International Research Journal of Multidisciplinary Scope (IRJMS), 2024; 5(2): 745-753

IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2024.v05i02.0633

Estimating R, L and C in Arithmetic Circuits using ML

Saravanakumar C*, Usha Bhanu N

Department of Electronics and Communication Engineering, SRM Valliammai Engineering College, Kattankulathur – 603 203, Tamilnadu, India. *Corresponding Author's Email: saravanakumarc.ece@srmvalliammai.ac.in

Abstract

In Very Large Scale Integration (VLSI) circuits, the estimation techniques for resistors (R), inductors (L), and capacitors (C) heavily rely on segmented circuit analysis, which involves usage of complex mathematical simplification models. These methods have been conventionally applied to estimate the behavior of circuits, but when faced with systems featuring unique circuit architectures, they often encounter inaccuracies and limitations. The significance of adders as fundamental building blocks in intricate circuit design cannot be overstated. In such complex circuits, various parameters, including parasitic resistances, inductances, and capacitances, engage an indispensable effort in the analysis of delays and performance. However, the conventional estimation methods fail to address the complexities of these systems and the impacts of parasitics accurately. To overcome these challenges, this paper proposes a novel approach that harnesses the potential of machine learning algorithms. By integrating machine learning into the estimation process, the research aims to achieve specific and precise analysis methods that can cater to the needs of modern VLSI circuits. The proposed methodology involves collecting a comprehensive dataset using different adder circuits. From each individual adder circuit layout, the relevant information on resistors, capacitors, and inductors is carefully extracted and compiled. This dataset serves as the foundation for training the machine learning models. Three standard machine learning models are employed in this study: adaboost, Tree, and k-Nearest Neighbors (kNN). Their task is to predict the values of resistors, inductors, and capacitors based on the input data from the adder circuits. Among these models, adaboost proves to be the most effective, exhibiting superior performance by achieving a reduced root mean square error of about 0.008. When compared to the Tree and kNN models, adaboost stands out as the optimal choice for accurately estimating the R values in VLSI circuits.

Keywords: AdaBoost, Arithmetic Circuits, Estimation, Machine Learning, VLSI.

Introduction

The continuous advancement of integrated circuit technology has revolutionized the field of electronics and digital systems. Very Large Scale Integration (VLSI) circuits have become the backbone of modern-day electronic devices, facilitating the integration of millions of transistors on a single chip. As the complexity of VLSI circuits increases, accurate estimation of resistors (R), inductors (L), and capacitors (C) has become imperative to ensure optimal circuit performance and reliability. Estimation techniques for R, L, and C have long been a subject of intense research in the VLSI community. Traditional approaches involve segmented circuit analysis, employing complex mathematical simplification models. While these methods have been effective in analyzing standard circuit configurations (1-4), they often fail to address the complexities associated with systems featuring unique circuit architectures. The presence of parasitic resistances, inductances, and capacitances significantly impacts circuit behavior, causing inaccuracies and limitations in the estimation process.

Segmented circuit analysis has been the conventional approach for estimating R, L, and C in VLSI circuits. This method involves breaking down the circuit into smaller segments and employing mathematical simplification models to analyze the behavior of each segment. While this approach has provided valuable insights into circuit characteristics, it falls short when dealing with non-standard circuit configurations and parasitic effects.

Parasitic elements such as parasitic resistances, inductances, and capacitances arise due to the physical layout of the circuit and the interactions between different components. These parasitic effects can significantly impact circuit performance and introduce inaccuracies in the estimation of R, L, and C values. The inaccuracies stemming from traditional estimation techniques pose substantial obstacles in the realm of modern VLSI circuit

(Received 18th February 2024; Accepted 27th April 2024; Published 30th April 2024)

design. In an era where the need for smaller, faster, and more energy-efficient electronic devices is ever-increasing, the precise estimation of parameters such as resistors (R), inductors (L), and capacitors (C) becomes even more imperative. Furthermore, the increasing complexity of VLSI circuits necessitates the development of more sophisticated analysis methods that can adapt to unique circuit architectures.

Machine learning algorithms have shown tremendous potential in diversified domain, inclusive of natural language processing, speech processing, and pattern recognition. The application of machine learning in VLSI circuit analysis presents a promising avenue for overcoming the limitations of traditional estimation techniques (5-7). By harnessing the power of machine learning, it is possible to develop specific and precise analysis methods that can effectively account for parasitic effects and unique circuit configurations.

