Machine Learning Based Power Estimation for CMOS VLSI Circuits

V. Govindaraj (govindaraj@ngp.ac.in) 
Dr NGP Institute of Technology

B. Arunadevi 
Dr NGP Institute of Technology

Research Article

Keywords: power estimation, Random Forest, BPNN, CMOS, VLSI

DOI: https://doi.org/10.21203/rs.3.rs-723965/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
MACHINE LEARNING BASED POWER ESTIMATION FOR CMOS VLSI CIRCUITS

Dr V.GOVINDARAJ* and Dr B.ARUNADEVI

*Assistant Professor, Dr NGP Institute of Technology, Coimbatore, Tamilnadu, India
Author Email: govindaraj@drngpit.ac.in

1 Professor, Dr NGP Institute of Technology, Coimbatore, Tamilnadu, India.

Abstract:

Nowadays, machine learning (ML) algorithms are receiving massive attention in most of the engineering application since it has capability in complex systems modelling using historical data. Estimation of power for CMOS VLSI circuit using various circuit attributes is proposed using passive machine learning based technique. The proposed method uses supervised learning method which provides a fast and accurate estimation of power without affecting the accuracy of the system. Power estimation using random forest algorithm is relatively new. Accurate estimation of power of CMOS VLSI circuits is estimated by using random forest model which is optimized and tuned by using multi-objective NSGA-II algorithm. It is inferred from the experimental results testing error varies from 1.4 percent to 6.8 percent and in terms of and Mean Square Error is 1.46e-06 in random forest method when compared to BPNN. Statistical estimation like coefficient of determination (R) and Root Mean Square Error (RMSE) are done and it is proven that random Forest is best choice for power estimation of CMOS VLSI circuits with high coefficient of determination of 0.99938. and low RMSE of 0.000116.

Keywords: power estimation; Random Forest; BPNN; CMOS; VLSI

1. Introduction

Due to recent development in VLSI technology million of transistors are fabricated in a single chip. Increase in number of transistor and operating speed in a chip and speed due which ultimately increase the power consumption and has become a crucial concern in submicron technology. In VLSI circuits power estimation at an earlier stage is highly needed, because it has a major impact on the reliability of VLSI circuits. Under this condition, at the higher levels of design abstraction average power estimation before a chip manufacturing is very much essential to calculate power budget and to take necessary steps to reduce power consumption. Hence less complex and low cost power estimation techniques is needed.

1.1 Previous work in power estimation

Power dissipation of the circuit depends on the inputs, frequency and operating voltage. Simulation and non simulation based method are two main categories to estimate average power. Amuru et al (2020) proposed an estimation technique for leakage power, delay in standard CMOS/FinFET digital cells. Bhanja and Ranganathan (2003) explained about switching activity estimation using Bayesian Networks considering both in internal nodes and inputs. Burch et al. (1993) and saxena et al (1997) proposed Simulation based method which includes Monte Carlo approach. Estimate the power for combinational and
sequential circuits was discussed by Buyukus et al (2006) and Kozhaya et al (2001). Adaptive neuro fuzzy application to power estimation was discussed by Govindaraj et al (2018). Kirei et al (2019) explained about estimation of power in CMOS integrated circuits with discrete time filters in which filter of various complexities are considered and power and area estimates are obtained. Ligang Hou et al (2006) proposed a method using Levenberg-Marquardt function to estimate power using Neural Network. Murugavel et al (2002) proposed a method using Least square estimation to estimate average power which minimize MSE through each iteration when compared to Monte Carlo approach. Under Probabilistic class, Najm et al. (1993) devised an algorithm to propagate of transition density values from the primary input to the primary output. BPNN and Radial Basis Function Neural Network (RBFNN) based power estimation of ISCAS’89 sequential Benchmark circuits have been performed by Ramanathan et al. (2013). Only two training functions Traingdm and Traingdx were used for analysis out of eleven different training algorithms available in MATLAB tool. Omnia S et al (2016) discussed about estimation of dynamic power consumption using circuit nodes and logic picture for CMOS combinational logic circuits. A. K. Shaw Halwai (2018) proposed a method to estimate power dissipation of phase detector circuit.

