SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V10I6P103

Volume 10 Issue 6, 17-26, June 2023

Original Article

An Analog Circuit Designing Model via Machine Learning for Stage Classification and Evolutionary Solution Optimization Algorithm

M. P. Varghese[1], T. Muthumanickam[2]

[1]*Department of Electronics and Communication Engineering, Vinayaka Mission's Research Foundation Deemed to be University, Salem, Tamil Nadu, India*
[2]*Department of Electronics and Communication Engineering, Vinayaka Mission's Kirupananda Variyar Engineering College, Vinayaka Mission's Research Foundation Deemed to be University, Salem, Tamil Nadu, India*

[1]*Corresponding Author : mpvargh@gmail.com*

Received: 24 March 2023 Revised: 18 May 2023 Accepted: 10 June 2023 Published: 30 June 2023

Abstract *- This work aims to propose a bottom-up, two-step process that streamlines the design of analogue devices by using machine learning techniques. The complicated nature of these difficulties, which involve numerous variables and objectives, necessitates using designers' skills and knowledge while designing analogue complementary metal-oxide-semiconductor (CMOS) integrated circuits. The study offers a framework detailing the unique characteristics of creating analogue circuits using machine learning, and it looks into the potential of libraries that contain open machine-learning models to assist designers. Traditionally, commercial CMOS or software simulations have been used to create neural network designs; however, these methods may not always provide the best results. A three-stage device design is used to validate the suggested method. Using a machine learning technique called the decision tree; the stage type is correctly predicted with an accuracy of 89.74% in the first phase. To create prediction logic, two rule induction techniques are also used. In the second step, four learning techniques, decision trees, random forests, gradient-boosted trees, and support vector machines, are used to forecast the typical parameters for each stage type. The support vector machine yields the best results and has the lowest error rates of all these methods.*

Keywords *- Artificial neural network, Analog system, CMOS circuit, Signal processing, Learning algorithm.*

1. Introduction

Finding the best circuit structure based on user requirements is a challenging part of designing electronic devices. These requirements describe the circuit's intended use, intended application, and necessary electrical standards. Designers need to know many simple circuits, their functioning concepts, and distinctive properties to complete this design task effectively. Designing intricate electronic modules and components such as functioning converters, filters, and generators is possible thanks to this understanding. Simple building blocks are the basis for various analogue circuits in electronics.

Essential steps in the design process include comprehending circuit functionality, potential variants, and circuit architecture theories. Electronic Design Automation (EDA) software is essential for enabling circuit design since it provides proper tools and component libraries that speed up the design process. Using such software reduces faults and bad circuit designs and saves time, money, and effort. Analogue circuits are sometimes necessary to the overall system design and cannot be replaced by digital processing. These consist of:

- Circuits on the input side are in charge of spotting, grabbing, boosting, and filtering signals from sensors, microphones, or antennas.
- Frequency synthesizers, phase-locked loops, analog-to-digital converters, and sample-and-hold circuits are examples of mixed-signal circuits. The interface between a system's input/output and a System-on-Chip's (SoC) digital processing units is made simpler by these building blocks.
- Circuits on the output side that transform digital signals into analogue signals amplify them and guarantee minimal distortion in the output signal.

1.1. Analog Circuit Design

Fig. 1 describes the design flow carried out during analog circuit design. The system-level components are

carried on to the circuit level, then to the implementation level, and vice versa. The comparison is made at each level, validation at each stage is carried out, and backtracing will be made if any discontinuities are found. Before beginning more intricate implementations at the device level, it is possible to examine the system architecture and improve overall system optimization using a hierarchical top-down design technique. By doing this, issues are found early in the design process, improving the chance that the design will work the first time and necessitating fewer if any, time-consuming redesign iterations.

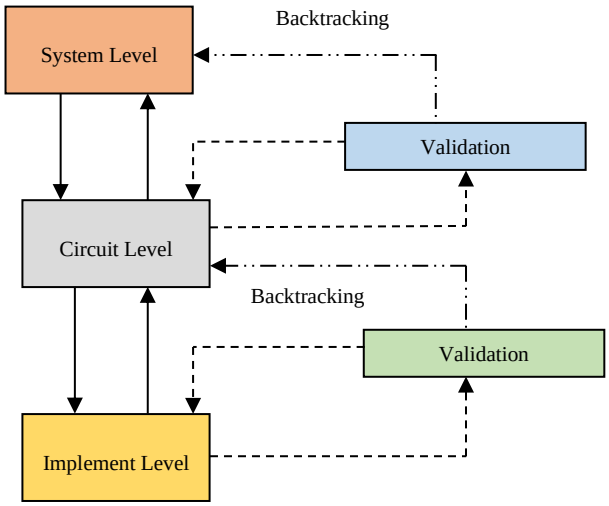

Fig. 1 Flow of tasks during an analog IC design

Specification translation is the process of converting specifications for each of the blocks from higher-level requirements. The block specifications may specify a device's gain, bandwidth, or transistor size depending on the models used at that level of abstraction. The size is then examined to ensure that the input specifications are met. In particular, the designer manually uses the tools to accomplish project goals, such as choosing the best combination of component sizes. We call this "circuit sizing". It is difficult to manually search for the best solution due to the objective function's search space's complexity, multidimensionality, and irregularity, which relate to design elements and circuit performance criteria[1].