The primary contribution of this paper are as listed below

- Integrating machine learning algorithms into the estimation process for resistors, inductors, and capacitors in VLSI circuits.
- Demonstrate the adaptability of machine learning models to different circuit configurations and design requirements.

A wide variety of full adder circuits are constructed and simulated using Microwind and Dsch Tool. The logical equations governing the full adders are listed below as equation [1] and equation [2].

$$Opt\_S = (Ipt\_A \wedge Ipt\_B \wedge Ipt\_C) \quad [1]$$

$$Opt\_C = (Ipt\_A.Ipt\_B) + (Ipt\_B.Ipt\_C) + (Ipt\_C.Ipt\_A) \quad [2]$$

In the above equations ^ symbol represents XOR operation. The schematic of the circuits under consideration is given in Fig. 1.

Full Adder Employing Half Adders (FAEHA)

A full adder is a crucial combinational logic in digital logic circuits, responsible for performing binary addition of 3 inputs – Ipt_A, Ipt_B and a input carry (Ipt_C) - to produce 2 number of yields: the sum (Opt_S) and the carry-out (Opt_C). To construct a full adder, we can employ an OR gate and 2 numbers of single bit adders. The first half adder is used to add inputs A and B, yielding a sum output (S1) and a carry-out output (C_out1) (8). Subsequently, the second half adder combines the carry-out output (C_out1) from the first half adder with the carry input (Ipt_C) and produces another sum output (S2) and carry-out output (C_out2). To generate the final sum (Opt_S) output, we take the XOR of the sum output (S2) from the next half adder and the carry input (Ipt_C). As for the final carry-out (Opt_C) output, it is obtained by performing an OR operation on the carry-out output (C_out2) from the next half adder and the carry-out output (C_out1) from the initial half adder.

Full Adder Employing MUX and XOR (FAEMX)

To construct a full adder using XOR gates and multiplexer (MUX), the sum (Opt_S) output is designed initially. The XOR gate is utilized to

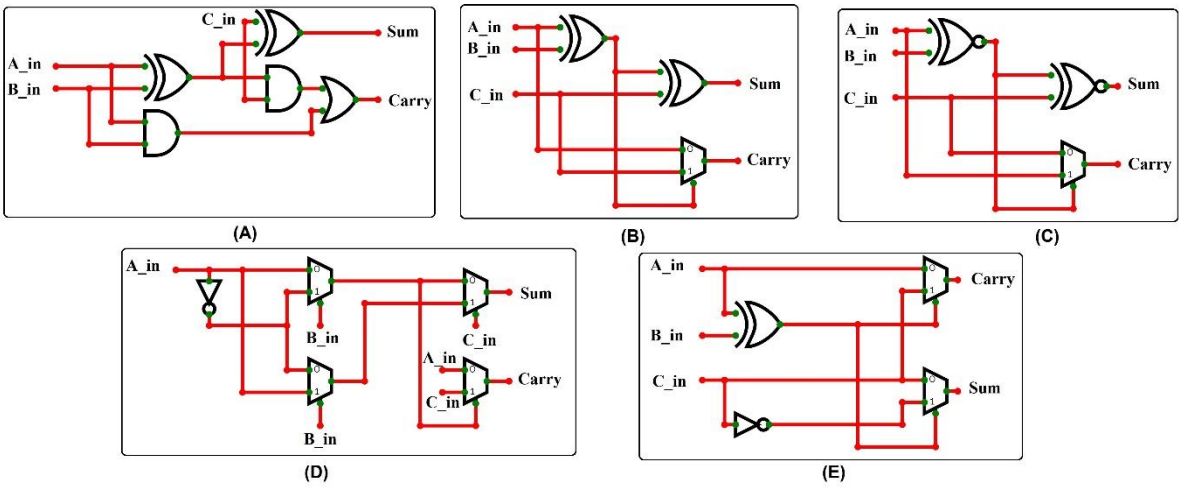

Figure 1: (A)FAEHA, (B) FAEMX, (C) FAEMXN, (D) FAEM; (E) FAEXMI

compute the bitwise sum of inputs Ipt_A, Ipt_B, and Ipt_C (9). The output of the XOR gate represents the sum bit. Next, the carry-out (Opt_C) output is accomplished using MUX, which selects the carry input (Ipt_C) or a combination of the carry input (Ipt_C) and the AND operation between Ipt_A and Ipt_B, based on the inputs Ipt_A and Ipt_B. The MUX acts as a switch that chooses between the carry input (Ipt_C) and the carry generated by Ipt_A and Ipt_B, depending on whether Ipt_A and Ipt_B are both 1 (indicating a carry). The output of the MUX represents the carry-out (Opt_C) bit. By integrating XOR gates and MUX, a full adder circuit can be constructed with minimal components, making it an efficient and widely-used method for binary addition in digital systems. This approach demonstrates the versatility of XOR gates and MUX in designing complex circuits, ensuring accurate computation of sums and carry-outs, critical for various arithmetic operations in computers and microprocessors.