1.2 Related work using Random Forest

Chao Chen et al (2020) discussed about classification of neural activities, brain-Computer Interfaces, classification of finger gestures during motor execution and imagery. Estimation of Cumulative Pollution Index of Insulator Strings Leakage using random forest was proposed by de santos et al [2020]. Metal Oxide Surge Arrester based surface condition identification using random forest is done by Das AK et al [2020]. Impedance estimation in power system using machine learning is proposed by Liang et al [2020]. Pilarski et al 2012, S. Seyedzadeh et al 2018 Peng, H., et al 2017 developed human and robot integration, Monitoring of electric power grid and building energy management systems using machine learning. Air Pollution measurement using Machine Learning Techniques was discussed by srivatsava et al [2018]. A perspective on machine learning in turbulent flows was discussed by Sandeep Pandey et al (2020) similarly kanchan et al (2020) proposed Medical Internet of things using machine learning algorithms for lung cancer detection.

Machine Learning models will discover the relationship between input variables and outputs of interest from the system being studied or learn from measured data or simulated data that represents the physical problem. Nowadays, machine learning (ML) models are receiving enormous attention in most of the fields due to their capability in modeling complex systems using historical data. The use of data-driven models have been successfully demonstrated in applications demanding for real-time estimation of the targets values. The proposed work employs an Random forest algorithm which has the ability to estimate the power of CMOS VLSI circuits, without the knowledge on actual circuit structure and interconnections. The Random forest algorithm results are compared
with BPNN results. Error percentage for BPNN and Random Forest is calculated to find the deviation from actual power to predicted power in which Random Forest outperforms BPNN.

2. Training and testing data

The database used for training and testing the BPNN and random forest network is obtained from the literature (Ligang Hou 2006) which is shown in Tables 1 and 2. BPNN network is trained by using database of 20 Benchmark ISCAS’89 sequential circuits and the same circuits are used for testing purpose. The training and testing consists of attributes such as considered for sequential circuits are number of inputs, outputs, D flip-flops, inverters, total number of gates, AND gates, NAND gates, OR gates and NOR gates.

Table 1. ISCAS’89 benchmark circuit data set for training BPNN/RF (Ligang Hou et al. 2006)

| BENCHMARK CIRCUIT | GATE | AND | INV | NOR | NAND | OR | DFF | IN | OUT | Monte Carlo Simulation power in mw. |
|--------------------|------|-----|-----|-----|------|----|-----|----|-----|--------------------------------------|
| S208               | 66   | 21  | 38  | 16  | 15   | 14 | 8   | 10 | 1   | 0.00698                             |
| S298               | 75   | 31  | 44  | 19  | 9    | 16 | 14  | 3  | 6   | 0.00912                             |
| S349               | 104  | 44  | 57  | 31  | 19   | 10 | 15  | 9  | 11  | 0.01856                             |
| S386               | 118  | 83  | 41  | 0   | 0    | 35 | 6   | 7  | 7   | 0.00121                             |
| S400               | 106  | 11  | 58  | 34  | 36   | 25 | 21  | 3  | 6   | 0.01065                             |
| S420               | 160  | 49  | 78  | 34  | 29   | 28 | 16  | 18 | 1   | 0.00993                             |
| S444               | 119  | 13  | 62  | 34  | 58   | 14 | 21  | 3  | 6   | 0.01172                             |
| S713               | 139  | 94  | 254 | 0   | 28   | 17 | 19  | 35 | 23  | 0.03743                             |
| S820               | 256  | 76  | 33  | 66  | 54   | 60 | 5   | 18 | 19  | 0.02831                             |
| S838               | 288  | 105 | 158 | 34  | 58   | 14 | 21  | 3  | 6   | 0.01292                             |
| S953               | 311  | 49  | 84  | 112 | 114  | 36 | 29  | 16 | 23  | 0.02458                             |
| S1238              | 428  | 134 | 80  | 57  | 125  | 112| 18  | 14 | 14  | 0.06347                             |
| S1423              | 490  | 197 | 167 | 92  | 64   | 137| 74  | 17 | 5   | 0.07181                             |
| S1494              | 558  | 354 | 89  | 0   | 0    | 204| 6   | 8  | 19  | 0.06018                             |
| S5378              | 1004 | 0   | 1775| 765 | 0    | 239| 179 | 35 | 49  | 0.23357                             |
| S9234              | 2027 | 955 | 3570| 113 | 528  | 431| 228 | 19 | 22  | 0.28004                             |
| S15850             | 3448 | 1619| 6324| 151 | 968  | 710| 597 | 14 | 87  | 0.51991                             |
| S35932             | 12204| 4032| 3861| 0   | 7020 | 1152| 1728| 35 | 320 | 1.22048                             |
| S38417             | 8709 | 4154| 13470|2279| 2050| 226| 1636| 28 | 106 | 1.14518                             |
| S38584             | 11448| 5516| 7805| 12  | 278  | 1452| 2621| 1185|1.87987                             |