1.2. Evaluation Criteria for the Data Quality Factor for the Set of Features

Despite recent advancements, no fully comprehensive automation tool is available to support the analogue design flow, indicating that automation tools and methodologies for analogue design are still not fully developed. In the pursuit of automating engineering processes, circuit design tools have incorporated machine learning and deep learning techniques [2-6]. These design methodologies based on machine learning require data collection, a deep understanding of electrical theory, and established techniques. Combining this expertise with knowledge of the benefits and characteristics of machine learning algorithms is essential to make the best design decisions [7-11]. To address the present demands, Hamolia and Melnyk emphasize the value of integrating modern automated design methodologies with EDA software [12]. In order to facilitate all phases of chip design, the authors also emphasise the birth of a brand-new scientific field that focuses on machine learning-based EDA. Ren et al. divide machine learning applications for EDA issues into predictors, optimizers, and generators [13].

They contend that in order to increase productivity, traditional EDA algorithms should work together with machine learning. It indicates that machine learning models could help the behavioural, structural, and physical design process in both top-down and bottom-up orientations. Similar topics have been the subject of some research, demonstrating fruitful outcomes, successful applications, and challenging issues.

A methodology to enhance the design of electronic circuits is presented by Dieste-Velasco et al. [14]. It is based on algorithms for artificial neural networks and the statistical method of experimental design. Before beginning more intricate implementations at the device level, it is possible to examine the system architecture and improve overall system optimization using a hierarchical top-down design technique[15]. By seeing issues early on in the design process, it is more likely that the design will work the first time and need little to no time-consuming redesign iterations. Topology selection on the top-down path is deciding on a group of blocks and the connections that will be used to carry out the input specifications.

The process of translating higher-level specifications into specifications for each block is known as specification translation. Depending on the models employed at that level of abstraction, the block specifications may define the device's gain and bandwidth or the transistors' size. After that, the sizing is examined to ensure the input requirements are met. They conclude that the suggested approach can be employed for effective parameter prediction and behavioural modelling of electronic circuits. Regression approaches are suitable for circuit modelling with quick speed and good accuracy, as demonstrated by Guerra-Gomez et al. in their study [16].

In their analysis of regression approaches used in the design of medium and large electronic circuits, Mina et al. analyse the speed at which various methods operate. In their discussion of earlier research on automating the design of integrated analogue circuits using MOS and CMOS technology, they also highlight the advantages of machine learning techniques for circuit designers, including supervised, unsupervised, and reinforcement learning [17]. Several academic articles [18-20] that look at current trends

and challenges in device placement and routing also discuss the application of machine learning at the physical level of chip design and improving designer productivity, mainly through machine learning [21] the development of new methodological methods in electronics results from advancements in data science and machine learning.

2. Methodology

The circuit is seen as a "black box" with known inputs and outputs. In contrast, the structural domain defines the circuit's layout, components, and connections. It makes it possible to transform a behavioural description into a collection of components and links that satisfy user needs. The physical domain focuses on putting the circuit on a PCB, considering chip area constraints and handling issues like component placement and routing [22]. Techniques from machine learning are used to find the best answers.

The structural domain explains the circuit's topology, encompassing stages, devices, and modules. A device can consist of one or more stages, a module may contain multiple devices, and a stage comprises components such as transistors, resistors, capacitors, diodes, etc. The physical domain efficiently positions and routes components, stages, devices, and modules in the PCB assembly. Machine learning assists in making accurate design choices and facilitating decision-making processes.

2.1. Determination of Potential Areas

A hierarchical bottom-to-top design technique was employed, consisting of four interconnected phases. Fig.2 illustrates the framework for machine learning-based circuit design, which encompasses the following stages:

- Finding appropriate components: This stage entails compiling libraries of elements with equations describing their electrical behaviour, such as transistors, resistors, capacitors, and diodes. Models for machine learning predict and categorise potential circuit components.
- The second step emphasises selecting appropriate stages to create the circuit device. Here are single-stage, two-stage, and multi-stage devices, each including circuitry supporting elements (usually transistors) and device elements [23-27]. All stages of schematics are arranged in libraries, and machine learning predicts and categorises their structure and behaviour.
- Connecting the stages: The stages are connected to create the gadget during this stage. Including additional circuits, such as feedback or correction circuits, is possible. Device behaviour, structure, placement, and routing on the PCB are predicted and categorised using device libraries and machine learning models.