Full Adder Employing MUX and XNOR (FAEMXN)

To construct a full adder using XNOR gates and MUX, the design procedure is started with the sum (S) output. The XNOR gate is employed to compute the bitwise sum of inputs Ipt_A, Ipt_B, and Ipt_C. The output of the XNOR gate represents the sum bit. To design the carry-out (Opt_C) output. MUX is instrumental in this step, as it selects the carry input (Ipt_C) or a combination of the carry input (Ipt_C) and the OR operation between Ipt_A and Ipt_B, depending on the values of Ipt_A and Ipt_B (10). The MUX acts as a switch, choosing between the carry input (Ipt_C) and the carry generated by Ipt_A and Ipt_B, considering whether Ipt_A and Ipt_B are both 0 (indicating no carry) or not. The output of the MUX represents the carry-out (Opt_C) bit. The integration of XNOR gates and MUX in the full adder circuit demonstrates an efficient and resource-effective approach to binary addition in digital systems (11).

Full Adder Employing MUX (FAEM)

Hussain (12) introduced a novel full adder circuit design that relies on the gate diffusion input (GDI) concepts and the switching patterns of the inputs. The adder's construction comprises two stages: an XOR-XNOR module in the first stage, followed by the last phase for producing the expected outputs. Leveraging the input switching patterns and employing GDI methods for every blocks, the circuit achieves minimized switching activities of the transistors. As a result, this design leads to significant improvements in power consumption, computational complexity and delay. The first stage of the full adder is the XOR-XNOR module, where the XOR gate is used to calculate the bitwise sum of the inputs, and the XNOR gate is employed to handle the carry bits. This intermediate stage plays a crucial role in preparing the data for the subsequent stage. In the final stage, the circuit processes the results from the XOR-XNOR module and generates the desired sum (Opt_S) and carry-out (Opt_C) outputs. By reducing the switching activities of the transistors through input switching activity pattern analysis and GDI techniques, the full adder circuit demonstrates improved performance metrics (13). The minimized switching activities lead to reduced propagation delays, enhancing the overall speed and efficiency of the full adder. Moreover, the reduction in power consumption results in energy-efficient operation, making it favorable for low-power applications. Furthermore, the computational complexity reduction simplifies the circuit design and facilitates easier integration into larger systems.

Full Adder Employing XOR, MUX and Inverter (FAEXMI)

The proposed design of a new full adder centers around the utilization of a XOR gate, two 2X1 multiplexers, and one CMOS inverter. The primary focus in developing this circuit lies in achieving two key objectives: minimizing power consumption and reducing the overall size of the full adder. To achieve the goal of minimum power consumption, careful consideration is given to the selection of components and the arrangement of the circuitry. By employing low-power devices and optimizing the switching activities of the transistors, the full adder design strives to significantly reduce power requirements while maintaining reliable operation. Simultaneously, the emphasis on creating a compact full adder size is of paramount importance (14). This objective involves efficient placement and integration of the XOR gate, multiplexers, and CMOS inverter within a limited chip area. Through careful layout and design techniques, the new full adder aims to occupy a smaller physical footprint, making it suitable for modern integrated circuit applications. By combining these two essential design

objectives, the proposed full adder aspires to offer a compelling solution for power-sensitive and space-constrained scenarios. The potential benefits of this design lie in its ability to contribute to energy-efficient computing systems and pave the way for higher-density circuit integration. Moreover, the compact size ensures effective utilization of valuable chip real estate, enabling more complex circuitry and functionality in VLSI chips (15).

Methodology

Dataset Collection

The initial footstep in the proposed methodology involves gathering a comprehensive dataset of adder circuits. Adders are fundamental components in VLSI circuits, widely used in arithmetic and logic operations. A diverse set of adder circuits is selected as discussed in section 2, to cover various circuit architectures and design complexities. Each adder circuit is simulated, and relevant information about the resistors, inductors, and capacitors is extracted.