Table 2. ISCAS’89 benchmark circuit data set for testing BPNN/RF (Ligang Hou et al. 2006)

| BENCHMARK CIRCUIT | GATE | AND | INV | NOR | NAND | OR | DFF | IN | OUT |
|--------------------|------|-----|-----|-----|------|----|-----|----|-----|
| S344               | 101  | 44  | 59  | 30  | 18   | 9  | 15  | 9  | 11  |
| S382               | 99   | 11  | 59  | 34  | 30   | 24 | 21  | 3  | 6   |
| S641               | 107  | 90  | 272 | 0   | 4    | 13 | 19  | 35 | 24  |
| S1488              | 550  | 350 | 103 | 0   | 0    | 200| 6   | 8  | 19  |
| S13207             | 2573 | 1114| 5378| 98  | 849  | 512| 669 | 31 | 121 |
3. Power estimation using BPNN

The method consists of the following steps.

- Construction of neural network
- Training phase
- Testing phase

3.1 Construction of neural network

A back-propagation neural network is constructed with four-layer feed-forward. First layer is set with ‘linear’ transfer function and ‘tansig’ function is chosen for remaining layers. BPNN network Parameters such as learning rate, epoch, hidden layer and momentum constant are varied for different training algorithms such as Traingd, Trainscg, Trainlda, Trainfg, Trainrp, Trainoss, Traindx, Trainsgf, TrainGdm, TrainCgp and TrainCgb. The database of sequential circuits consists of nine attributes, therefore number of inputs for the network is considered is nine. The generalised architecture of BPNN is shown in Figure 1.

3.2 Training phase

1: Two-third of the input vectors from ISCAS’89 benchmark circuits database are extracted and to train the BPNN (Harris et al 1994).

2: Normalization is done for input vectors and their corresponding target vectors. Since the hidden layers has tan-sigmoidal activation function normalization is done between the −1 to +1.

3: The BPNN network is trained with the normalized input vectors and their corresponding normalized target vectors.

3.3 Testing phase

1: One-third of the input vectors from ISCAS’89 benchmark circuits database are used for testing (Harris et al 1994).

2: Before testing, the parameters used for input test vectors are normalized.

3: Outputs vectors are generated for these normalized test input vectors by the BPNN network.

4: Original value is obtained from Reverse normalization process.

