

Adaptive Sampling Algorithm for ANN-based Performance Modeling of Nano-scale CMOS Inverter

Dipankar Dhabak and Soumya Pandit, *Member, IEEE*

Abstract—This paper presents an adaptive technique for generation of data required for construction of artificial neural network-based performance model of nano-scale CMOS inverter circuit. The training data are generated from the samples through SPICE simulation. The proposed algorithm has been compared to standard progressive sampling algorithms like arithmetic sampling and geometric sampling. The advantages of the present approach over the others have been demonstrated. The ANN predicted results have been compared with actual SPICE results. A very good accuracy has been obtained.

Keywords—CMOS Inverter, Nano-scale, Adaptive Sampling, Artificial Neural Network

I. INTRODUCTION

Circuit simulation tools are indispensable components for estimating the performances of nano-scale CMOS integrated circuits [1]. In an optimization-based design procedure, the circuit simulation tools are embedded within a stochastic global search optimization procedure, used for the task of circuit sizing. However, the task of circuit simulation is computationally expensive, which increases almost in an exponential way in the nano-scale regime [2]. Therefore, lots of researches have been accomplished in creating comprehensive performance models of integrated circuits [3]. These models predict the performances of an integrated circuit as functions of the design parameters. As a result, these models may act as surrogates for full circuit simulation. The models are often multidimensional and nonlinear in order to accurately capture the intricate details of the performance parameters [3], [4]. In this paper, artificial neural network (ANN) has been used for constructing accurate performance models of nano-scale CMOS inverter circuit.

Several data mining algorithms, including ANN have an important property that as the training set size increases, the accuracy increases until at some point it saturates, i.e., as the training set size increases beyond a certain value, the predictive accuracy does not increase significantly. A too-small training set will thus results in sub-optimal generalization performance. On the other hand, a too-large training set will results in lot of training time consumption without any significant advantage. In addition, the procedure of training sample generation is often very costly, especially for integrated circuit design applications. The task of determining an optimal training set size for acceptable predictive accuracy is therefore, an

important challenge for developing an ANN-based predictive performance model.

In this paper, we present an adaptive sampling algorithm for generation of an optimum size of training sample set. This allows a unified model building process that incorporates both data generation and neural network training. The present algorithm has been used to construct the performance model of a nano-scale CMOS inverter. The training data are generated from the samples through SPICE simulation. The accuracy of the constructed model is found to be quite high. The present algorithm has been compared with other standard progressive sampling algorithms.

The rest of the paper is organized as follows. Section II discusses related works in performance modeling and adaptive sampling. An overview of performance modeling using ANN is described in Section III. The adaptive sampling algorithm is described in details in Section IV. Section V presents the numerical results and finally conclusion is drawn in Section VI.

II. RELATED WORK

Artificial neural network (ANN) has been widely used for solving electronic engineering problems. In [5], ANN has been used for selecting the channel length and width of a MOS transistor for specific drain current. Signal and noise behavior of microwave transistors are modeled by multi layer perceptron (MLP) neural network in [6]. In [7], [8], ANN has been used for the task of technology independent circuit sizing for analog and digital integrated circuits. ANN has been used in [9] for simulation of nano-scale CMOS circuits. In [10], ANN has been used for modeling an on-chip spiral inductor.

The difference between static and dynamic sampling for data mining has been discussed in [11]. The concept of progressive sampling using arithmetic sampling and geometric sampling has been discussed in [12]. In [13], a learning curve sampling technique has been discussed for model-based clustering problem. A dynamic adaptive sampling algorithm using Chernoff inequality for ANN classification problems has been presented in [14]. In [15], neural network training driven adaptive sampling algorithm for microwave modeling has been described. An adaptive sampling technique for modeling of analog circuit performance parameters using pseudo-cubic splines is discussed in [16].