For data collection, commercial VLSI design tools the Microwind and Dsch are utilized to generate circuit layouts and perform simulations. These simulations involve applying various test cases and stimulus signals to the adder circuits to observe their responses. The simulations record the circuit characteristics, such as delay, power consumption, and parasitic effects, which are used as features for training the machine learning models. The Table 1 shows the generated data set for the different adder circuits with its R, L and C Values.

Table 1. Generation of dataset

Adder	Node	C(fF)	R(kohms)	L(nH)	Power(uW)
FAEHA	Ipt_A	3.89	0.542	0.02	12.03
FAEHA	Ipt_B	3.87	0.582	0.02	12.03
FAEHA	Ipt_C	3.98	0.583	0.02	12.03
FAEHA	Opt_S	1.15	0.18	0.01	12.03
FAEHA	Opt_C	1.3	0.182	0.02	12.03
FAEMX	Ipt_A	3	0.556	0.02	9.272
FAEMX	Ipt_B	1.73	0.291	0.01	9.272
FAEMX	Ipt_C	3.07	0.481	0.03	9.272
FAEMX	Opt_S	0.87	0.18	0.01	9.272
FAEMX	Opt_C	2.51	0.379	0.03	9.272
FAEM	Ipt_A	2.89	0.555	0.02	10.026
FAEM	Ipt_B	1.73	0.291	0.01	10.026
FAEM	Ipt_C	4.32	0.555	0.02	10.026
FAEM	Opt_S	2.66	0.38	0.04	10.026
FAEM	Opt_C	2.22	0.378	0.03	10.026
FAEMXN	Ipt_A	3.26	0.623	0.03	7.395
FAEMXN	Ipt_B	1.73	0.291	0.01	7.395
FAEMXN	Ipt_C	2.97	0.48	0.02	7.395
FAEMXN	Opt_S	0.87	0.18	0.01	7.395
FAEMXN	Opt_C	2.51	0.379	0.03	7.395
FAEXMI	Ipt_A	5.31	0.742	0.05	8.723
FAEXMI	Ipt_B	7.77	1.629	0.04	8.723
FAEXMI	Ipt_C	5.74	0.717	0.05	8.723
FAEXMI	Opt_S	2.78	0.381	0.04	8.723
FAEXMI	Opt_C	3.1	0.383	0.04	8.723

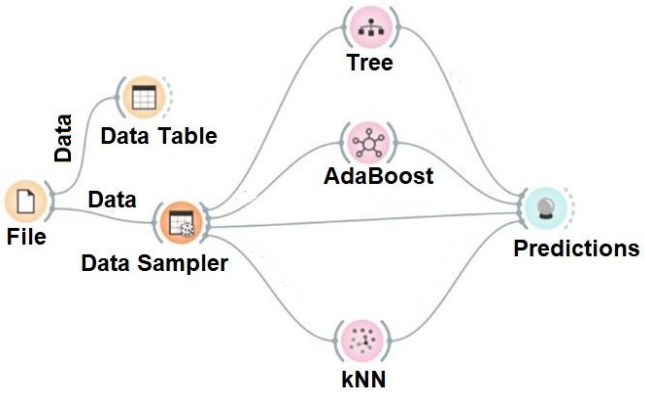

Figure 2: Simulation in Orange Tool

Feature Engineering

Feature engineering is a crucial step in preparing the dataset for machine learning. The collected simulation data contains a multitude of features, including parasitic resistances (R), inductances (L), and capacitances (C), as well as other circuit parameters. It is essential to preprocess and select the most relevant features to reduce dimensionality and enhance model performance.

During feature engineering, data normalization and scaling techniques are applied to bring all features within a standardized range. Additionally, feature selection methods, such as correlation analysis and feature importance ranking, are employed to identify the most influential features for R, L, and C estimation.

Machine Learning Model Selection

In this step, various machine learning algorithms are considered for the estimation of resistors, inductors, and capacitors in VLSI circuits. Standard machine learning models, including adaboost, Decision Trees, and k-Nearest Neighbors (kNN), are evaluated for their performance on the preprocessed dataset.

The selection process from the dataset involves dividing into training and testing sets. The dataset selected for training is chosen to train the machine learning models on the relevant features and corresponding R, L, and C values. The testing set is then employed to assess the accomplishment of the models and generalization ability. The simulation environment in Orange Data Mining tool is displayed in Fig.2.

Model Training and Optimization

The selected machine learning models are trained using the training dataset. During the training process, hyperparameters of the models are fine-tuned to achieve the best possible performance. Procedures like grid search and cross-validation are applied to optimize the hyperparameters and prevent overfitting.