The fourth step, "realising modules," is concerned with developing more sophisticated electrical goods. Modules can comprise one or more connected devices fulfilling a specific user requirement. There may be additional circuits for parameter enhancement or correction. Machine learning predicts and categorises the behaviour, structure, device location, and PCB routing by considering the transfer functions of the devices and the overall module function.

Fig.2 shows the framework for the design of analog circuits. Designing analogue circuits is a complex task that involves a deep understanding of circuit theory, device physics, and analog design principles. While machine learning frameworks have made significant progress in various domains, their application to analogue circuit design is still relatively limited. Nonetheless, some approaches and tools can aid in the design process.

Analog Circuit Synthesis: Machine learning frameworks can be used to automate the synthesis of analogue circuits by learning from existing designs. These frameworks typically rely on a combination of neural networks, reinforcement learning, and evolutionary algorithms to generate circuit architectures and parameter values that meet specified performance criteria.

- Circuit Optimization: Once a circuit is designed, machine learning techniques can be used to optimize its performance. This can involve automated parameter tuning, sensitivity analysis, and gradient-based optimization methods to fine-tune the circuit's behaviour and improve performance metrics.
- Performance Prediction: Machine learning models can be trained to predict the performance of analogue circuits based on their architecture and parameter values. This can help designers explore various design options more efficiently and reduce the need for extensive simulations or physical prototypes.

In the behavioural design of devices, stages are seen as "black boxes," focusing on their input and output parameters without worrying about the precise implementation details. On the other side, structural design investigates different potential configurations. The device's design keeps a feedback configuration in place [28, 29]. Machine learning algorithms, a subset of artificial intelligence, have been gradually incorporated into electronic design automation (EDA) software in recent years to help with engineering jobs across the electronics design process. Machine learning's use in EDA tools has drawn much interest. It offers much potential for automating various engineering jobs and answering many EDA problems.

The device must have input, intermediate, and output stages in line with Fig 3. Even though it does not magnify the voltage signal. The output stage must produce a low output resistance. This implies that the input and intermediate stages must provide the appropriate amplification.

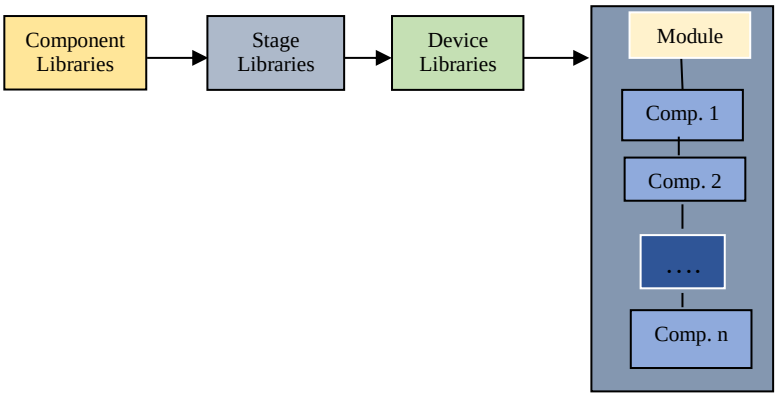

Fig. 2 Machine learning framework for designing analogue circuits

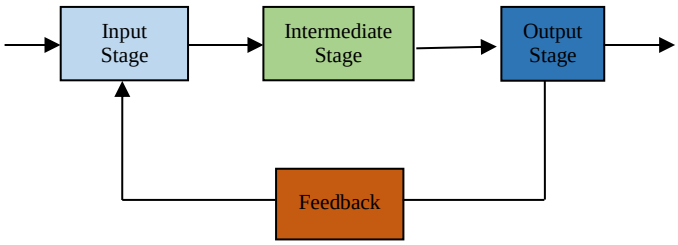

Fig. 3 Structure of multi-stage design

The first step is introducing a library containing regularly used device stages, their transfer functions, and specific characteristics. A "device" is an electronic gadget that boosts an electrical signal's current, voltage, or power. It transforms the voltage supply source's (VCC) electrical energy into a form that may be delivered to the load on the output circuit. Device circuits may use feedback circuits while transmitting signals from the input to the output [30]. Establishing the overall transfer function while considering each particular stage's input, intermediate, and output functions is crucial when working with multi-stage devices. A block diagram of a multi-stage device with multiple phases to meet various purposes is shown in Figure 3. The first input stage's primary objective is to provide high gain and effectively suppress common-mode signals to prevent unwanted interference from being amplified and transmitted to subsequent stages[31].