D.Dhabak is associated with the A.K.Choudhury School of Information Technology, University of Calcutta, Kolkata, India

S.Pandit is associated with the Institute of Radio Physics and Electronics, University of Calcutta, Kolkata, India

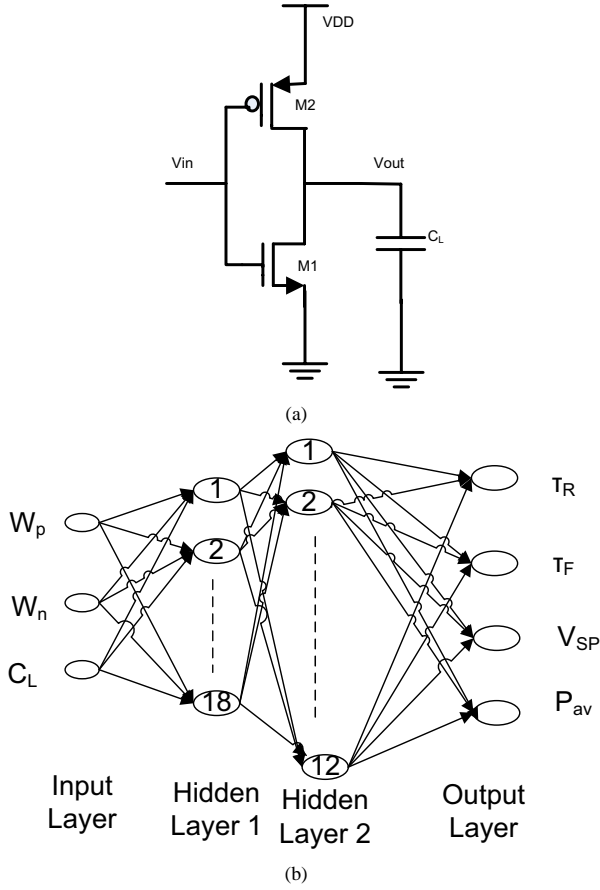


Fig. 1. (a). CMOS Inverter (b) ANN Architecture

III. PERFORMANCE MODELING USING ANN

This section briefly discusses the procedure for generation of performance models for nano-scale CMOS inverter using ANN [17], based on the general methodology described in [18].

A. Problem Formulation

The basic circuit diagram of a CMOS inverter is shown in Fig. 1(a). Let p and q be the number of input and output neurons of an ANN structure. Let \bar{X} be the p dimensional input vector containing the circuit design parameters, i.e., the channel width W_n of the NMOS transistor, the channel width W_p of the PMOS transistor and the output load capacitor C_L . Let $\bar{\rho}$ be the q dimensional output vector containing the performance parameters of the design, i.e., output rise time (τ_R) and fall time (τ_F), inverter switching point (V_{SP}) and average power consumption (P_{av}). Thus the inputs and outputs of the performance model are as follows

$$\bar{X} = [W_n, W_p, C_L] \quad (1)$$

$$\bar{\rho} = [\tau_R, \tau_F, V_{SP}, P_{av}] \quad (2)$$

The performance model is thus written as

$$\bar{\rho} = f(\bar{X}) \quad (3)$$

TABLE I
RANGE OF CIRCUIT DESIGN PARAMETERS

Parameters	Min	Max
$W_n(nm)$	90	1000
$W_p(nm)$	90	1000
$C_L(pF)$	1	5

This relationship between the circuit design parameters and the performance parameter is generally strongly nonlinear and multi-dimensional. Traditionally this is evaluated through SPICE simulation. The corresponding neural network model is written as

$$\bar{\rho} = f_{ANN}(\bar{X}) \quad (4)$$

where f_{ANN} is a neural network, ρ is a q dimensional output vector of neural model responses, \bar{X} is the ANN input vector, w contains all the weight parameters required to construct the ANN structure. This work therefore, attempts to construct f_{ANN} such that it is a faithful approximation of the original function f .

B. ANN Model Development

1) *Data Generation*: In order to generate training and test data, CMOS inverters are constructed corresponding to the circuit design parameters listed in Table I. The channel length of both the transistors is fixed at minimum of the process technology, i.e., 45nm. The other process technology parameters are taken from Berkeley Predictive Technology model file [19]. Based on Halton sequence generator [20], uniformly distributed samples are generated within the specified range. The training and test data corresponding to those sample points are generated through T-SPICE simulation using BSIM4 model. Transient analysis and DC transfer sweep analysis are performed in order to extract the performance parameters.

2) *Data Scaling*: It is observed from Table I that the input parameters vary over a wide range. Similarly the output performance parameters vary over a wide range. Therefore, a systematic pre-processing of training data, referred to as data scaling is required for efficient construction of the ANN model. In this work, we have used linear scaling of the data between 0 and 1, described by the following formula

$$\tilde{x} = \tilde{x}_{min} + \frac{x - x_{min}}{x_{max} - x_{min}} (\tilde{x}_{max} - \tilde{x}_{min}) \quad (5)$$

and the corresponding de-scaling formula is given by

$$x = x_{min} + \frac{\tilde{x} - \tilde{x}_{min}}{\tilde{x}_{max} - \tilde{x}_{min}} (x_{max} - x_{min}) \quad (6)$$

where x , x_{min} , x_{max} represent the original data and represent \tilde{x} , \tilde{x}_{min} , \tilde{x}_{max} the scaled data.