To assess the accuracy of the trained models, metrics like Root Mean Square Error (RMSE), and Mean Square Error (MSE) are calculated using the equation [3] and equation [4].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{DP} \|S(i) - \hat{S}(i)\|^2}{DP}} \quad [3]$$

$$MSE = \sqrt{\frac{\sum_{i=1}^{DP} (S(i) - \hat{S}(i))^2}{DP}} \quad [4]$$

Where DP is the total count of data, S(i) represents the calculated sample value at i^{th} iteration, $\hat{S}(i)$ is its respective predicted value.

The model with the lowest error values on the testing set is considered the most suitable for R, L, and C estimation in VLSI circuits.

Model Evaluation and Comparison

Once the models are trained and optimized, they are evaluated on the testing dataset to assess their performance in estimating R, L, and C values. The evaluation involves comparing the predicted values to the ground truth values obtained from circuit simulations. The model with the highest accuracy and the lowest error values is identified as the most effective for VLSI circuit analysis.

Results and Discussion

The machine learning models are developed with the Orange Data mining Tool installed in a Dell Computer System with intel i5 processor running on Windows OS. The developed data set of the adder circuits with different parameters are used as input dataset from Table 1. In Table 4, the estimated values of resistance (R), Capacitance (C) and inductance (L) are showcased, derived from three different predictive models: adaboost, Tree, and kNN. Each of these models was employed to

analyze the circuit's characteristics and predict the R, C and L values. It serves as a valuable reference, shedding light on the estimated inductance values achieved through the adaboost, Tree, and kNN models. The presentation of results in Table 2 is instrumental in understanding the strengths and limitations of each model in accurately predicting inductance. It should be noted that the data inputs are shuffled to strengthen the prediction accuracy. In statistics and machine learning, the MSE is a regularly used measure to assess the efficacy and efficiency of predictive models. It calculates the average of the squared value of difference between a dataset's actual (ground truth) values and anticipated values. The predicted values are produced from the model, and the squared difference between each anticipated value and the appropriate real counterpart is then calculated to provide the MSE. The MSE value is then calculated by averaging the squared differences among each of the dataset's data values. A lower MSE signifies greater accuracy and precision since the model's predictions are more closely aligned with the measured data. From the graph in Fig. 3 shows that the adaboost model over performs the other models.

Table 2. Estimation of R, C and L values using machine learning model

Adder	Node	R(K)				C(fF)				L(nH)			
		Actual	Predicted			Actual	Predicted			Actual	Predicted		
			Ada	Tree	KNN		Ada	Tree	KNN		Ada	Tree	KNN
FAEMX	Opt_S	0.18	0.18	0.18	0.304	0.87	0.87	1.05	2.24	0.01	0.01	0.01	0.02
FAEMXN	Ipt_B	0.291	0.291	0.291	0.391	1.73	1.73	1.85	2.27	0.01	0.01	0.01	0.02
FAEHA	Ipt_A	0.542	0.582	0.566	0.563	3.89	3.61	3.91	2.84	0.02	0.02	0.02	0.02
FAEXMI	Opt_S	0.381	0.381	0.381	0.436	2.78	2.78	2.61	3.89	0.04	0.04	0.04	0.03
FAEM	Ipt_B	0.291	0.291	0.291	0.344	1.73	1.73	1.85	2.76	0.01	0.01	0.01	0.02
FAEMX	Opt_C	0.379	0.379	0.379	0.435	2.51	2.51	2.61	2.24	0.03	0.03	0.03	0.03
FAEM	Opt_S	0.38	0.381	0.381	0.45	2.66	2.66	2.61	2.76	0.04	0.04	0.04	0.03
FAEHA	Ipt_B	0.582	0.582	0.566	0.563	3.87	3.87	3.91	2.84	0.02	0.02	0.02	0.02
FAEXMI	Ipt_C	0.717	0.717	1.173	0.805	5.74	5.74	6.27	3.99	0.05	0.05	0.05	0.04
FAEMX	Ipt_A	0.556	0.556	0.531	0.436	3	3	3.13	2.24	0.02	0.02	0.03	0.03
FAEHA	Ipt_C	0.583	0.583	0.566	0.563	3.98	3.87	3.91	2.84	0.02	0.02	0.02	0.02
FAEM	Ipt_C	0.555	0.555	0.566	0.578	4.32	3.61	3.61	2.76	0.02	0.02	0.02	0.03
FAEMXN	Ipt_A	0.623	0.623	0.682	0.449	3.26	3.26	3.13	2.27	0.03	0.03	0.03	0.03
FAEHA	Opt_S	0.18	0.182	0.18	0.282	1.15	1.15	1.05	2.84	0.01	0.01	0.01	0.02
FAEHA	Opt_C	0.182	0.182	0.18	0.282	1.3	1.3	1.05	2.84	0.02	0.02	0.01	0.02
FAEXMI	Ipt_A	0.742	0.742	0.682	0.576	5.31	5.31	6.27	3.99	0.05	0.05	0.05	0.04
FAEMXN	Ipt_C	0.48	0.48	0.431	0.431	2.97	2.97	3.05	2.27	0.02	0.02	0.03	0.03
FAEXMI	Ipt_B	1.629	1.629	1.173	0.805	7.77	7.77	6.27	4.98	0.04	0.04	0.05	0.04
FAEMXN	Opt_S	0.18	0.18	0.18	0.264	0.87	0.87	1.05	2.27	0.01	0.01	0.01	0.01
FAEXMI	Opt_C	0.383	0.383	0.431	0.436	3.1	3.1	3.05	3.89	0.04	0.04	0.04	0.03
FAEMX	Ipt_C	0.481	0.481	0.531	0.436	3.07	3.07	3.05	2.24	0.03	0.03	0.03	0.03
FAEM	Ipt_A	0.555	0.555	0.531	0.47	2.89	3.61	3.61	2.76	0.02	0.02	0.02	0.03
FAEM	Opt_C	0.378	0.378	0.379	0.397	2.22	2.22	1.85	2.76	0.03	0.03	0.03	0.03
FAEMXN	Opt_C	0.379	0.379	0.379	0.431	2.51	2.66	2.61	2.27	0.03	0.03	0.03	0.03
FAEMX	Ipt_B	0.291	0.291	0.291	0.304	1.73	1.73	1.85	2.24	0.01	0.01	0.01	0.02