2.2. Feature Extraction and Visualization

Additionally, the first stage needs to have a high input impedance. This section describes how the suggested system can be used in an artificial neural network (ANN) that uses ANNs. A voltage adder's outputs are multiplied by a constant using an analogue multiplier. The voltage adder produces the signal from the non-linear function generator and the sum of the voltages from the weight matrix Wij, which represents the synaptic weights (Fig. 2). Each neuron composing the neural architecture is designed using operational amplifiers (op-amps)[32]. This means that to develop the ANN shown in Figure 3, we implemented the neural circuit's topology, as shown in Figure 2. It should be remembered that each neuron node comprises two components: an activation function (sigmoid) and a summation function. While the latter uses an array of op-amps and voltage limiters (diodes), the former is implemented using an op-amp inverting adder (created based on Figure 4). The function's block diagram in Figure 2 includes two continuous-time integrators (whose outputs rely on the feedback network), one analogue multiplier, one summation device, and one non-linear function generator to calculate the objective function gradient at the circuit level. The optimum parameters are the signals produced by the integrators[33]. This circuit displays a more resilient output (insensitivity to tiny perturbations) in the presence of parameter fluctuations. The function generator precisely determines the gradient of the goal function.

Fig. 4 shows the suggested machine learning-based device design approach. First, the stage type must be predicted, and then the typical parameters for each stage type must be forecast. The method postulates the presence of a library with device stages, where information on the purpose and makeup of each stage is generated. In order to learn from datasets, supervised machine learning methods, such as rule extraction algorithms, are used. This produces models predicting each stage's stage type (input, intermediate, or output) and critical parameters. To help designers choose the best stages for a three-stage device, this study used a machine learning-driven approach to analyse data about the stages (input, intermediate, and output). Additionally, significant factors related to each stage type were predicted using machine learning models.

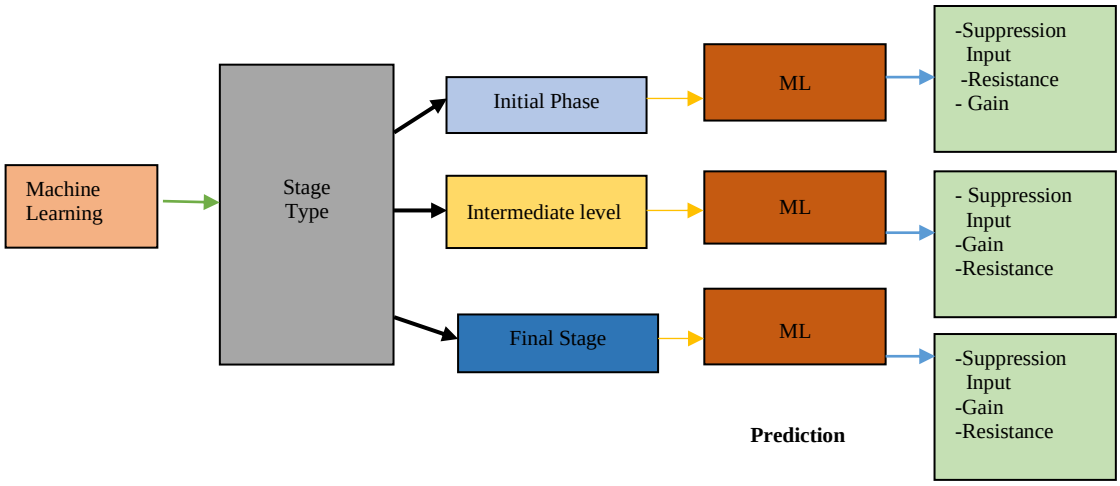

Fig. 4 Process for designing devices using machine learning

Techniques: Various techniques can be used for feature extraction, depending on the type of data and the specific problem. Some standard techniques include:

- Statistical measures: Calculating descriptive statistics like mean, median, variance, etc., to summarize numerical data.
- Dimensionality reduction: Methods like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), or t-SNE (t-Distributed Stochastic Neighbor Embedding) to reduce the dimensionality of data while preserving important information.
- Frequency analysis: Techniques like Fourier Transform or Wavelet Transform extract frequency-related features from signals.
- Image feature extraction: Methods like SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), or CNN-based approaches to extract features from images.

3. Result

When tested using a training/testing data ratio of 60%/40%, the model created for predicting the stage type had an accuracy of 89.74%. Figure 6 displays the confidence level (from 0 to 1) and the probability of accurate predictions for each stage type (input, intermediate, and output). Higher levels of confidence indicate better chances of accurate forecasts. Particularly for input stages, the confidence is consistently 1, showing a high confidence level in these phases' predicting[34]. However, the confidence levels for intermediate and output phases are less than 1, indicating less certainty in forecasting these stage types. This work subjected the same dataset to Rule induction and Trees to Rules, two rule extraction algorithms.