3) *Data Organization*: The generated data is divided into two sets, namely training data set and test data set. The training data is used to guide the training procedure, i.e., updating the NN weight parameters. A portion of the training data set is used for validating the training procedure. The test data is used to independently examine the final quality of the trained neural model in terms of accuracy and generalization capability.

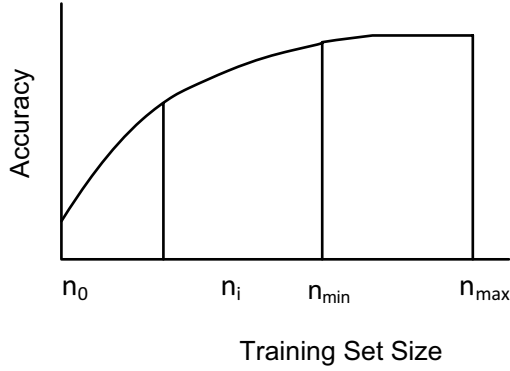


Fig. 2. Hypothetical Learning Curve for ANN Model

4) *Neural Network Training*: A standard 4-layer feedforward MLP architecture has been considered in order to construct the ANN model of the inverter. This is illustrated in Fig. 1(b). During the training procedure, the weight parameters and the bias values are adjusted in order to minimize the training error. For this purpose, we have used Levenberg-Marquardt (LM) back propagation method as the training algorithm. The training goal is set to 10^{-7} . The training algorithm of Matlab toolbox has been used.

5) *ANN Model Accuracy*: In order to verify the accuracy of the constructed ANN model, statistical measures such as average relative error and correlation coefficient between the neural outputs and actual SPICE generated values are calculated for each output parameter. These are defined as follows

$$E = \frac{1}{n\rho} \sum_1^n (\rho - \rho') \quad (7)$$

$$R = \frac{n \sum \rho \rho' - \sum \rho \sum \rho'}{\sqrt{[n \sum \rho^2 - (\sum \rho)^2] - [n \sum \rho'^2 - (\sum \rho')^2]}} \quad (8)$$

Here, n , ρ and ρ' are the number of samples in the data set, ANN model output and corresponding SPICE simulated value respectively. The correlation coefficient is a measure of how closely the ANN outputs fit with the target values. It is a number between 0 and 1. If there is no linear relationship between the estimated values and the actual targets, then the correlation coefficient is 0. If the number is equal to 1.0, then there is a perfect fit between the targets and the outputs. Thus, higher the correlation coefficient, the better it is.

IV. ADAPTIVE SAMPLING ALGORITHM

This section first discusses the motivation of the algorithm, followed by an overview of the algorithm and its details.

A. Motivation

The requirement of adaptive sampling algorithm for construction of an ANN-based performance model is based upon three observations. First, the predictive performance/accuracy of the ANN increases initially with the increase of training data set size, however, beyond a certain data set size, the

```

Input: nmax, δ, Y
Output: nmin corresponding data set Dmin

Step 1:
(a) Initialize n0=0.1nmax
(b) Generate initial data set D0 using Halton sequence and SPICE simulator
(c) Initial performance u∧(-1)=0

Step 2:
For iteration i=0 to imax do
  (a) Set m=Di
  (b) Apply ANN to m and determine u∧(ni)
  (c) IF |u∧(ni)-u∧(ni-1)| ≤ ε && u∧(ni) ≤ Y
      TERMINATE Return ni and Di
    ELSE
      Calculate ni+1 using (9)
      IF ni+1 < nmax
        (i) Generate Di+1 data set using Halton sequence and SPICE simulator
        (ii) Goto Step 2(a)
      ELSE
        Data set exhausted BREAK

```

Fig. 3. Adaptive Sampling Algorithm

accuracy does not increase significantly. This is referred to as the learning characteristics of an ANN algorithm. The curve describing the performance as a function of the sample size of the training data is often called the learning curve. A typical plot of the learning curve of an ANN predictive model is shown in Fig. 2. It is observed from this curve, that the models built with training set size lower than n_{min} , will have lower accuracy compared to that of the models built with training set size n_{min} . On the other hand, models built with training set size greater than n_{min} , will not have any significant higher accuracy compared to that of the models built with training set size n_{min} . Second, the computational cost of training an ANN model increases as function of the size of the training data set. Third, the cost of training data generation for circuit performance modeling is quite expensive.

