive, scalable design guidance for serpentine time delay variation in digital system," in *2012 IEEE 62nd Electronic Components and Technology Conference*, 2012.
18. W. Soh, K. See, W. Chang, Richard, M. Oswal and L. Wang, "Comprehensive analysis of serpentine line design," in *2009 Asia Pacific Microwave Conference*, Singapore, pp. 1285-1288, 2009.
19. S. H. Hall and H. L. Heck, *Advanced Signal Integrity for High-Speed Digital Designs*. Hoboken, NJ, USA: Wiley, 2011, pp. 152-153.
20. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 1999, pp. 122-219.
21. D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *IJCNN International Joint Conference on Neural Networks*, vol. 3, pp. 21-26, Jul. 1990.
22. T. Vujčić, T. Matijević, J. Ljucović, A. Balota and Z. Ševarac, "Comparative analysis of methods for determining number of hidden neurons in artificial neural network," in *Cent. Eur. Conf. Inf. Intell. Syst.*, Varaždin, Croatia, pp. 219-223, 2016.
23. H. Kabir, L. Zhang, M. Yu, J. Wood, P. H. Aen, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microwave Magazine*, pp. 105-118, May 2010.
24. A. Kabiri, Q. He, M. H. Kermani, and O. M. Ramahi, "Design of a controllable delay line," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 5, pp. 1080-1087, 2010.
25. B. J. Rubin and B. Singh, "Study of meander line delay in circuit boards," *IEEE Transactions on Microwave Theory and Techniques*, vol. 48, no. 9, pp. 1452-1460, 2000.
26. W. D. Guo, G. H. Shiue, C. M. Lin and R. B. Wu, "Comparisons between serpentine and flat spiral delay lines on transient reflection/transmission waveforms and eye diagrams," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 4, pp. 1379-1387, 2006.
27. G. Cybenko, "Approximations by superpositions of sigmoidal functions," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303-314, 1989.
28. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.
- M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989-993, 1994.
29. H. Demuth, M. Beale, *Neural Network Toolbox*, MA, Natick: The MathWorks, Inc.