![Figure 1 Architecture of BPNN](image-url)
4 Proposed Random forest based power estimation

In the proposed method, a Random Forest (RF) model is preferred due to its ability to predict multiple output values simultaneously. Until one record remains as subset, RF divides the records into smaller and smaller subsets since RF is an ensemble of randomized decision trees. The nodes and leaf nodes are called inner and final sets. RF needs a number of hyper-parameters to be set and to achieve maximum accuracy, these parameters are tuned as per simulated data to get accurate power estimation of CMOS VLSI Circuits. The constructed model is trained and tested using 10-fold cross-validation. Implementation is done using Python programming language, and the proposed work is carried out on a Computer with Intel Core i7-6700 3.4GHz CPU, 16GB RAM.

Figure 2 Work flow of RF Algorithm

5. Results and discussion

BPNN is a feed-forward back propagation type neural network consists of Learning rate, Momentum constant, activation function and training algorithm as four important parameter. Momentum constant is varied between 0.1 to 0.9, number of epochs is varied between 150 to 2700,
hidden layer neurons are selected between 10 to 17, 0.3 to 0.8 is the range of variation for learning rate for 11 different training algorithms and activation functions are chosen as tansig and logsig. Table 3 gives Regression results comparison of ISCAS’89 benchmark circuits using BPNN for 11 different training functions.

Table 3 Regression results comparison of ISCAS’89 benchmark circuits using BPNN

| Training Function | Layer Size | Epochs | Number of Input Attributes | Slope  | Y-intercept | Regression Value | Deviation in % | MSE          |
|-------------------|------------|--------|-----------------------------|--------|-------------|------------------|---------------|--------------|
| Trainscg          | 9:15:15:1  | 253    | 9                           | 1      | -0.0034     | 0.9999           | 0.01          | 6.254E-05   |
| Trainscgp         | 7:15:15:1  | 300    | 9                           | 1.8    | -0.025      | 0.9988           | 0.12          | 0.02672     |
| Trainsggb         | 9:16:15:1  | 100    | 9                           | 1.7    | -0.023      | 0.9993           | 0.07          | 0.02403     |
| Trainsoss         | 8:14:15:1  | 600    | 9                           | 0.99   | 0.0017      | 0.9991           | 0.09          | 0.01924     |
| Trainsbgf         | 9:15:15:1  | 500    | 9                           | 1.4    | -0.016      | 0.9993           | 0.07          | 0.01793     |
| Trainsgd          | 9:15:15:1  | 750    | 9                           | 1.8    | -0.027      | 0.9940           | 0.6           | 0.014096    |
| Trainsgdm         | 8:13:14:1  | 400    | 9                           | 0.015  | 0.0016      | 0.9920           | 0.8           | 0.02485     |
| Trainsgdx         | 8:14:15:1  | 225    | 9                           | 0.34   | 0.018       | 0.9979           | 0.21          | 0.01602     |
| Trainscgf         | 7:15:15:1  | 256    | 9                           | 1.1    | -0.0036     | 0.9990           | 0.1           | 0.00528     |
| Trainsgda         | 9:15:14:1  | 500    | 9                           | 0.58   | 0.011       | 0.9956           | 0.44          | 0.025339    |
| Trainsgp          | 8:16:15:1  | 250    | 9                           | 1      | -1.6x10^-5  | 0.9998           | 0.02          | 0.06346     |

Trainscg training function which is based on conjugate gradient algorithm provides better results for sequential circuits when compared to other training algorithms. From the regression analysis it is found that the results for ISCAS’89 sequential benchmark circuits deviates from ideal power by 0.01% to 0.8%. The trainscg training function gives the minimum regression value among other 11 training algorithms. Trainscg under conjugate gradient category is best suited for power estimation for sequential circuits for a layer size of 9:15:15:1 with 9 inputs and 253 epochs. From the Figure 3, it is inferred that trainscg predicts power very close to power estimator tool than remaining training algorithm.