B. Overview of the Algorithm

The algorithm takes as inputs: (i) The maximum sample size n_{max} , corresponding to which the sample set can be generated, (ii) a very small value ϵ , which is used to formulate the stopping criteria and (iii) desired accuracy Y of the model. It gives as output the minimum sample size n_{min} and the corresponding data sample D_{min} . The algorithm starts with an initial sample size $n_0 = 0.1 \times n_{max}$. Corresponding to this, the initial data set D_0 is generated through Halton sequence generator and SPICE simulation. Subsequently the next sample size $n_{(i+1)}$ and the corresponding data set $D_{(i+1)}$ is generated through an iterative procedure. The algorithm terminates when a stopping criterion is satisfied. In addition, if the optimum value cannot be located within n_{max} , the algorithm breaks. The pseudo code of the algorithm is described in Fig. 3

C. Details of the Algorithm

The problem may be considered to be a decision-theoretic problem, where we have to judiciously decide how to compute

TABLE II
COMPARISON OF THE SAMPLE SIZE AND COMPUTATION TIME FOR
DIFFERENT METHODS TO REACH CONVERGENCE

Method	Sample size	ARE	cpu time
Full Sampling	1030	0.0648	-
Arithmetic Sampling	1000	0.0643	117.2
Geometric Sampling	800	0.0717	46.8
Our Algorithm	923	0.0647	65.6

$n_{(i+1)}$ i.e., the sampling schedule, initial sample value and the stopping criteria. These are discussed as follows

1) *Initial Sample Size*: From the preliminary knowledge about learning curve characteristics, an useful conjecture is to take a small initial sample size (determination of the starting sample size is an open problem). In the present work, it is heuristically assumed to be $n_0 = 0.1 \times n_{max}$.

2) *Sampling Schedule*: A 'myopic' strategy has been adopted, where we assume that the current performance measure of the ANN is the optimal one. The next sample size is believed to be distributed somewhat around the current sample size. We assume this distribution to be Gaussian distribution. The mean of the Gaussian distribution is kept at the current point and the variance is assigned so as to have about 99.73% (equivalent to 3σ) of the points in the given domain ($n_0 \leq n_i \leq n_{max}$). The variance σ is found by solving the equation

$$3\sigma = \frac{(n_{max} - n_0)}{2} \quad (9)$$

With this variance, the next sample size is calculated by the formula

$$n_{i+1} = \mu + \sigma N \quad (10)$$

where mean $\mu = n_i$ and N is a random number drawn from a Gaussian distribution with zero mean and unity standard deviation σ .

3) *Stopping Criteria*: An important component of the sampling algorithm is the stopping criteria. Let the current stage be i and the previous stage be $(i-1)$ and the corresponding performance measures be $\hat{u}(n_i)$ and $\hat{u}(n_{i-1})$ respectively. The following inequality is considered as one of the stopping criteria.

$$|\hat{u}(n_i) - \hat{u}(n_{i-1})| \leq \epsilon \& \hat{u}(n_i) \leq \Upsilon \quad (11)$$

where ϵ is a very small value, depending upon the chosen application. Simultaneously the desired accuracy of the model has to be satisfied. It may be noted that the performance measure $\hat{u}(i)$ is calculated based on the average relative error E , as discussed in (8). In addition, if the algorithm does not find the value of the optimal sample set within the given bound of the sample size, the algorithm will terminate.

V. NUMERICAL RESULTS

In this section, we present numerical results to demonstrate the utility of the present algorithm. We will first provide a test example and then discuss how this algorithm is applied to the problem of performance model construction of CMOS inverter.

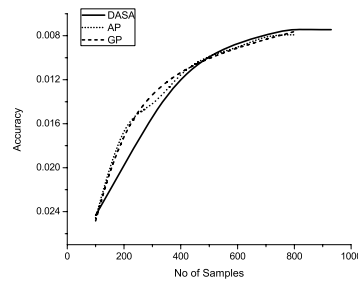


Fig. 4. Comparison of learning curves for concrete problem between different sampling techniques