Fig. 5 describes the effect of power conservation at different frequency ranges when a noisy signal is encountered. Based on the gathered data, rule induction techniques use machine learning to build formal rules in an "if-then-else" manner. With the aid of these guidelines, it is possible to predict the different stage types and reach insightful conclusions. In addition to improved explanation and comprehension of the underlying logic relevant to the particular topic being studied, in our case, gadget construction, these methodologies have several benefits [35,36]. The extracted rules give information about choosing the stage type based on some typical parameters. An examination of the derived logic reveals that an output stage has a low output resistance, while an input stage has a high input resistance. They are referred to as intermediate phases for levels that do not fit these criteria.

3.1. Data Quality Factors Comparison

Results from the second algorithm, Trees to Rules, illustrate the rule-based logic used to identify the stage type. The stage is classified as an input stage if the common-mode rejection ratio (CMRR) is high Design Space Exploration: Machine learning techniques can assist in exploring the vast design space of analogue ICs.

Machine learning models can identify appropriate design choices, trade-offs, and design patterns by learning from a dataset of existing designs. This knowledge can be used to guide the designer during the exploration phase, suggesting promising design alternatives and providing insights into design decisions [38].

Design Automation: Machine learning can automate repetitive or time-consuming design tasks, allowing designers to focus on higher-level tasks. For example, machine learning models can automatically extract relevant features from circuit layouts or assist in circuit verification processes. This automation can improve the efficiency and productivity of the analog IC design process.

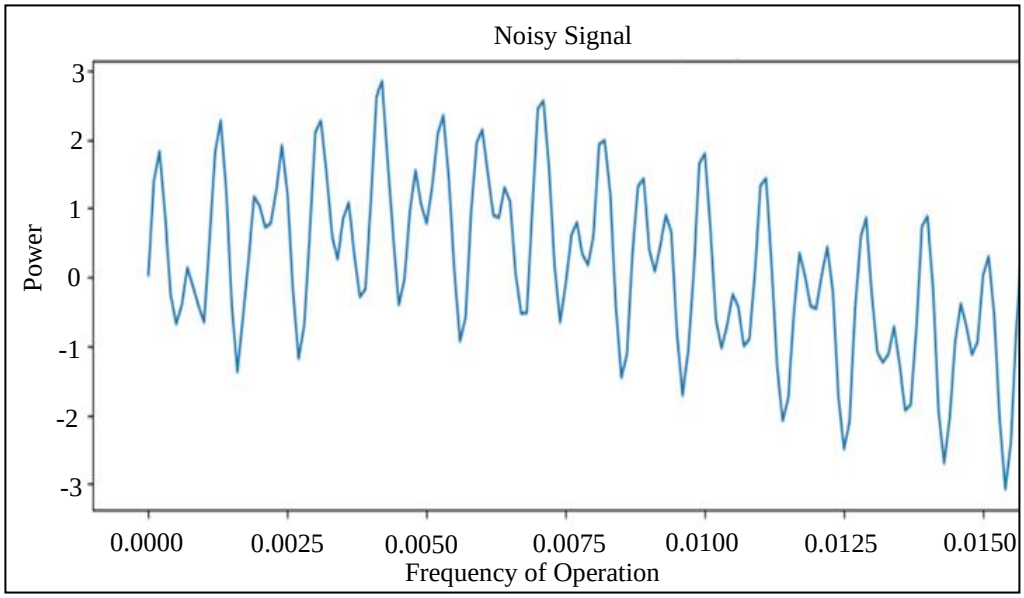

Fig. 5 Power distribution based on the frequency of operation for a noisy signal

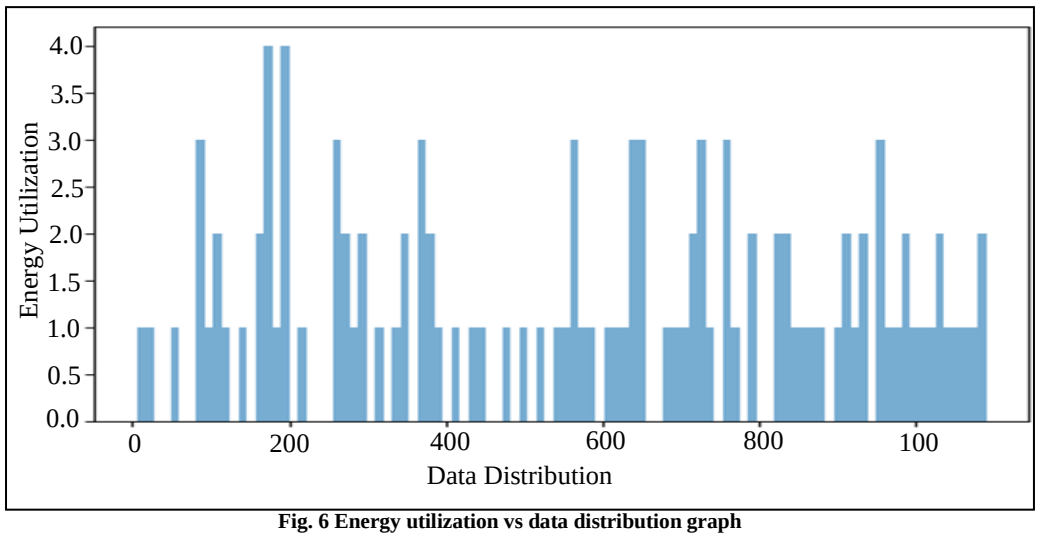

Fig. 6 Energy utilization vs data distribution graph

Fig. 6 describes how energy utilization is analyzed using the available data distribution. Two additional rule induction techniques emphasised the extracted logic even more. The second step entailed learning details about various device stages and doing a regression job using four machine-learning methods.