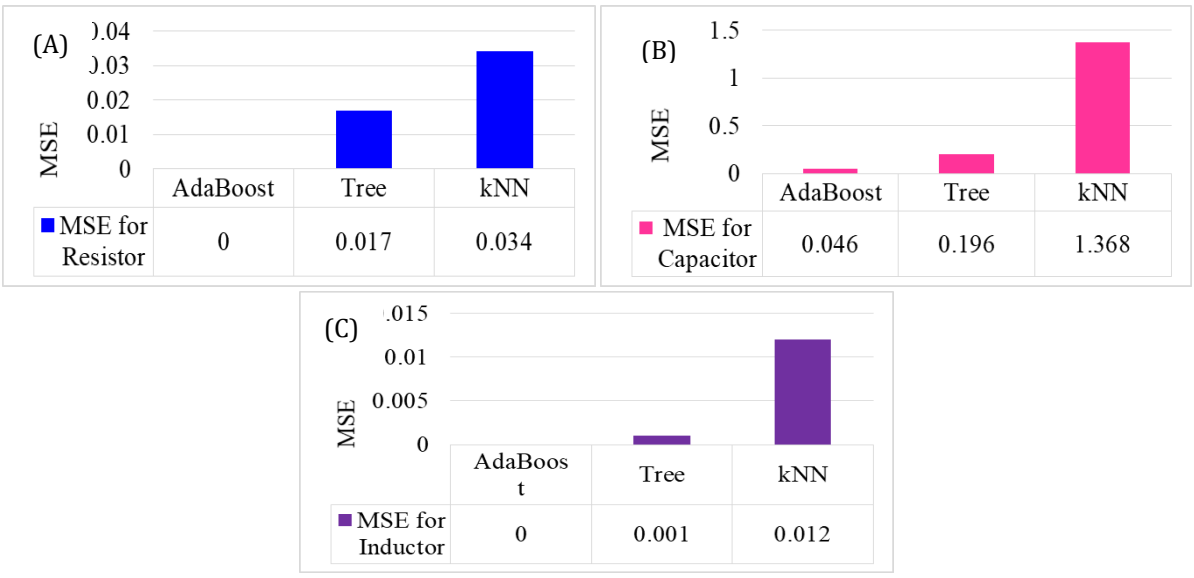

Figure 3: Mean Square Error of Three models for R (A), C (B) and L (C) estimation

Root Mean Square Error is a metric that is widely employed to assess the accuracy and performance of predictive models, particularly in regression tasks. It is a variation of the Mean Square Error that addresses some of its limitations, making it a preferred choice for evaluating model predictions. The square root of the average of the squared discrepancies among the predicted values and the matching actual (ground truth) values in a dataset is employed to figure out the RMSE. By taking the square root, RMSE brings the metric back to the original scale of the data, which is especially beneficial when interpreting the error in real-world units or when comparing multiple models. Like MSE, a lesser RMSE value implies that the predictions made by the model are accurate towards the true values, signifying higher accuracy and precision. However, RMSE has the added advantage of being less sensitive to outliers or extreme values in the dataset due to the square root operation, as it dampens the impact of large errors. For the developed model the RMSE is calculated as depicted in Fig. 4.