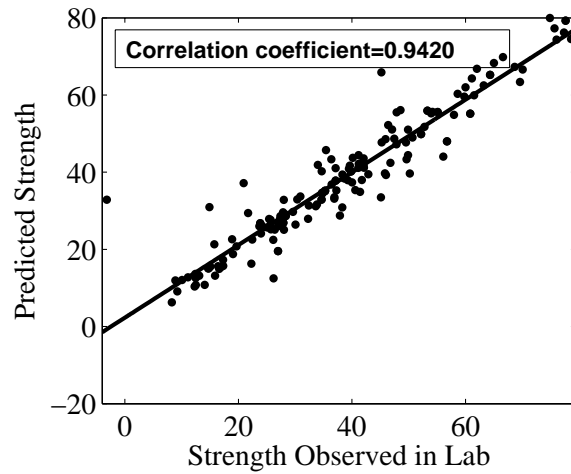


Fig. 5. Scatter plot between ANN predicted strength and actual results

A. Test Case

The algorithm is applied to a test case of predicting the compressive strength of high performance concrete using ANN [21]. The compressive strength of concrete is a function of eight parameters, which serve as the inputs of the ANN structure. These are Cement, water, coarse aggregate content, fine aggregate content, age of testing, fly ash, blast furnace slag, superplasticizer and water-to-binder ratio. A total number of 1030 samples (input-output data) are provided [22]. The chosen network architecture is feedforward multilayer perceptron, with one hidden layer and eighth neurons. This is similar to that used in [21].

The convergence of our algorithm has been compared to other approaches such as full sample, arithmetic sampling and geometric sampling. In full samples, all the available data are used for ANN prediction. In arithmetic sampling, a fixed number of samples (100 in this case) are added until convergence is reached. In geometric sampling, the training set size is increased geometrically with common ratio 2. The initial sample size in each case is 100. For our algorithm, we have used $\epsilon = 1 \times 10^{-4}$.

The optimum sample size n_{min} is found out to be equal to 923. A comparison of the total number of samples (av-

TABLE III

COMPARISON OF THE TOTAL NUMBER OF ITERATIONS AND CPU TIME REQUIRED FOR THE DIFFERENT METHODS TO REACH CONVERGENCE FOR THE INVERTER PROBLEM.

Method	Iteration Count	cpu time
Arithmetic Sampling	9	189.6
Our Algorithm	6	117.5

eraged over 10 runs) and the computation time required for the different methods is provided in Table II. The learning curve corresponding to our algorithm, arithmetic sampling and geometric sampling is illustrated in Fig. 4. This timing information is based on PC with Core-2-duo processor and 2GB RAM. We observe that compared to full sampling, using our algorithm same accuracy can be obtained with much less number of samples. This observation is also true while comparing with arithmetic sampling. For geometric sampling, we observe that the convergence could not be reached. With the current sampling schedule for geometric sampling, the algorithm overshooted the optimum point. As far the computation time is considered, we observe that our algorithm takes less time compared to the arithmetic sampling.

The correlation plot between the predicted ANN data and actual data considering 923 samples is shown in Fig. 5. The correlation coefficient is also shown. This is found to be same as that obtained when all the available samples have been used for ANN prediction and also as reported in [21].

B. CMOS Inverter

The data generation procedure has been carried out using the standard sampling schemes as well as our algorithm. The modeling results of the three sampling techniques are summarized in Fig. 6(a) - 6(d) for each of the chosen performance parameters. The optimum sample size n_{min} is found out to be equal to 288. We observe from the learning curves, that for all the cases, the geometric sampling technique cannot reach the convergence. The arithmetic sampling technique reaches the optimum point with more iterations compared to that required for our algorithm. In each iteration new samples are generated for ANN training. The quantitative data regarding this is provided in Table III. This timing information is based on PC with Core-2-duo processor and 2GB RAM.

It may be noted that the efficiency of the sampling algorithm in locating the optimum size n_{min} neither depends on the choice of the initial sample size nor on the maximum sample size n_{max} . This is illustrated for the rise time τ_R output of the inverter in Fig 7 below. In order to verify the quality of the resultant ANN, we measure the various quality metrics discussed above. The percentage E measured on test data for all the outputs are summarized in Table IV. We observe that a very good accuracy has been obtained in each case. Figure 8(a) - 8(d) respectively show τ_R , τ_F , V_{SP} and P_{av} for 100 designs, obtained through ANN and SPICE simulations. We observe that all the ANN outputs show good matching with SPICE results. The scatter plots between the ANN predicted results and SPICE simulations are shown in Fig. 9(a) - 9(d). We observe nearly perfect diagram with unity correlation