The Support Vector Machine algorithm showed the fewest mistakes out of all of these. The idea of an open library of machine learning models for circuits was implemented to assist designers in the challenging and time-consuming processes involved in developing analogue circuits, devices, and modules at many levels (structural, behavioural, and physical). While analogue design automation tools have been unavailable, manual investigation of solutions has resulted in expensive, challenging, and constrained designs for analogue design.

3.2. Interpretation of Results

The findings demonstrate the usefulness of rule induction algorithms in electronic knowledge discovery through data mining. Electronic Design Automation (EDA) software can be coupled with the created logic as a supporting tool[39]. The formalization of the analogue circuit design process and the automatic development of formal rules benefit designers in specific ways. In the second stage of the investigation, four different Machine Learning Algorithms, Decision Tree, Random Forest, Gradient Boosted Tree, and Support Vector Machine, were employed [41,42] to discover the most effective model for predicting

parameters associated with various types of phases. To create machine learning models, datasets at each stage were considered together with the associated typical parameters. Figure 7 is the only illustration of the prediction charts from the predictive models created for input stage 4. As can be seen, the Support Vector Machine method yields the best results for this regression task. Additional phases of parameter prediction produced similar results.

Fig. 7 describes the energy prediction evaluation for the predicated floor planning, placement and the error of occurrence that is predicted. Using evolutionary algorithms inspired by natural selection and genetic algorithms, the synthesis process can explore a vast design space and converge towards optimal or near-optimal solutions. The algorithm starts with an initial population of candidate circuits evaluated based on performance metrics, such as power consumption, speed, and noise characteristics. The best-performing circuits are selected for reproduction, and their genetic information is recombined and mutated to create new circuit variants for the next generation.

Machine learning in analogue integrated circuit (IC) design has gained significant attention and shown promising results. Machine learning techniques leverage large amounts of data and computational power to enable more efficient and accurate design processes. Here are some key findings and outcomes of applying machine learning in analogue IC design: Modeling and Optimization: Machine learning algorithms can model complex analogue behaviour, such as transistor-level circuit simulations, and optimize circuit performance[39]. Machine learning models can capture the relationships between input variables (e.g., transistor dimensions, circuit topology) and desired circuit metrics (e.g., power consumption, gain, bandwidth) by learning from existing data. These models can then guide the optimization process and predict the performance of new circuit configurations. In Circuit Synthesis, machine learning algorithms have automatically generated circuit topologies and configurations.

Machine learning models can learn the underlying patterns and design rules by training on a known circuit dataset and their corresponding performance metrics. This enables the generation of novel and optimized circuit designs based on user-defined objectives or constraints. Machine learning-based circuit synthesis can help accelerate the design process and explore a larger design space. Yield Enhancement: Analog IC design is sensitive to process variations, and ensuring a high yield is critical for manufacturing. Machine learning can assist in yield enhancement by predicting the impact of manufacturing variations on circuit performance.

By training on data from process variations, machine learning models can estimate the statistical variations in circuit parameters and guide the design process towards robust, yield-enhanced designs. It is important to note that the application of machine learning in analog IC design is still an active area of research and development. While it has shown promising results, challenges remain, such as the need for extensive and diverse datasets, the interpretability of machine learning models, and the incorporation of domain knowledge into the learning process. However, as machine learning techniques continue to advance and more data becomes available, the integration of machine learning in analog IC design is expected to have a significant impact, enabling faster, more robust, and innovative circuit designs.