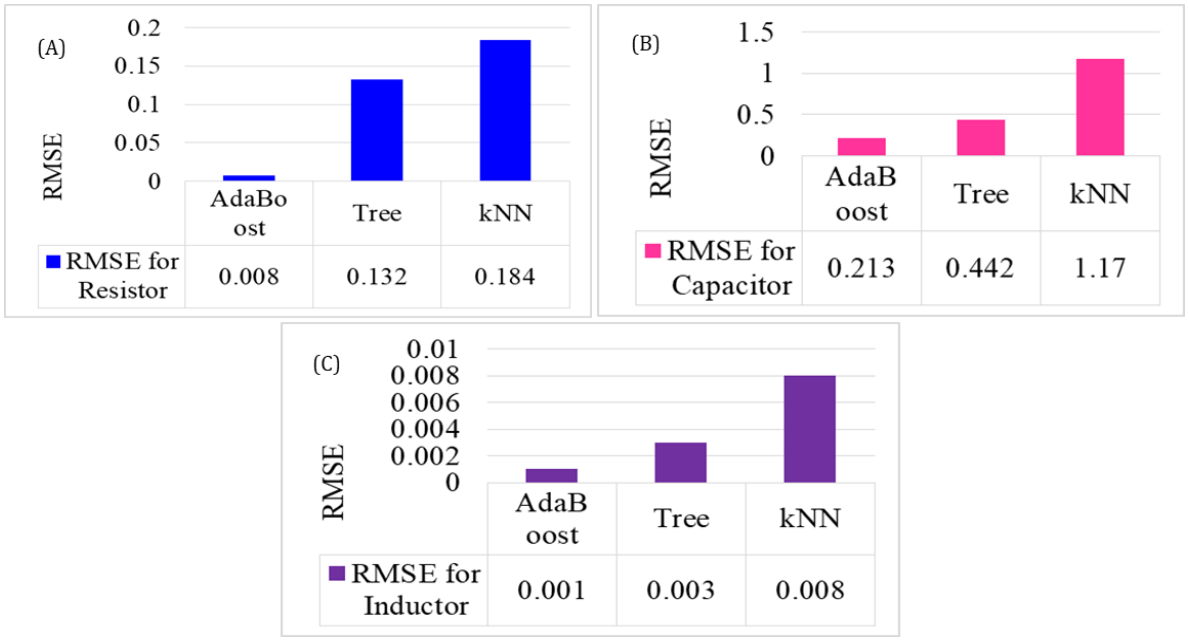

Figure 4: Root Mean Square Error of Three models for R(A), C(B) and L(C) estimation

Conclusion

The integration of machine learning algorithms for accurate estimation of resistors, inductors, and capacitors in VLSI circuits represents a significant advancement in the field of circuit design and analysis. Traditional methods based on complex mathematical simplification models often lead to errors, especially in systems with unique circuit architectures. However, the proposed methodology leverages machine learning techniques to overcome these challenges effectively. By using various adder circuits and collecting datasets comprising resistor, inductor, and capacitor values, the methodology successfully trains and optimizes machine learning models like adaboost, Tree, and kNN. Through rigorous evaluation, the adaboost model emerges as the most effective, exhibiting reduced mean square error and superior accuracy compared to the other models. The results of this study demonstrate the potential of machine learning in improving the precision and reliability of estimating critical parameters in VLSI circuits. By minimizing errors in estimation of R, L and C parameters, the proposed approach enhances circuit performance and efficiency. Moreover, the adaptability of the methodology to different circuit architectures ensures its applicability in diverse electronic devices and systems.

This research opens up new avenues for further exploration in the field, encouraging the application of machine learning techniques to tackle complex circuit design challenges. As the demand for high-performance, energy-efficient electronic systems continues to grow, the integration of machine learning in VLSI circuit design will play a decisive identity in shaping the future of modern computing and technology.