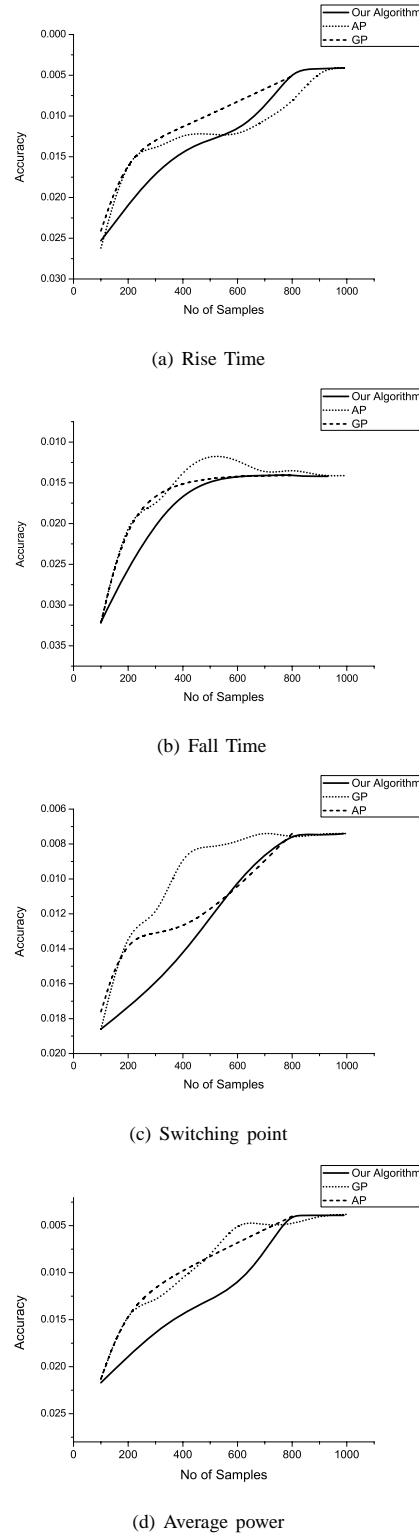


Fig. 6. Learning Curves of the various outputs obtained through different sampling techniques.

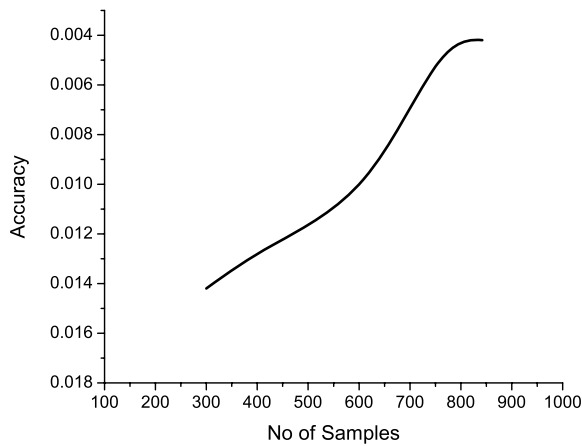


Fig. 7. Learning Curves of the rise time τ_R output illustrating the independence of the algorithm on the bound of the sample size.

coefficient. These demonstrate the accuracy of the constructed ANN model.

TABLE IV
ANN MODEL ACCURACY

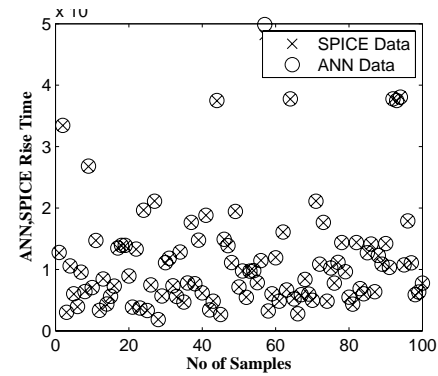
Error	Output	ARE (%) Test data
E	τ_R	0.42
	τ_F	1.41
	V_{SP}	0.74
	P_{SP}	0.39
R	τ_R	0.9999
	τ_F	0.9998
	V_{SP}	0.9999
	P_{SP}	0.9999