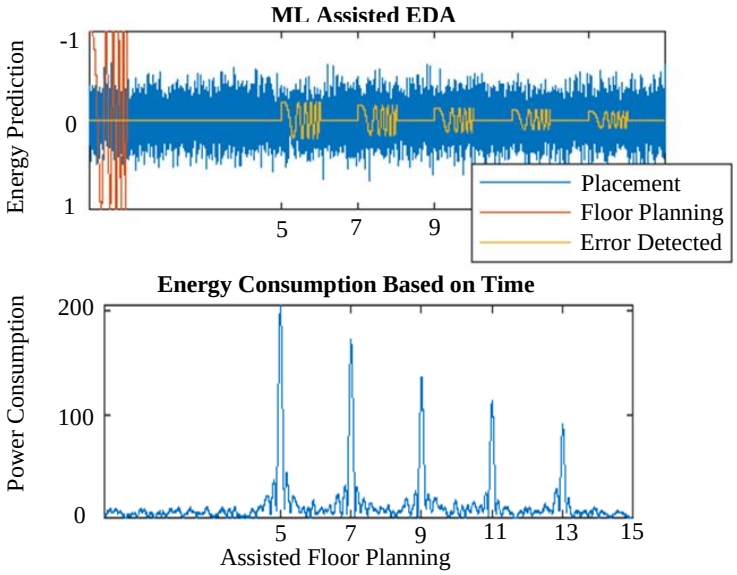

Fig. 7 Energy analysis based on placement, floor planning and error detection

4. Conclusion

In order to simplify the design of analogue devices, the study offers a two-step process that uses rule induction approaches and machine learning algorithms. The goal is to help designers choose appropriate stages for device realisation by offering predictions on stage types and producing rule-based logic. The method also helps select appropriate parameters based on user requirements for a particular stage type. The approach's effectiveness was demonstrated by constructing a three-stage system where the capabilities and crucial factors were fully known. Using the Decision Tree algorithm, a supervised machine learning technique, the stage types were first classified with an accuracy of 89.74%. The evolutionary synthesis of analogue IC design is a promising approach that leverages evolutionary algorithms to automate and enhance the process of designing analogue circuits. It offers the ability to explore a vast design space, handle trade-offs, and generate innovative solutions. While it is not a substitute for human expertise, it can significantly improve the efficiency and effectiveness of analog IC design.

Moreover, evolutionary synthesis can overcome some limitations of traditional manual design approaches. Analog IC design is a time-consuming and iterative process that heavily relies on the designer's expertise and intuition. Evolutionary synthesis reduces human effort and enables the exploration of a larger design space by automating parts of the design process. It can also generate innovative circuit topologies and configurations that human designers might not have considered, leading to novel solutions and improved performance. This contrasts with the benefits of various EDA tools and design approaches for digital IC design.

References

[1] Kirit V. Patel et al., "Design and Implementation of Effective Elliptic Curve Cryptography Accelerator using Hardware/Software Co-Design on Zynq Board," *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 327-335, 2022. [CrossRef] [Publisher Link]

[2] Zhiyao Xie et al., "FIST: A Feature-Importance Sampling and Tree-Based Method for Automatic Design Flow Parameter Tuning," *25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 19-25, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jihye Kwon, Matthew M. Ziegler, and Luca P. Carloni, "A Learning-Based Recommender System for Autotuning Design Flows of Industrial High-Performance Processors," *56th ACM/IEEE Design Automation Conference (DAC)*, pp. 1-6, 2019. [Google Scholar] [Publisher Link]

[4] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective Approaches to Attention-Based Neural Machine Translation," *Computation and Language arXiv: 1508.04025*, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] C. Venkataiah, K. Satya Prasad, and T. Jaya Chandra Prasad, "Effect of Interconnect Parasitic Variations on Circuit Performance Parameters," *International Conference on Communication and Electronics Systems (ICCES)*, pp. 1-4, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[6] Mohamed Saleh Abouelyazid, Sherif Hammouda, and Yehea Ismail, "Connectivity-Based Machine Learning Compact Models for Interconnect Parasitic Capacitances," *ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD)*, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Aixi Zhang et al., "Field-Based Parasitic Capacitance Models for 2D and 3D Sub-45-nm Interconnect," *4th Asia Symposium on Quality Electronic Design (ASQED)*, pp. 110-116, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[8] Nur Kurt Karsilayan, Jim Falbo, and Dusan Petranovic, "Efficient and Accurate RIE Modeling Methodology for BEOL 2.5D Parasitic Extraction," *IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 519-522, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[9] John H. Lau, "Evolution Challenge and Outlook of TSV 3D IC Integration and 3D Silicon Integration," *International Symposium on Advanced Packaging Materials (APM)*, pp. 462-488, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[10] Weibing Gong et al., "A Parasitic Extraction Method of VLSI Interconnects for Pre-Route Timing Analysis," *International Conference on Communications, Circuits and Systems (ICCCAS)*, pp. 871-875, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[11] P. Rajeswari, and Theodore S Chandra, "Partitioning of VLSI Circuits on the basis of Standard Genetic Algorithm and Comparative Analysis of Partitioning Algorithms," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 12, pp. 126-133, 2022. [CrossRef] [Publisher Link]