Abbreviation

VLSI – Very Large Scale Integration, R – Resistor, L – Capacitor, C – Inductor, kNN – K Nearest Neighbors, RMSE – Root Mean Square Error, MSE – Mean Square Error

Acknowledgement

None

Author Contributions

Saravanakumar C (Corresponding author): Led the conceptualization of the study, designed the methodology, conducted data collection, developed and implemented models, analyzed and interpreted results, and contributed significantly to the writing and editing of the manuscript, Usha Bhanu N: Contributed to the conceptualization of the study, participated in data collection, contributed to the analysis and interpretation of results, provided critical revisions to the manuscript, and approved the final version for submission.

Conflict of Interest

The authors declare no conflicts of interest.

Ethics Approval

This study does not involve human subjects or experiments on animals, thus negating the need for ethics approval.

Funding

This research is unfunded.

References

1. Amuru D, Zahra A, Vudumula HV, Cherupally PK, Gurram SR, Ahmad A, Abbas Z. AI/ML algorithms and applications in VLSI design and technology. Integration. 2023 Jun 16.
2. Govindaraj V, Arunadevi B. Machine Learning Based Power Estimation for CMOS VLSI Circuits. Applied Artificial Intelligence. 2021;35(13):1043-1055. DOI: 10.1080/08839514. 2021.1966885
3. Khan IU, Mittal N, Ansari MA. Applications of VLSI Design in Artificial Intelligence and Machine Learning. In: Kumar A, Tripathi SL, Rao KS, eds. Machine Learning for VLSI Chip Design. 2023.
4. Bansal M, Priya. Machine Learning Perspective in VLSI Computer-Aided Design at Different Abstraction Levels. In: Shakya S, Bestak R, Palanisamy R, Kamel KA, eds. Mobile Computing and Sustainable Informatics. Springer; 2022.
5. Saravanakumar C, Sivanantham P, Sowmiya P, Sasikala A, Syed Nasruddin S, Subhashini N. The Meritorious Effects of Machine Learning Algorithms in Assessment of Power Consumption by Arithmetic Circuits in IC Design. In: Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT). 2023. doi: 10.1109/ICECCT56650.2023.10179816
6. Yao E, Basu A. VLSI Extreme Learning Machine: A Design Space Exploration. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2017;25(1):60-74. doi: 10.1109/TVLSI.2016.2558842
7. Joshi S, Kim C, Ha S, Cauwenberghs G. From algorithms to devices: Enabling machine learning through ultra-low-power VLSI mixed-signal array processing. In: 2017 IEEE Custom Integrated Circuits Conference (CICC). 2017. pp. 1-9. doi: 10.1109/CICC.2017.7993650
8. Yadav V, Mehra R. Full Adder Design by Area Minimization. International Journal of Engineering and Technical Research. 2014;3(Special Issue):305-307.
9. Tavakolaee H, Baleghi Ardeshir. Fast Mux-based

Adder with Low Delay and Low PDP. Journal of Advances in Mathematics of Communications. 2019;7(3):385-392. doi: 10.22044/jadm.2018.7177.1853
10. Balasubramanian, Mastorakis. High Speed Gate Level Synchronous Full Adder Designs. WSEAS Transactions on Circuits and Systems. 2019;8(2):290-300.
11. Peng SY, Minch BA, Hasler P. Analog VLSI implementation of support vector machine learning and classification. In: IEEE International Symposium on Circuits and Systems (ISCAS). 2008. pp. 860-863. doi: 10.1109/ISCAS.2008.4541554
12. Hussain Chaudhury. Fast and High-Performing 1-Bit Full Adder Circuit Based on Input Switching Activity Patterns and Gate Diffusion Input Technique. Circuits Systems and Signal Processing. 2021;40:1762–1787. doi: 10.1007/s00034-020-01550-3
13. Huang Hu, He Liu, Ma Shen, Wu Xu, Zhang Zhong, Ning Yuzhe, Ma Yang, Yu Yang. Machine Learning for Electronic Design Automation: A Survey. ACM Transactions on Design Automation of Electronic Systems. 2021;26(5):1–46.
14. Abirami AP, Asritha K, Durga M, Saravanakumar C. A Time Varying Capacitance as Load to Study Area Efficient Arithmetic Circuits. In: Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). 2022. doi: 10.1109/ICAECT54875.2022.9807971
15. Chandrasekaran S, Nageswaran UB. A Mutated Addition–Subtraction Unit to Reduce the Complexity of FFT. Applied Nanoscience. 2023;13:2935–2944. doi: 10.1007/s13204-021-0227-5