VI. CONCLUSION

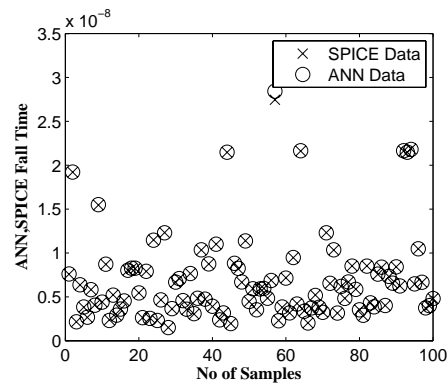
Data generation is an important step toward developing accurate ANN model. Using a large set of training data does not always results in significant improvement in prediction accuracy of the model. On the other hand, a small sampling set may yield an inaccurate model. In addition, the data generation procedure is often a costly procedure, especially for integrated circuit designs. The present adaptive sampling algorithm provides a unified way of generating samples incorporating both data generation and neural network training procedure. The algorithm drives the data generator to increment the size of the sampling set, until there is no significant improvement in the model accuracy. The algorithm has been used to construct the performance model of a nano-scale CMOS inverter circuit. The accuracy of the so constructed model is found to be quite high. In addition, the proposed algorithm has been compared with other standard progressive sampling algorithm and the advantages of our approach have been demonstrated.

ACKNOWLEDGMENT

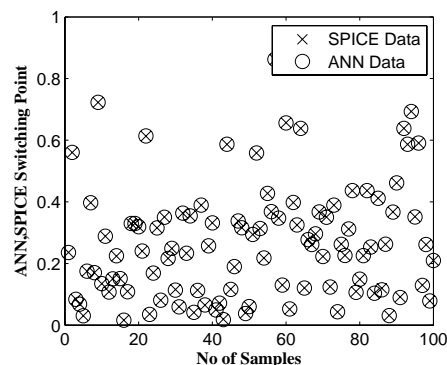
The second author thanks the Department of Science and Technology, Govt. of India for financially supporting the present work through DST-FAST Young Scientist scheme.



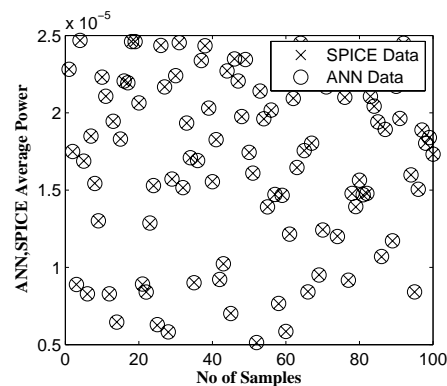
(a) Rise Time



(b) Fall Time

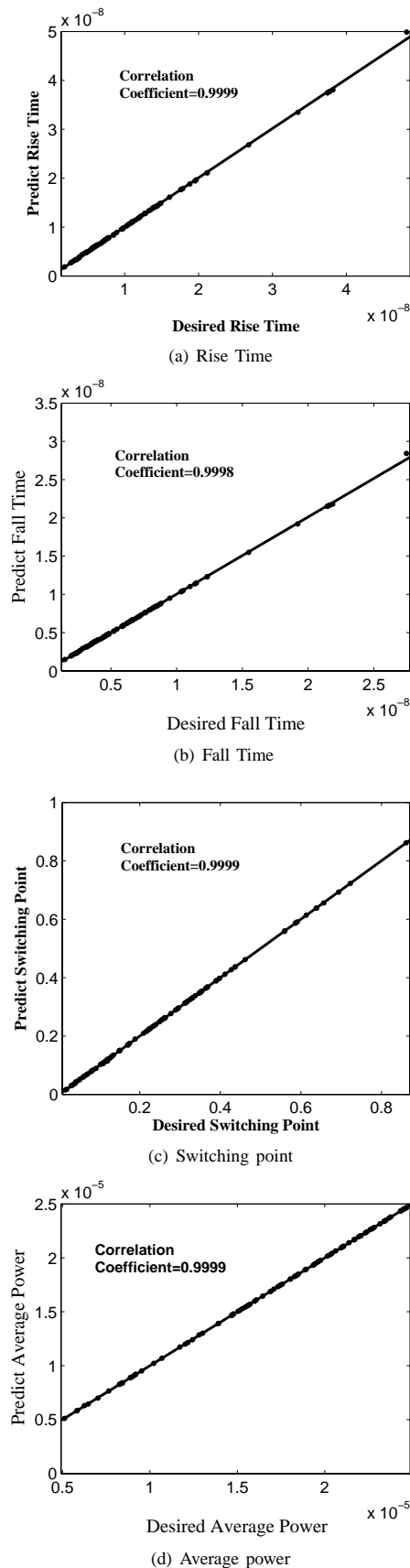


(c) Switching point



(d) Average power

Fig. 8. SPICE simulation versus ANN prediction output.