[12] Kelin J. Kuhn et al., "Process Technology Variation," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2197-2208, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[13] Lijie Sun et al., "A Novel Customized RC Tightened Corner Modeling Methodology using Statistical SPICE Simulation in Advanced FinFET Technology," *14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pp. 1-4, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] Kaiming He et al., "Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1026-1034, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[15] Johannes de Fine Licht et al., "Transformations of High-Level Synthesis Codes for High-Performance Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1014-1029, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Razvan Nane et al., "A Survey and Evaluation of FPGA High-Level Synthesis Tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 10, pp. 1591-1604, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[17] Guyue Huang et al., "Machine Learning for Electronic Design Automation: A Survey," *ACM Transactions on Design Automation of Electronic Systems*, vol. 26, no. 5, pp. 1-46, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ecenur Ustun et al., "Accurate Operation Delay Prediction for FPGA HLS using Graph Neural Networks," *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1-9, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Walter Lau Neto et al., "LSOracle: A Logic Synthesis Framework Driven by Artificial Intelligence: Invited Paper," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mohamed Baker Alawieh et al., "High-Definition Routing Congestion Prediction for Large-Scale FPGAs," *25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 26-31, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Fanchao Wang et al., "Accelerating Coverage Directed Test Generation for Functional Verification: A Neural Network-Based Framework," *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, pp. 207-212, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Fabian Pedregosa et al., "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825-2830, 2011. [Google Scholar] [Publisher Link]

[23] Paolo Mantovani et al., "Hl5: A 32-bit RISC-V Processor Designed with High-Level Synthesis," *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Dejun Jiang et al., "Could Graph Neural Networks Learn Better Molecular Representation for Drug Discovery? A Comparison Study of Descriptor-Based and Graph-Based Models," *Journal of Cheminformatics*, vol. 13, pp. 1-23, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Walter Lau Neto et al., "LSOracle: A Logic Synthesis Framework Driven by Artificial Intelligence: Invited Paper," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Ecenur Ustun et al., "LAMDA: Learning-Assisted Multi-Stage Autotuning for FPGA Design Closure," *IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 74-77, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[27] Steve Dai et al., "Fast and Accurate Estimation of Quality of Results in High-Level Synthesis with Machine Learning," *IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 129-132, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[28] Walter Lau Neto et al., "SLAP: A Supervised Learning Approach for Priority Cuts Technology Mapping," *58th ACM/IEEE Design Automation Conference (DAC)*, pp. 859-864, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[29] Shobha N. Pawar, Pradeep B. Mane, and Milind P. Gajare, "Ku Band Ultra-Low Phase Noise PLL Frequency Synthesizer using 0.18 µm CMOS Process," *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 10-25, 2022. [CrossRef] [Publisher Link]

[30] Cunxi Yu, and Zhiru Zhang, "Painting on Placement: Forecasting Routing Congestion using Conditional Generative Adversarial Nets," *Proceedings of the 56th Annual Design Automation Conference*, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[31] Mouna Karmani et al., "A New Dual Differential FullAdder Design for CED Based Fault Tolerant Circuits," *International Journal of Engineering Trends and Technology*, vol. 69, no. 12, pp. 257-266, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Yasasvi V. Peruvemba et al., "RL-Guided Runtime-Constrained Heuristic Exploration for Logic Synthesis," *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1-9, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[33] Ghasem Pasandi, Shahin Nazarian, and Massoud Pedram, "Approximate Logic Synthesis: A Reinforcement Learning-Based Technology Mapping Approach," *20th International Symposium on Quality Electronic Design (ISQED)*, pp. 26-32, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[34] Cunxi Yu, and Wang Zhou, "Decision Making in Synthesis Cross Technologies using LSTMs and Transfer Learning," *ACM/IEEE 2nd Workshop on Machine Learning for CAD (MLCAD)*, pp. 55-60, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[35] Shuangnan Liu, Francis C. M. Lau, and Benjamin Carrion Schafer, "Accelerating FPGA Prototyping through Predictive Model-Based HLS Design Space Exploration," *Proceedings of the 56th Annual Design Automation Conference*, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[36] Biying Xu et al., "WellGAN: Generative-Adversarial-Network-Guided Well Generation for Analog/Mixed-Signal Circuit Layout," *56th ACM/IEEE Design Automation Conference (DAC)*, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[37] Deepti Raj, A. B. Kalpana, and Manoj Kumar Singh, "Optimal FSM's State Encoding for Low power using Dynamic Boundary Difference Mutation Strategy in Evolutionary Programming," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 10, no. 1, pp. 160-167, 2023. [CrossRef] [Publisher Link]

[38] Cunxi Yu et al., "DAG-Aware Logic Synthesis of Datapaths," *53rd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1-6, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[39] Walter Lau Neto et al., "LSOracle: A Logic Synthesis Framework Driven by Artificial Intelligence: Invited Paper," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[40] Jason Luu et al., "VPR 5.0: FPGA CAD and Architecture Exploration Tools with Single-Driver Routing Heterogeneity and Process Scaling," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 4, no. 4, pp. 1-23, 2011. [CrossRef] [Google Scholar] [Publisher Link]