Figure 3 ISCAS’89 benchmark circuits regression result comparison using BPNN for 11 training algorithms
Table 4 Mean square error of ISCAS’89 Benchmark circuits using BPNN/RF

| Benchmark Circuits | Actual Output | BPNN (MSE) | Random Forest (MSE) |
|--------------------|---------------|------------|---------------------|
|                    |               | Predicted Output | Squared Error | Predicted Output | Squared Error |
| S344               | 0.01846       | 0.0193      | 7.056E-07        | 0.0182          | 6.76E-08      |
| S382               | 0.01046       | 0.0176      | 5.098E-05        | 0.0103          | 2.56E-08      |
| S386               | 0.01628       | 0.0182      | 3.684E-06        | 0.0153          | 9.604E-07     |
| S400               | 0.01065       | 0.01087     | 4.84E-08         | 0.0104          | 6.25E-08      |
| S641               | 0.03629       | 0.0246      | 0.0001367        | 0.0338          | 6.2E-06       |

Mean Square Error: 3.842E-05, 1.46E-06

Predicted power output and Mean square error of ISCAS’89 Benchmark circuit is shown in Table 4, which is obtained by comparing actual circuit output with neural network and random forest predicted output. MSE is calculated for both BPNN and RF. MSE of RF is 1.46E-06 whereas BPNN is 3.842E-05. This shows ML models discover relationships between input variables and outputs of interest from the system being studied, learn from measured or simulated data. Figure 4 Comparison of ISCAS’89 benchmark circuits Actual output with RF and BPNN predicted output.

![Figure 4 Comparison of ISCAS’89 benchmark circuits Actual output with RF and BPNN predicted output.](image)
5.1 Prediction Error

Training and testing prediction error is calculated using equation (1)

\[ \text{Error} \% = \frac{A - B}{A} \times 100 \]  

(1)

where \( A \) is the actual value and \( B \) is the predicted output value by testing the network.

Prediction error comparison of BPNN with RF is reported in table 5. It is infer that RF prediction error is varying from 1.4% to 6.8% but BPNN prediction error is varying from 4.5% to 68.2%. RF prediction error is less than BPNN because it can deal with regression and classification problems of multiclass, small sample data, and without data pre-processing procedures. Figure 5 gives Comparison of Prediction error of RF and BPNN in which RF outperforms BPNN.

Table 5 Error calculation for BPNN and RF

| Benchmark Circuit | BPNN Error in (%) | Random Forest Error in (%) |
|-------------------|-------------------|---------------------------|
| S344              | 4.55038           | 1.408451                  |
| S382              | 68.26             | 1.529637                  |
| S386              | 11.7936           | 6.019656                  |
| S400              | 2.06573           | 2.347418                  |
| S641              | 32.21273          | 6.861394                  |

Figure 5 Comparison of Prediction error of RF and BPNN
5.2 Statistical Analysis of BPNN and RF

Performance of RF and BPNN network can be determined by doing statistical analysis such as Root Mean Square Error (RMSE) and Coefficient of determination (R) which is given below in Equation (2) – (3).

\[ RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Y_{i}^O - Y_{i}^C)^2}{N}} \]  

\[ R = \frac{\sum_{i=1}^{N}(Y_{i}^O - Y_{O})(Y_{i}^C - Y_{C})}{\sqrt{\sum_{i=1}^{N}(Y_{i}^O - Y_{O})^2} \sum_{i=1}^{N}(Y_{i}^C - Y_{C})^2} \]  

where \( Y^O \) is the mean of observed value, \( Y_{i}^O \) is the observed value, \( Y^C \) is the mean of calculated value, and \( Y_{i}^C \) is the calculated value. RMSE is used as a measure to find the difference between predicted values and measured values. RMSE can be used as an indicator of model accuracy or precision. A good network/model should have a low RMSE value, or as close to zero. The coefficient of determination \( R \) determines a linear correlation between measured values and values simulated by the model, optimal value is 1. RMSE and R for BPNN and ANFIS is shown in Table 6, from the table we infer that RMSE of RF is less when compared to BPNN and R value of RF is very close to 1. Considering these measures, the RF model achieves high accuracy for power estimation.