REFERENCES

- [1] Georges.G.E. Gielen and Rob.A. Rutenbar. Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits. *Proceedings of the IEEE*, Vol.88:pp.1825–1852, December 2000.
- [2] T. McConaghy and G. Gielen. Automation in Mixed-Signal Design: Challenges and Solutions in the Wake of the Nano Era. In *Proc. of ICCAD*, pages 461–463, November 2006.
- [3] Rob.A. Rutenbar, Georges.G.E. Gielen, and J.Roychowdhury. Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs. *Proceedings of the IEEE*, Vol.95:pp.640–669, March 2007.
- [4] W. Daems, G. Gielen, and W. Sansen. Simulation-Based Generation of Polynomial Performance Models for the Sizing of Analog Integrated Circuits. *IEEE Trans. CAD/IC*, Vol.22:pp.517–534, May 2003.
- [5] M. Avci and T. Yildirim. Neural Network Based MOS Transistor Geometry Decision for TSMC 0.18 Process Technology. In *Proc. of ICCS*, pages 615–622, 2006.
- [6] F. Gunes, F. Gurgen, and G. Torpi, H. Signal-noise neural network model for active microwave devices. *IEEE Proceedings Circuits Devices and Systems*, Vol.143:pp.1–8, 1996.
- [7] N. Kahraman and T. Yildirim. Technology Independent Circuit Sizing for Fundamental Analog Circuits using Artificial Neural Networks. In *Proc. of PRIME*, pages 1–4, 2008.
- [8] N. Kahraman and T. Yildirim. Technology independent circuit sizing for standard cell based design using neural network. *Digital Signal Processing*, Vol.19:pp.708–714, 2009.
- [9] F. Djefalla, M. Chahdib, A. Benhayaa, and M.L. Hafiane. An approach based on neural computation to simulate the nanoscale CMOS circuit. *Solid State Electronics*, Vol.51:pp.48–56, 2007.
- [10] S.K. Mandal, S. Sural, and A. Patra. ANN-and PSO-Based Synthesis of On-Chip Spiral Inductors for RF ICs. *IEEE Transaction CAD/IC*, Vol.27:pp.188–192, January 2008.
- [11] G.H. John and P. Langley. Static Versus Dynamic Sampling for Data Mining. In *Proc. of Knowledge Discovery and Data Mining*, 1996.
- [12] F. Provost, D. Jensen, and T. Oates. Efficient Progressive Sampling. In *Proc. of Knowledge Discovery and Data Mining*, pages 23–32, 1999.
- [13] C. Meek, B. Thiesson, and D. Heckerman. The Learning-Curve Sampling Method Applied to Model-Based Clustering. *Journal of Machine Learning Research*, Vol.2:pp.397–418, February 2002.
- [14] A. Satyanarayana and I. Davidson. A Dynamic Adaptive Sampling Algorithm for Real World Applications: Finger Print Recognition and Face Recognition. In *Proc. of ISMIS*, pages 631–640, 2005.
- [15] V.K. Devabhaktuni and Q.J. Zhang. Neural Network Training-Driven Adaptive Sampling Algorithm for Microwave Modeling. In *Proc. of European Microwave Conference*, 2000.
- [16] G. Wolfe and R. Vemuri. Adaptive Sampling and Modeling of Analog Circuit Performance Parameters with Pseudo-Cubic Splines. In *Proc. of ICCAD*, pages 931–936, 2004.
- [17] D. Dhabak and S. Pandit. Performance Modeling of Nano-scale CMOS Inverter using Artificial Neural Network. In *Proc. of IESPC*, pages 33–36, 2011.
- [18] Q.J. Zhang, K.C. Gupta, and V.K. Devabhaktuni. Artificial Neural Networks for RF and Microwave Design: From Theory to Practice. *IEEE Trans. MTT*, Vol.51:1339–1350, April 2003.
- [19] W. Zhao and Y. Cao. New Generation of Predictive Technology Model for Sub-45 nm Early Design Exploration. *IEEE Transactions Electron Devices*, Vol.53:pp.2816–2823, November 2006.
- [20] L. Kuipers and H. Niederreiter. *Uniform distribution of sequences*. Dover Publications.
- [21] I. C. Yeh. Modeling of Strength of High-Performance Concrete using Artificial Neural Network. *Cement and Concrete Research*, Vol.28:pp.1797–1808, 1998.
- [22] <http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>.

Fig. 9. Scatter diagram for ANN prediction output..