Table 6 Statistical analysis test on BPNN and RF

| Parameter | BPNN     | Random Forest |
|-----------|----------|---------------|
| RMSE      | 0.0004499| 0.000116      |
| R         | 0.84696  | 0.99938       |

6. Conclusion

In this paper, a supervised learning method to estimate the Power of CMOS VLSI circuits is presented. The proposed RF model is an alternative approach to the conventional techniques like SPICE circuit simulation which are based on the assumption of predefined empirical equations and arbitrary parameters. The results of RF are highly accurate and the details of circuit structure and interconnections are not required. Random Forest is less computationally than BPNN. A random forest for a decision tree will give a different interpretation but with better performance on the other hand BPNN requires much more data for the result to be effective. Hence, RF with exclusive characteristics appears as a better choice for estimation of power in CMOS VLSI circuits. It is
proven that by using statistical estimation like coefficient of determination and Root Mean Square Error (RMSE), RF performs well for power estimation application with high coefficient of determination of 0.99938 and low RMSE of 0.000116.

Conflict of Interest: Author don’t have any conflict of interest in submitting the paper to this journal.

REFERENCES

[1] Amuru, M. S. Ahmed and Z. Abbas (2020). An Efficient Gradient Boosting Approach for PVT Aware Estimation of Leakage Power and Propagation Delay in CMOS/FinFET Digital Cells. IEEE International Symposium on Circuits and Systems (ISCAS), Sevilla, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180600
[2] Bai, Y. Li, X. Wang, J. Xie and C. Li (2016). Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological conditions”, Atmos. Pollut. Res., vol. 7, no. 3, pp. 557-566, 2016
[3] Bhanja, S., & Ranganathan, N. (2003). Switching activity estimation of VLSI circuits using Bayesian networks. IEEE Transactions on Very Large Scale Integration Systems, 11(4), 558–567. doi:10.1109/TVLSI.2003.816144
[4] Burch, R., Najm, F. N., Yang, P., & Trick, T. N. (1993). A Monte Carlo approach for power estimation. IEEE Transactions on Very Large Scale Integration Systems, 2(1), 63–71. doi:10.1109/92.219908
[5] Buyuks, M. K., & Najm, F. N. (2006). Early power estimation for VLSI circuits. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 24(7), 1076–1088. doi:10.1109/TCAD.2005.859004.
[6] Chao Chen, Peiji Chen, Abdelkader Nasreddine Belkacem, Lin Lu, Rui Xu, Wenjun Tan, Penghai Li, Qiang Gao, Duk Shin, Changming Wang & Dong Ming (2020). Neural activities classification of left and right finger gestures during motor execution and motor imagery, Brain-Computer Interfaces. DOI: 10.1080/2326263X.2020.1782124.
[7] de Santos H and M. Á. Sanz-Bobi(2020). A Cumulative Pollution Index for the Estimation of the Leakage Current on Insulator Strings. IEEE Transactions on Power Delivery, vol. 35, no. 5, pp. 2438-2446, doi: 10.1109/TPWRD.2020.2968556.
[8] Das A, K. S. Dalai and B. Chatterjee, "Cross Stockwell transform aided Random Forest based surface condition identification of Metal Oxide Surge Arrester employing Leakage current signal," 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 1775-1778, doi: 10.1109/TENSYMP50017.2020.9230802.
[9] Govindaraj, V & Ramesh, J (2018). Adaptive neuro fuzzy inference system-based power estimation method for CMOS VLSI circuits. International Journal of Electronics, Taylor and Francis,, vol. 105, no. 3, pp. 398-411.
[10] Harris, C. J. (1994). Advances in intelligent control. Boca Raton, FL: CRC Press. ISBN 9780748400696.
[11] Hou, L., Zheng, L., & Wu, W. (2006). Neural Network Based VLSI Power Estimation. 2006 8th International Conference on Solid-State and Integrated Circuit Technology Proceedings. doi:10.1109/icsict.2006.306506
[12] Kanchan Pradhan & Priyanka Chawla (2020). Medical Internet of things using machine learning algorithms for lung cancer detection. Journal of Management Analytics, 7:4, 591-623, DOI: 10.1080/23270012.2020.1811789
[13] Kirei, S C. Farcas and M. D. Topa (2019). Power and Area Estimation of Discrete Filters in CMOS Integrated Circuits. Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Poznan, Poland, 2019, pp. 67-70, doi: 10.23919/SPA.2019.8936762.
[14] Klein, F., Leao, R., Arajo, G., Santos, L., & Azevedo, R. (2009). A multi-model engine for high-level power estimation accuracy optimization. IEEE Transactions on Very Large Scale Integration Systems, 17(5), 660–673. doi:10.1109/TVLSI.2009.2013627
[15] Kle K. Shaw Halwai (2020). Estimation of Power and Delay of CMOS Phase Detector and Phase-Frequency Detector Using Nano Dimensional MOS Transistor. IEEE VLSI DEVICE CIRCUIT AND SYSTEM (VLSI DCS), Kolkata, India, 2020, pp. 294-298, doi: 10.1109/VLSIDCS47293.2020.9179921.
[16] Kozhaya, J. N., & Najm, F. N. (2001). Power estimation for large sequential circuits. IEEE Transactions on Very Large Scale Integration Systems, 9(2), 400–407. doi:10.1109/ICCAD.1997.643581.

[17] Liang, H. Guo and T. Zheng(2019).Real-Time Impedance Estimation for Power Line Communication. IEEE Access, vol. 7, pp. 88107-88115, 2019, doi: 10.1109/ACCESS.2019.2925464.

[18] Murugavel, A. K., Ranganathan, N., Chandramouli, R., & Chavali, S. (2002). Least-square estimation of average power in digital CMOS circuits. IEEE Transactions on Very Large Scale Integration Systems, 10(1), 55–58. doi:10.1109/92.988730

[19] Peng, H., Lima, A.R., Teakles, A (2017). Evaluating hourly air quality forecasting in Canada with nonlinear updatable machine learning methods. Air Qual. Atmos. Hlth. No. 10, Issue 2, pp 195–211, March 2017.

[20] Pilarski, M. R. Dawson, T. Degris, J. P. Carey, and R. S. Sutton (2012). Dynamic switching and real-time machine learning for improved human control of assistive biomedical robots. 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Rome, Italy, June 2012, pp. 296-302.

[21] Ramanathan, P., Surendiran, B., & Vanathi, P. T. (2013). Power estimation of benchmark circuits using artificial neural networks. Pensee Journal, 75(9), 427–433.

[22] Saxena, V., Najm, F. N., & Hajj, I. N. (1997, March 17–20). Monte-Carlo approach for power estimation in sequential circuits. European Design and Test conference, (ED&TC’97), pp. 416–420. doi:10.1109/EDTC.1997.582393

[23] Sandeep Pandey, Jörg Schumacher & Katepalli R. Sreenivasan (2020) A perspective on machine learning in turbulent flows. Journal of Turbulence, 21:9-10, 567-584, DOI: 10.1080/14685248.2020.1757685.

[24] Seyedzadeh, F. Pour Rahimian, P. Rastogi, and I. Glesk (2020). Tuning machine learning models for prediction of building energy loads. Sustainable Cities and Society, 47, p. 101484, 2019/05/01/2019.

[25] Seyedzadeh, F. P. Rahimian, I. Glesk, and M. Roper, "Machine learning for estimation of building energy consumption and performance: a review," Visualization in Engineering, 6, 5.

[26] Srivastava, S. Singh and A. P. Singh (2018). Estimation of Air Pollution in Delhi Using Machine Learning Techniques. International Conference on Computing, Power and Communication Technologies (GUCON), 2018, pp. 304-309, doi: 10.1109